

道路網応用のための空間索引木の提案と最短経路探査への応用

蒲原 智也^{†1} 上島 紳一^{†1}

本稿では、道路網をエッジ重み付きグラフととらえ、ネットワークボロノイ図を用いてグラフを分割し、構成した空間索引木とそれを用いた最短経路探査法を提案する。空間索引木の生成にはまず、道路網を表すグラフに対して、ネットワークボロノイ分割を行うことで、全体領域を部分領域に分割する。この分割によって得られた部分領域ごとに、経路探査のための前処理を実行して空間索引を求めておくことによって、経路探査の計算コストを大幅に減らすことができる。各部分領域は、領域内・隣接領域間の最短経路、領域の大きさの指標などの3種類の空間索引を持つ。さらに、部分領域をランダムに選択して隣接領域単位で統合することによって、より広い部分領域を得る。この操作を繰り返すことによって領域の包含関係を持ち、平衡木の性質を持つ空間索引木を漸次的に生成する。次に、提案木構造を用いて、領域の統合を用いた再帰的な経路探査アルゴリズムと領域の大きさの指標を用いた経路探査アルゴリズムを与え、その有効性について議論する。提案手法の有効性を確認するため、国土地理院数値地図の道路網データに実際に適用する。まず空間索引木の生成時の評価として、母点の選択確率の木構造への影響、生成時間・サイズなどについて調べる。次に経路探査時の評価として、2つの提案アルゴリズムの定量的な評価をA*探索アルゴリズムと関連研究の階層型経路ビューモデルと比較しながら行う。提案手法により、デジタル道路情報を利用して、空間索引木を構成でき、2点間の経路探査やエッジに沿った距離に基づく範囲問合せなどの時間を圧縮できる。

Proposal of Spatial Index Tree and Its Application to Shortest Path Search in Road Networks

TOMOYA KAMBARA^{†1} and SHINICHI UESHIMA^{†1}

In this paper the authors propose a new spatial index tree and discuss its application to shortest path problems. In view of road maps as graphs with edge costs, a given graph is partitioned by use of Network Voronoi Diagram, to generate subregions of the target space. Preprocessing spatial indices in each

subregion, computation cost in shortest path search is largely reduced by use of the indices. Spatial indices are (1) shortest paths in each subregion, (2) shortest paths between adjacent subregions, and (3) the maximum/minimum path lengths of each subregion. Furthermore, wider subregions are generated by integration of those adjacent subregions that are randomly selected. By repeating partitionings and integrations, spatial index tree is successively generated in a bottom-up fashion. The tree is a balanced tree that has inclusion relations among subregions, in the sense that higher-level nodes cover the regions of lower-level nodes. Then, two shortest path search algorithms are proposed by use of the spatial index tree. One is a reflexive search algorithm using subregions integration, and the other is a search algorithm based on distance estimation using spatial indices. In order to verify efficiency of the proposed approach, numerical simulations are provided for digital road map data issued from Geographical Survey Institute. One is for investigation of (1) effects of probability in random selection of generators on the structure of spatial index tree, (2) generation costs and file size for the spatial index tree, and the other is for evaluation of search time costs of two proposed shortest path search algorithms, in comparison with A* search algorithm and the existent hierarchical path view model. The approach can be applied to various search algorithm to reduce search time costs, including distance range queries, k-nearest neighbor search.

1. はじめに

近年、地理情報システムや道路網応用の研究分野で、最短経路探査、範囲問合せ(DRQ)、k-最近傍探査問題(kNN)などに有効な空間索引の研究が注目を集めている¹⁾⁻⁵⁾。これらの研究では、店舗などの位置の固定したオブジェクトだけでなく、移動体の位置などに対する問合せに答えるため、道路網をグラフと見なし、オブジェクトと道路網を分離した手法をとっている。

本稿においても、交差点をノード、交差点間を結ぶ道路をエッジとし、道路片の長さを重みとしたエッジ重み付きグラフとして道路網をモデル化する。これを分割して得られる部分グラフ群を用いた木構造を空間索引木として提案する。

ここでは、2種類のネットワークボロノイ図⁶⁾⁻⁸⁾を用いて、ランダムに選択した母点によるノード間距離を基準にしたグラフ分割を行い、隣接領域間を統合することで、対象道路網

^{†1} 関西大学大学院総合情報学研究所

Graduate School of Infomatics, Kansai University

に対して、木の上位ノードを生成して、木をボトムアップに生成する。

その結果、同じ母点選択確率に対しては、ほぼ一定の高さとノード次数を持つ平衡木となる。空間索引木は、領域が包含関係をも持つ。つまり、より上位のノードが持つ領域が、下位のノードの領域を完全に包含する。上位ノードの境界が下位ノードの境界に含まれるという特徴がある。

各部分領域は、領域内・隣接領域間の最短経路、領域の大きさの指標などの3種類の空間索引を持つ。これは、部分領域に対して、部分領域を通過する経路と距離見積りをあらかじめ領域ごとに計算しておくことに相当する。

次に、道路網上の2地点間の最短経路探索手法として、領域統合型の探索手法と距離見積りを用いた探索手法を提案し、A*経路探索法と既存の関連手法である階層型経路ビューモデル^{9),10)}と比較しながら有効性を確認する。前者では、道路網上の最短経路探索において、詳細に探索しなければならない領域は、出発点、目的地点を含む領域となり、計算時間が短縮される。後者では、領域の大きさの指標を用いて出発点から目的地点までの距離を見積り、見積り距離を用いて経路をたどる。

提案手法は、領域の分割に、ネットワークポロノイ図を利用しながら、グラフにSkipList¹¹⁾の考え方を応用して、ノードのランダム選択と、領域分割・統合を組み合わせたボトムアップな階層化手法となっている。

提案木構造は、道路網を基盤としているため、様々な道路応用問題に利用できるものと考えられる。

2章では、関連研究と比較し、3章でネットワークポロノイ図と提案木構造のボトムアップな生成法について述べる。4章では、提案木構造の生成アルゴリズムと各部分領域での空間索引の構成について、5章では、2種類の最短経路探索アルゴリズムについて述べ、6章で実データを用いた定量的評価を行う。

2. 関連研究との比較

これまで、道路網モデルとして、交差点をノード、道路片をエッジとした重み付きグラフを用いた、多数の距離索引付け手法が提案されている。

Kolahdouzanら⁵⁾は、ネットワークポロノイ図による領域分割を用いて、k-NN問題の効率的な計算法を提案している。同手法では、索引化の対象の位置を基準にグラフを生成している。このため索引の対象の種類ごとにグラフを生成する必要がある。また、索引化の対象ごとのグラフは互いにまったく異なり、階層化などを適用する構造となっていないため、

k-NN以外の道路網応用へ容易に適用できない。

Kriegelら³⁾は、前処理として、参照点をグラフに埋め込み、すべての索引化の対象からの距離をあらかじめ計算して、ベクトル空間に写像することで、DRQとk-NNに適用している。問合せに対しては、参照点までの距離を基準に距離を見積もり、フィルタリングと精選の2段階探索により高速化している。同様に、Hjaltasonら¹²⁾は、距離空間に一般化してリビッツ埋め込みと多次元ベクトルの低次元化の観点から、類似探索についてまとめている。

Huら⁴⁾は、平面上のノードとオブジェクト間の距離を離散化して近似し、distance signatureを構成することで、格納コストと探索コストを低減化している。カテゴリの構成には、平面上のオブジェクトの分布としてデータセットの密度を用いている。

Sankaranarayananら²⁾は、経路コヒーレンスの概念を用いて、特定の地点から、第1近傍として到達可能な地域を色分けして、四分木を基礎としたSILKデータ構造を用いて地点を索引付けし、k-NN問題に適用している。

また、直接にグラフを対象としていないが、Eppsteinら¹³⁾は、四分木構造上でオブジェクトの存在しないノードを削除して、データに到達するための冗長なパスを短縮した圧縮四分木を提案し、範囲検索の効率化を行っている。

Jingら^{9),10)}は、2次元方向それぞれに空間を分割して空間的近さを求める独自の方法によって、グラフを分割・階層化して作成したデータ構造を提案している。分割には、得られた部分グラフ上の、すべての地点の組合せ間の最短経路を前処理で求め、グラフ上のすべての2地点間の最短経路を経路ビューとしている。

これらに対して、提案手法では、ネットワークポロノイ分割を利用しながら、平面グラフにSkipList¹¹⁾の考え方を応用して、ノードのランダム選択と、領域分割・統合を組み合わせボトムアップに木構造を生成している。この木構造は、領域の包含関係を考慮した平衡木となっている。

また、ネットワークポロノイ図を用いて平面グラフのノードの位置関係に基づいてグラフを分割することは、エッジに付けられた距離情報をもとに、大まかに抽象化した地図を構成することに相当する。

連続平面のポロノイ図の階層化手法は、CGの分野でLOD(Level Of Detail)表現を可能にした詳細度別表示に用いられている¹⁴⁾。この構成方法は、定義した評価関数をもとに階層化するため、各階層での領域の包含関係が満たされず、範囲問合せにおける範囲の絞り込みが行えないため、道路網応用に適用しにくい。

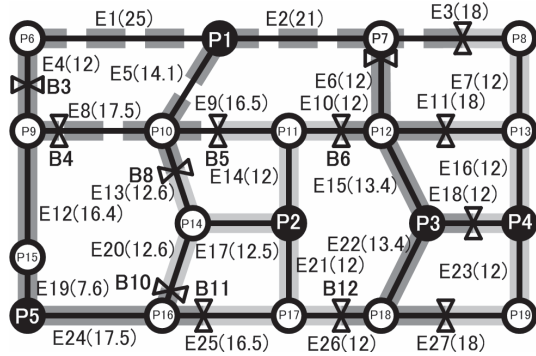


図 1 母点集合 P_1, \dots, P_5 に対するネットワークボロノイ図
 Fig. 1 Network voronoi diagram for generators P_1, \dots, P_5 .

次章では、領域分割に用いるネットワークボロノイ図を定義し、提案木構造の生成方法の基本的な考え方について述べる。

3. 空間索引木の構造

3.1 ネットワークボロノイ図によるグラフ分割

道路網において、交差点をノード、道路片をエッジと見なした無向グラフ $G = (N, E)$, (N : ノード集合, $E \subseteq N \times N$: エッジ集合) を考える。さらに、各エッジは重みを持ち、各ノード n は 2 次元での座標 $(n.x, n.y)$ を持つものとする。図 1 では、ノードを P_i , エッジを E_i で示し、各エッジの重みを括弧内に示す。たとえば、 P_1 - P_{10} 間は、 E_5 で接続され重み 14.1 を持つ。

G に対して、ノードの部分集合を母点集合として与えると、次の 2 種類のネットワークボロノイ図を定義できる⁶⁾⁻⁸⁾。

[ノードネットワークボロノイ図](以後、ノード NV 図) ノード NV 図は、母点以外の各ノードに対して最も近い母点(ノードの所属母点という)を決定し、母点に関して G のすべてのノードを分割した図をいう。

ノード NV 図の生成は、並列ダイクストラ法⁸⁾を用いる。これは、複数の母点をスタートノードとし、各ノードに対して、1 つの所属母点を決定するダイクストラ法の拡張である。

図 1 で、母点集合は、 $\{P_1, P_2, P_3, P_4, P_5\}$ である。 P_1 を所属母点とするノードは、 P_{10}, P_7, P_6 である。特に P_6 について見ると、母点 P_1 に重み 25 で最も近く、他の母点

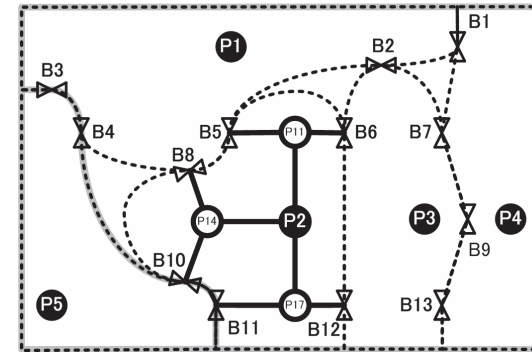


図 2 P_1, \dots, P_5 の母点領域の隣接関係図
 Fig. 2 Adjacency among generator regions for P_1, \dots, P_5 .

P_2 へ重み 54.6, P_3 へ重み 71.4, P_4 へ重み 83.4, P_5 へ重み 36 となっていることが確認できる。他のノードに対しても同様である。また、あるノードが 2 つ以上の母点到等距離である場合については、所属母点は一意に決定できないため、ここではどちらか一方の母点到所属するものとして扱う。

[エッジネットワークボロノイ図](以後、エッジ NV 図) G のエッジに関して、エッジ上の任意の地点から各母点への重みを考えれば、各地点の所属母点を決定することができる。この場合、1 つのエッジ上のすべての地点が同一母点到所属する場合と、1 つのエッジの途中で所属母点が変わる場合がある。途中でエッジの所属母点が変わる地点を境界点といい、そのようなエッジを境界エッジという。図 1 では、 \times が境界点を示す。 G のすべてエッジを所属母点で分割して得られた図をエッジ NV 図という。

すべてのエッジの所属については、各エッジごとにノード NV 図の生成後に決定できる。

図 1 で、 E_{12} 上のどの地点でも最も近い母点は、 P_5 である。したがって、 E_{12} は母点 P_5 に所属する。境界点を持つエッジは、途中のどこかで所属母点が変わるエッジである。たとえば、 E_4 の両端のノード P_6, P_9 の所属母点とその重みを見ると、 P_6 は重み 25 で P_1 に所属し、 P_9 は重み 24 で P_5 に所属する。所属する母点異なるため、 E_4 上に境界点があることが確認できる。このときの境界点の位置は、 P_6 から P_9 までの E_4 の重み 5.5 の地点となっている。よって、 E_4 上の境界点は、母点 P_1 と P_5 より重み 30.5 の地点である。[母点領域] 母点とその母点到所属するノードとエッジを集めてできる領域を母点領域と呼ぶ。つまり、各母点領域は、(i) 1 つの母点、(ii) 母点到所属するノード、(iii) 母点到所属す

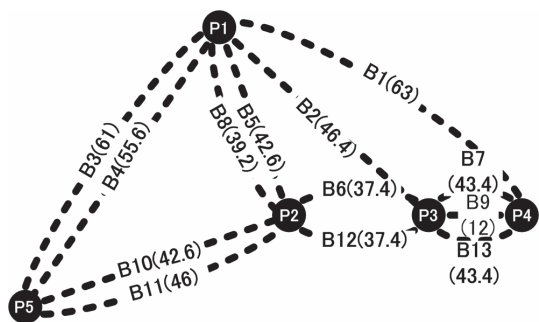


図 3 母点領域の隣接関係を示すグラフ (G_{11})
 Fig. 3 Graph of adjacency among generator regions (G_{11}).

るエッジ, (iv) 領域の端を示す境界点, (v) 境界点を持つエッジの境界点までの部分から構成される。ここで、母点領域間は、それぞれ有限個の境界点で、他の母点領域に隣接していることに注意する。

図 2 は、母点 P1 から P5 に対する母点領域の隣接関係を示す。P2 の母点領域は、境界点 B5, B8 で P1 に隣接し、境界点 B6, B12 で P3 に隣接し、境界点 B10, B11 で P5 の母点領域に隣接している。

各母点領域において、母点は、その母点領域の中心の役割を持ち、母点領域を代表する。提案する空間索引木においては、各母点が木の上位ノードを構成する。

3.2 母点領域の統合による木の階層化

・母点領域の隣接関係の抽出

前節のグラフ分割で得られた母点領域をノードとして扱う。母点領域間の経路は、必ず境界点を通るため、境界点を利用した経路で構成することができる。つまり、隣接する 2 つの母点領域間の境界点を接続部として、その境界点までのそれぞれの母点からの最短経路を用いて、ノードの隣接関係を結ぶエッジを構成できる。

前節で生成した境界点は、以降の階層化においては、その境界点を経由した経路として見る。

たとえば、図 3 で G_{11} 上の B3 のエッジは、P1 と P5 を結ぶ B3 を経由した経路であり、それを構成した原地図上での経路と距離は、図 1 のノード名を用いて以下のとおりになる。

$$B3|G_{11} = path(P1, P6, P9, P15, P5)|G$$

$$B3(61)|G_{11} = E1(25) + E4(12) + E12(16.4) + E19(7.6)|G \quad (1)$$

同様に、B4 に対しては、

$$B4|G_{11} = path(P1, P10, P9, P15, P5)|G$$

$$B4(55.6)|G_{11} = E5(14.1) + E8(17.5) + E12(16.4) + E19(7.6)|G \quad (2)$$

となる。このように、母点間を結ぶ最短経路は、境界点ごとに作成する(図 3)。母点 P1 と P5 を結ぶ場合、他母点領域(たとえば P2 の母点領域)を経由する経路は考慮しない。

同様に、他の隣接する 2 つの母点領域の母点間の経路も求めることができ、図 3 のグラフ G_{11} が得られる。 G_{11} では、エッジは、(1), (2) などの経路をエッジとして表した仮想的なエッジであり、領域の間の接続部の境界点と両母点間の距離を明示している。同一の組合せの母点間は、境界点ごとに結ばれるため、エッジが複数存在するので、 G_{11} は多重グラフとなる。

G_{11} は、平面上の母点間の隣接関係を通して、母点間の相対的な位置関係を示している。ノードは母点領域であるので、それぞれの母点領域が分散的に隣接関係を持つことで、母点領域の位置関係を示している。

以降、上位層と下位層のグラフを区別するために階層 i でのグラフ G を G_{1i} とする。

・母点領域の統合

G_{11} において、各ノードは領域を示すため、2 つ以上のノードを統合すれば、より広い領域を示すノードを生成できる。このため、 G_{11} のノード集合の部分集合を、新たな母点集合として与え、 G_{11} に対して、ノード NV 図を生成して、 G_{11} のノード集合を分割する。つまり、新たな母点集合に対する G_{11} のノードの母点への所属関係を一意に決定する。

図 3 で、 G_{11} の全ノード集合 $\{P1, P2, P3, P4, P5\}$ の部分集合として、新たな母点集合として $\{P3, P5\}$ を与えたと仮定する。すでに、 G_{11} では、母点間の経路の長さが求められているため、図 3 のエッジの重みに基づいて計算すると、ノードの所属関係により、ノード NV 図を生成できる。

G_{11} のノード P1, P2, P4 は母点 P3 に所属し、P5 の母点領域は、そのままである。これにより、母点領域 P1, P2, P4 は、母点領域 P3 に吸収されて統合される(図 4)。母点となるノードは、自身に所属するノードの領域をすべて統合する。 G_{11} 上で

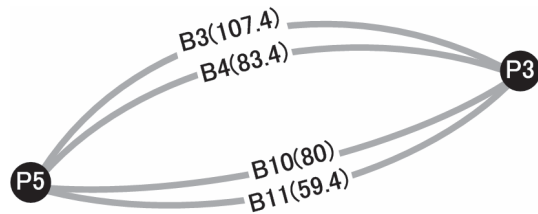


図 4 母点領域の隣接関係を示すグラフ (G_{12})
Fig. 4 Graph of adjacency among generator regions (G_{12}).

P_3 の母点領域に所属: P_1, P_2, P_3, P_4

P_5 の母点領域に所属: P_5

(3)

となる。ここで、母点 P_1, P_2, P_3, P_4, P_5 は、 G_{11} のノードとして見ている。

次に、 G から G_{11} を構成したときと同様に、母点間の最短経路を境界点ごとに求めれば、2つのノード P_3, P_5 を持つグラフ G_{12} を作成できる。 P_3, P_5 を結ぶ経路は、境界点 B_3, B_4, B_{10}, B_{11} を通る。

G_{12} では、 G_{11} 上に境界点を新たに生成するのではなく、原地図 (図 1) 上の境界点を利用する。つまり、統合前の領域の端の境界点の中から、統合により得られた領域の端を示す境界点を選ばれる。これにより、統合前の領域の境界点、統合後の領域の境界点を含むため、領域間に包含関係を持たせることができる。

上の統合の様子を原地図 G の上で見てみる。図 5 は、母点 P_3, P_5 の所属関係に基づき、統合される場合の領域を示す。 G_{11} の P_1, P_2, P_4 と P_3 の間にある境界点グループ A, B, C は、それぞれ G_{12} では境界点でなくなり、 G_{12} での B_3, B_4, B_{10}, B_{11} が境界点となる。このように、統合により得られた領域の端を示す境界点の数は、統合前と比較して減少する。

以降、上位層では下位層と同じ境界点が用いられるために、階層 i での境界点 j を B_{ij}^j と表すことにする。

・階層化

ノードと母点領域を同一視することを行い、ノード NV 分割と統合を繰り返すことで、包含関係を維持した木構造をボトムアップに生成することができる。以下、全領域を包含するノードが発生するまで上位層の生成を繰り返す。図 6 は、ノードの所属関係、領域の包含関係を示した木構造である。

また、図 7 に階層ごとに作成されたグラフを示す。 G_{10} (原地図) から G_{11} を生成すると

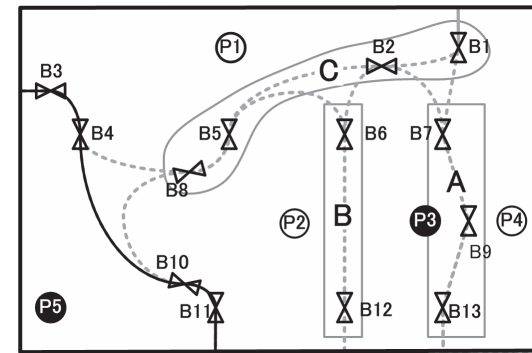


図 5 領域 P_3, P_5 による領域 P_1, P_2, P_4 の統合
Fig. 5 Merging of generator regions for P_1, P_2, P_4 by generator regions for P_3, P_5 .

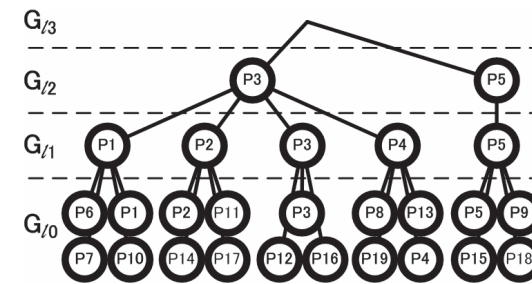


図 6 領域の包含関係を示す木構造表現
Fig. 6 Tree representation for inclusion relations of region.

き、母点のランダム選択を行い、ノード NV ・エッジ NV 分割により、境界点が発生している。 G_{11} は、分割された領域間の経路を示し、これが上位層を生成する場合、分割対象となるグラフである。 G_{12} は、 G_{11} で上位層を生成した結果、得られたグラフである。

3.3 空間索引

提案木では、分散型の空間索引として、各母点領域が次の 3 種を持つ。

- (1) 各階層での隣接する母点領域間で求めた母点間の最短経路と距離
- (2) 各階層での母点領域内で
 - すべての境界点間の最短経路と距離
 - すべての境界点と母点間の最短経路と距離

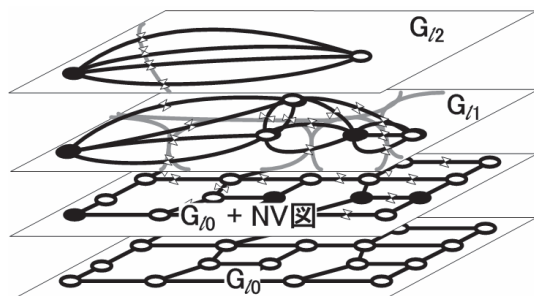


図 7 空間索引木の階層構造
Fig. 7 Hierarchical structure of spatial index tree.

(3) 各階層での母点領域の大きさの指標

(1) は、前節において、階層化の手順の間に求められる。グラフ上でランダムな母点選択を行っても、実際に移動するのはグラフ上であるため、境界点を中心にした道路に沿った最短経路をつねに求めておく必要がある。

(2) は、母点領域を外部から通過するための経路と距離である。また、母点を母点領域の代表点と見なして、該当する母点領域の境界点までの距離である。

(3) は、次に示す。

上記(1)は、母点領域間の隣接関係によって、母点領域間の相対的な位置関係を表す。また、(2)、(3)は、グラフ(道路網)に沿った距離索引である。

これらを、距離と位置関係に基づく空間索引とする。

空間索引を用いて、最短経路探索・範囲問合せなどの距離に基づく探索と範囲に基づく問合せを行うことができる。最短経路探索は、出発地点と目的地点間の途中にある母点領域内での探索を省略することができる。また、範囲問合せでは、問合せ点周辺から、段階的に探索する階層を変化させていくことにより、問合せ点周辺の探索を省略することができる。

[母点領域の大きさの指標] 母点領域の大きさを示す指標として、各階層の各母点領域において、その母点領域を横断する経路の最大経路長と最小経路長の2つの指標を用いる。空間索引木の G_{i_j} での母点領域 j において、境界点間の最短経路の距離を $d(B_{i_j}^n, B_{i_j}^m)$ とする。さらに、すべての境界点の組合せのなかで最大の距離を $\Delta_{i_j}^j$ とする。同様に、すべての境界点の組合せのなかで最小の距離を $\delta_{i_j}^j$ とする(図8)。

つまり

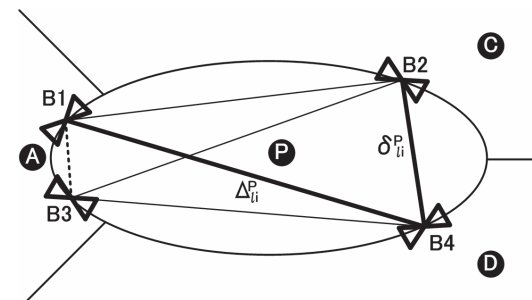


図 8 Δ と δ の定義
Fig. 8 Definition of Δ and δ .

$$\Delta_{li}^j = \max_{n,m} d(B_{li}^n, B_{li}^m) \quad (4)$$

$$\delta_{li}^j = \min_{n,m} d(B_{li}^n, B_{li}^m) \quad (5)$$

と表すことができる。しかし、このとき、2つの境界点在同一の母点の組合せの場合(たとえば、図8の $B1[A:P] \cdot B3[A:P]$)、この場合、 $\Delta_{li}^j \cdot \delta_{li}^j$ を生成する条件に合致しないものとする。これは、 $\Delta_{li}^j \cdot \delta_{li}^j$ とともに、ある領域 A からある領域 P を経由し領域 B に到達するための領域 P における指標であり、領域 P での最大(あるいは最小)の距離ではないことによる。図8はある領域 P に対しての Δ_{li}^j と δ_{li}^j の相関関係を示している。これらは、5.2節で後述する経路探索において目的地点までの最短経路長を見積もる場合に利用する。

3.4 空間索引木の特徴

前節までの手順で得られる提案木は、以下の性質を持つ。

- 節は、原地図の部分領域を表し、領域間の位置関係と距離関係を保持した複数の詳細度別のグラフからなる。
- 枝は領域の包含関係を表し、また、下位の領域の持つ属性を包含された上位の領域へ集約させることが可能になり、地図応用などに有効である。
- 各階層で、ノードは、空間索引として、道路に沿った領域内の複数の横断経路と距離、領域間の隣接関係に基づく位置関係、領域の最大・最小横断距離などを持つ。
- 原地図から、ノードを選択してボトムアップに木を生成しているため、平面上の道路の分布の歪みに依存せず、平衡木の性質を持つ(図6)。すなわち、節の次数がほぼ一定であり、各節が表す領域内の道路片数がほぼ一定となる。このため、道路上の2地点間

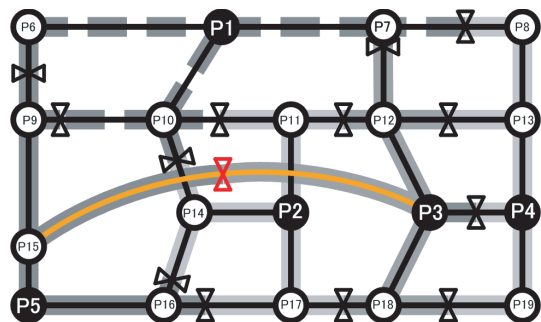


図 9 立体交差を持つ 3 次元グラフ
Fig. 9 3-dimensional graph with two level crossing.

の最短経路を探索する際、提案木の展開ノード数はほぼ均等となる。

3.5 立体交差などの三次元グラフへの適用

道路網に立体交差が存在しても NV 図を生成することができる。例として、図 1 に立体交差として P3 と P15 の間を直接つなぐ経路を追加したときの NV 図を図 9 に示す。母点領域 P3 と P5 が図 1 では、直接隣接しないが、この立体交差の経路により互いに隣接することになり、赤い境界点を新しく持つ。この境界点についても、それぞれの母点境界点間経路と境界点間経路を生成できる。したがって、3.2 節の方法と同様に階層化し、空間索引木を生成できる。

これは、提案空間索引木が、グラフ上の母点領域の隣接関係と距離のみの索引であるためであり、このように立体交差を持つ三次元グラフにも同様に適用できる。

3.6 一方通行などの有向グラフへの適用

図 1 のエッジに方向を追加する (図 10)。矢印は、通過できる方向を示しており、両向きの矢印は、両方向に通過できるエッジを示す。すべてのノードは、他の任意のノードと互いに到達可能としている。

[ノード NV 分割] 有向グラフに対して、ノード NV 分割を行う場合、次の 2 通りの分割が考えられる。

- (1) 母点から所属ノードに到達できる (OUT ノード NV 図)
- (2) 母点に所属ノードから到達できる場合 (IN ノード NV 図)

[エッジ NV 分割] エッジ NV 分割では、(1), (2) のいずれの場合も境界点の定義を変更する必要がある。このとき境界点の位置は、下記の 2 通り考えられる。

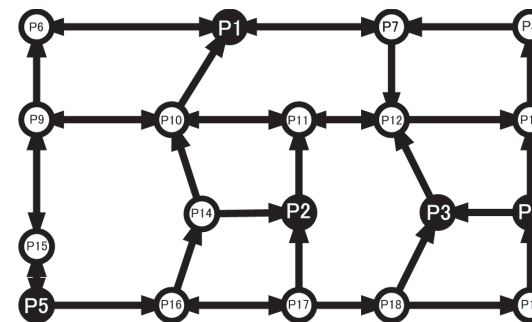


図 10 一方通行を考慮した有向グラフ
Fig. 10 Digraph for one way graph.

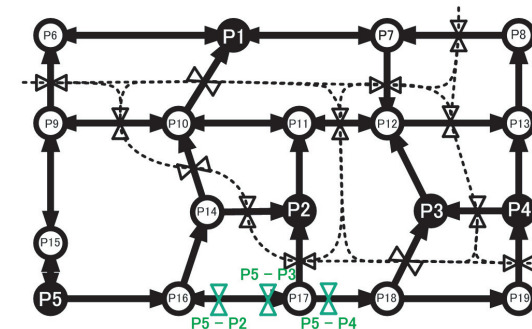


図 11 母点から到達できるノードの NV 分割 (OUTNV)
Fig. 11 NV diagram from generators.

(A) 2 母点間をつなぐ経路の中間

(B) 両端のノードの所属母点が違うエッジ上の中間

(A) を用いると、同じエッジ上に複数の境界点が存在する。たとえば、図 11 での緑の境界点である。これらは、P17 を経路する経路と母点の組合せから決定される境界点の位置を示している。境界点は領域の端を示すため、境界点の位置により領域が重なっていることが分かる (図 12)。

(B) を用いると、上記のように領域の重複は発生せず、エッジ NV 分割を一意に決定できる。ここでは、境界点の位置の定義として (B) を採用する。

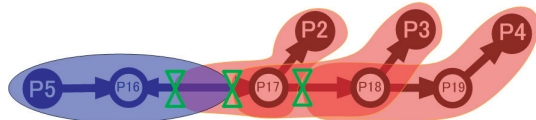


図 12 境界点の位置から発生する母点領域の重複

Fig. 12 Overlapped generator regions caused by boundary definition (A).

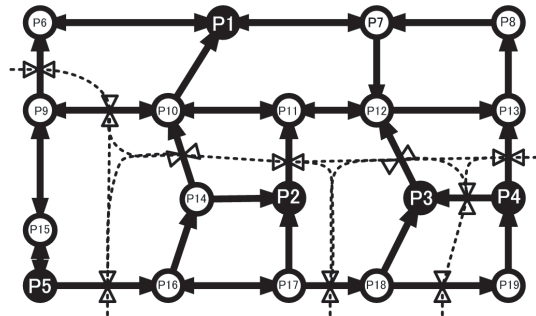


図 13 母点に到達できるノードの NV 分割 (INNV)

Fig. 13 NV diagram to generators.

このとき境界点は、エッジ上に存在するため、向きを持つ必要がある。

また、無向グラフでの境界点の位置は、(A)、(B) の条件をとともに満たす。

無向グラフに対する NV 図 (図 1・図 2) と、母点に到達できるノードに対する INNV 図 (図 13) と、母点から到達できるノードに対する OUTNV 図 (図 11) のそれぞれの母点領域の形は、まったく異なる。

母点境界点間経路は、境界点を持つエッジの向きと途中にある経路により、方向・経路が決定される。図 13 の INNV 図を用いて母点境界点間経路を生成した場合、図 14 が生成できる。図 14 で母点に到達できる境界点からの経路は紫で、母点から到達できる境界点への経路を赤、またどちらでも可能な経路を緑で示している。境界点と母点間で母点領域内を通過して到達できない組合せは、破線で示している。

同様に、図 13 を用いて生成した境界点間経路を図 15 に示す。境界点間で到達できない組合せ (母点領域に進入する 2 つの境界点どうし・母点領域から進出する 2 つの境界点どうし) は表記していない。また進入進出の組合せにおいても、母点領域内のみを通過して到達できないものは、破線で示している。

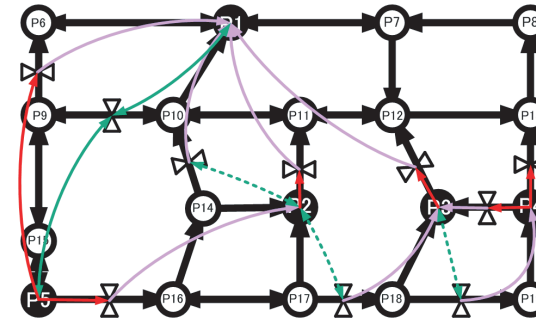


図 14 INNV 図における母点境界点間経路

Fig. 14 Route between generators and boundaries in INNV diagram.

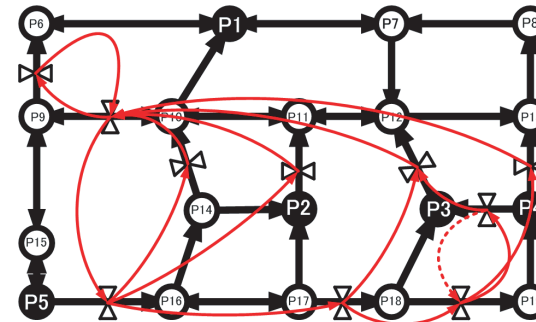


図 15 INNV 図における境界点間経路

Fig. 15 Route between boundaries in INNV diagram.

このように有向グラフに対しても本手法を適用できる。さらに詳細な議論は別稿に譲る。

4. 空間索引木の生成アルゴリズム

3章で述べた工程を木構造の生成の流れとして示すと図 16 となる。本章では空間索引木の生成アルゴリズムについて詳細に述べる。

アルゴリズムでは、各階層において次の 2 つの経路を算出する。

- (1) 母点とすべての境界点の間をつなぐ最短経路
 - (2) すべての 2 つの境界点の組合せに対して、間をつなぐ最短経路
- (1) は、各層での各母点領域の大きさで経路を示し、(2) は、母点領域を横断するすべて

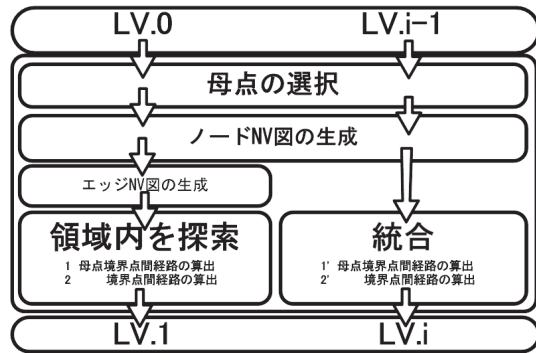


図 16 空間索引木の生成手順

Fig. 16 Generation procedure of spatial index tree.

| | | | | |
|-------|--------------|-------------------------------------------------------------|-----------------|-------------------------------------------------------------------|
| 構造化情報 | Node | NodeID 母点nodeID 母点への 一歩先のnodeID 母点への距離・経路 LV | Edge | EdgeID 母点nodeID-境界点ID or 境界点ID-境界点ID 距離・経路 所属母点 LV |
| | Delta | 母点nodeID Δ δ LV | Boundary | 境界点ID 母点nodeID1 母点nodeID2 母点への距離 LV |
| 原地図 | Node | NodeID X座標 Y座標 | Edge | NodeID1 NodeID2 距離 |

図 17 ノード・エッジ・境界点の保持データ

Fig. 17 Attributes of nodes, edges, boundaries (1).

の経路と距離を示す。

4.1 データ構造

[ノード・エッジ・境界点] 図 17 にノード・エッジ・境界点の保持データを示す。下が与えられた原地図の情報であり、上が空間索引木の構造部分を示す。また図 18 に構造化部分の持ち方を図示する。

Node の情報は、ノード NV 図によって生成できる。Boundary は、エッジ NV 図から生成する。また Edge については、3.2 節に示す方法で生成することができる。

図 18 において、上段の Boundary は、母点の隣接関係と距離を持つ。下段の Edge は、1 つ下の階層での経路集合と、どの母点に所属する Edge を示す。

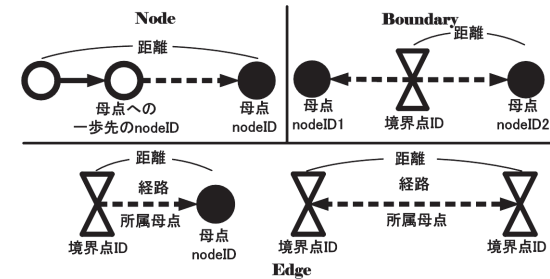


図 18 構造化情報のデータ

Fig. 18 Attributes of nodes, edges, boundaries (2).

Algorithm 1 Procedure Node Network Voronoi Diagram

```

1. for each  $p \in N$ 
2. do  $pX \leftarrow \text{Random}(0,1)$ 
3. if  $pX < K$  then
4.    $V(p) \leftarrow p; D(p) \leftarrow 0; P(p) \leftarrow p$ 
5.   List_h add [p, D(p)]
6. else
7.    $V(p) \leftarrow \text{null}; D(p) \leftarrow \infty; P(p) \leftarrow \text{null}$ 
8. end if
9. end for
10. List_h を初期ノードとしてダイクストラ探索を行う
    
```

$N =$ すべてのNode
 $p =$ Node
 $K =$ 母点選択の閾値
 $V(p) =$ pの所属母点
 $D(p) =$ pから所属母点V(p)までの距離
 $P(p) =$ pから所属母点V(p)への一歩先のnodeID
 $\text{List}_h =$ 展開候補(波頭)リスト [(p, D(p))]

各領域について、領域の大きさを示す $\Delta \cdot \delta$ の算出は、構造化情報部分の Edge を利用する。Edge の所属母点ごとに、最大と最小の経路長を記憶し、最大のものを Δ とし、最小のものを δ とする。

4.2 第 1 層のグラフ生成

[LV.0 から LV.1 の生成法] 原地図 G_{10} に対応する地図を生成するために、すべてのノード集合より、確率 p でランダムに母点を選択し、ノード NV 分割、エッジ NV 分割、経路の生成を行う。

ノード NV 分割より、 G_{10} が選択された母点数に分割される [Algorithm1]。

[境界点の抽出] ノード NV 分割により得られた母点領域間の境界点を求める。境界点はエッジ上に存在するため、エッジの両端のノードの母点を判定することにより、エッジの所属関係と境界点の位置を算出することができる。ここでは、境界点が母点間の中心にあることから 4 行目の計算となる。5 行目はエッジ内での端点からの距離を求めている [Algorithm2]。

1. [母点-境界点間経路の算出法] エッジ NV 分割において、各境界点に対して母点までの

Algorithm 2

Procedure Edge Network Voronoi

1. for each $e \in E$ do
2. $(n1, n2) = e$ // $V(n)$: n の母点 $e.length$: e の持つ距離
3. if $V(n1) \neq V(n2)$ // すなわち、 e が境界辺であるとき
 // 境界点を b とする
4. $V(n1)$ から b への距離 $d1 = [D(n1) + e.length + D(n2)]/2$
5. $n1$ からの b への距離 $d2 = d1 - D(n1)$
6. $V(b) = [V(n1), V(n2)]$; $B.edgeNo = e.No$; **Boundary add** b
7. else // **Boundary**: 境界点集合
8. $V(e) = V(n1)$
9. end if
10. end for

Algorithm 3

Procedure 母点境界点間経路の生成(LV.1)

1. for each $b=(n1, n2) \in \text{Boundary}$ do // $n1, n2$: 境界辺の両端ノード
2. while $n \neq V(b)$

| | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 3. if $P(n) \neq V(n)$ 4. Path.list add n; $n \leftarrow P(n)$ 5. else if $P(n) = V(n)$ 6. Path.list add n; break 7. end if | <ol style="list-style-type: none"> 2. nを $n1, n2$に対して 3. エッジの両端のノードから始め、 4. Pathを生成する 5. ※距離は、境界点算出時の母点から 6. boundaryへの距離と等しい。 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
8. end while
9. end for

距離は算出されているので、境界点と母点間の距離を再度求める必要はなく、境界点を持つノードから母点へノード NV 分割で求められた経路をたどる。

各領域のすべての境界点に対して、母点までの経路 Path.list を求める。このときノード NV 分割時にダイクストラ展開されているノードの母点への1歩先のノード $p(n)$ を利用する [Algorithm3]。

図 19 で B5 から P2 の経路は、B5 を持つノード P11 から P2 までの経路である。ただし、一般には、複数のノードをわたって Path.list を生成する。

2. [境界点間経路の算出法] 各母点領域において、すべての境界点から他のすべての境界点に対して、最短経路を求める [Algorithm4]。

図 20 で B12 と B5 間の経路を求める場合は、B12 を持つ P17 と B5 を持つ P11 間の経路を求めればよい。ここでは、P17 から P11 にダイクストラ法で経路を求める。同様に B6, B8, B10, B11 に対しても境界点間の経路を求める。

4.3 第 2 層以上の上位層 (第 i 層) のグラフ生成
[LV.i-1 から LV.i の生成方法] LV.i-1 での母点集合からランダムに母点を選択し、ノード

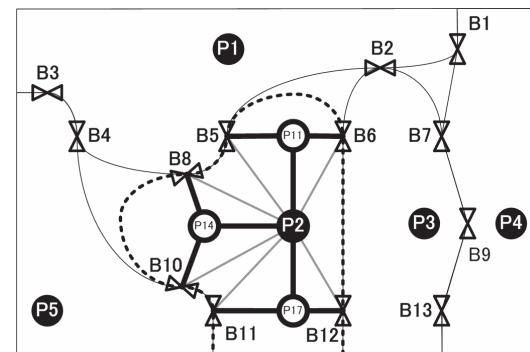


図 19 母点-境界点間経路の算出 (Level1)
Fig. 19 Route generation between generator P2 and boundaries (Level1).

Algorithm 4

Procedure 境界点間経路の生成(LV.1)

1. for each $v \in V$ // V : 全ての母点集合
2. for each $n \in N$ $e \in E$ $b \in \text{Boundary}$

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none"> 3. NodeList add n if $V(n) == v$ 4. EdgeList add e if $V(e) == v$ 5. BoundaryList add b if $V(b) == [v, *]$ or $[*, v]$ | <ol style="list-style-type: none"> 2. vの母点領域に含まれる要素 (ノード・エッジ・境界点) の取り出し |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
6. end for
7. for each $b \in \text{BoundaryList}$
8. NodeList EdgeList のみを用いて
9. BoundaryList に含まれる b 以外の他の境界点に経路を求める
10. end for
11. end for

NV 分割を行い、各ノードの母点所属を決定する。これにより、各ノードの統合する領域は、一意に決定される。

第 i 層のノードを構成するために第 i-1 層のノードの持つ領域である部分領域の統合が必要である。統合処理の結果、2つの領域で共通の境界点が隠れるため、その境界点を經由する経路のつなぎ換え処理が必要となる。図 5 では、P3 に P1, P2, P4 が統合されるため隠れる境界点が A, B, C である。

図 16 での統合においては、次の 2つの処理を行う。母点-境界点間経路の算出 (1') と境界点間経路の算出 (2') である。

1'. [母点-境界点間経路の算出法] 各境界点から母点までの経路を算出する。第 1 層と異なり上位層の生成には、領域の統合が必要である。3.2 節 [領域の統合] に従って、図 21 で P3 を主とし P2 を統合する場合について述べる。図 5 での B における処理にあたる。

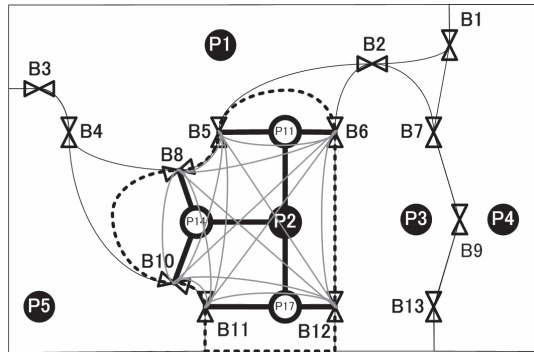


図 20 境界点経路の算出 (Level1)
Fig. 20 Route generation between boundaries (Level1).

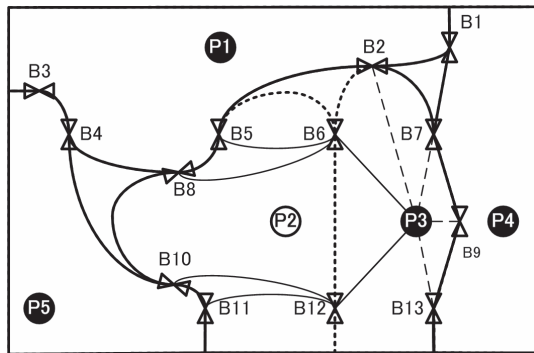


図 21 母点領域の統合 (1') (Level2)
Fig. 21 Route generation from P3 to new boundaries in region merging (Level2).

この統合で、境界点 B5, B8, B10, B11 が新たに P3 の母点領域に追加され、境界点 B2, B7, B9, B13 は、そのまま維持し、境界点 B6, B12 は、境界点ではなくなる。

統合で新しく G_{i2} で P3 の境界点になる B5 について見ると、母点 P3 までの経路を求めるには、母点領域 P2 と P3 の境界点 B6, B12 を取り出し、それらを経由する母点 P3 までの最小の距離を持つ経路を採用する。

$$Path(B_{i2}^5, P_{i2}^3) = Path(B_{i1}^5, B_{i1}^6) + Path(B_{i1}^6, P_{i1}^3) \quad (6)$$

$$Path(B_{i2}^5, P_{i2}^3) = Path(B_{i1}^5, B_{i1}^{12}) + Path(B_{i1}^{12}, P_{i1}^3) \quad (7)$$

```

Algorithm 5 統合
Procedure //主たる領域をPi 従たる領域をPjとする
1. T = { Bj | Pi, Pjを母点にもつ境界点 } //消える境界点
2. for each Bj ∈ [Pjの境界点集合]-T //新規境界点
3.   find Bk ∈ T
4.   that minimizes |Path(Bj,Bk)|+|Path(Bk,Pi)|
5. end for
6. T = { Bj | Pi, Pjを母点にもつ境界点 } //消える境界点
7. for each Bj ∈ [Pjの境界点集合]-T //新規境界点
8.   for each Bi ∈ [Piの境界点集合]-T
9.     find Bk ∈ T
10.    that minimizes |Path(Bj,Bk)|+|Path(Bk,Bi)|
11.  end for
12.end for
    
```

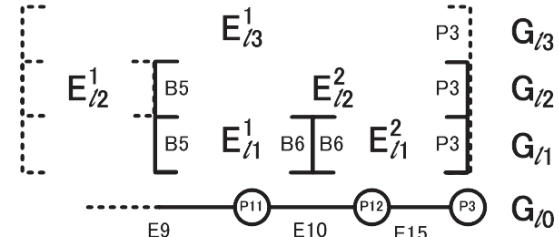


図 22 エッジの包含関係
Fig. 22 Merge edges.

この場合、 G_{i1} で求めた経路 B5-B6 と、経路 B6-P3 をあわせて経路 B5-B6-P3 とし、同様に経路 B5-B12-P3 も生成し、この2つの経路長を比較して、短い方を採用する [Algorithm5 前半部]。

図 22 は、各階層での境界点とエッジに着目した図である。たとえば、 E_{i2}^2 は、 E_{i1}^1 と E_{i1}^2 のエッジから構成されている。

図 23 は、リストで格納された経路の操作手順を示す。e1, e2 を連結した経路 e5 と e3, e4 を連結した経路 e6 を生成する。e5, e6 を比較して短い方の経路 e5 を選択する。

統合において、P3 が統合前から持つ境界点 (B2, B13 など) については、新たに母点-境界点間経路を計算せず、そのまま当該層での境界点とする。

2'. [境界点間経路の算出法] すべての境界点間の経路を求める必要がある。

統合する領域 P3 に追加される新規境界点 B5, B8, B10, B11 から、P3 が統合前に元々持つ境界点 B2, B7, B9, B13 への経路を求める計算が必要である。

たとえば新規境界点 B5 と B7 をつなぐ経路の場合は B6, B12 を通る経路があり、それ

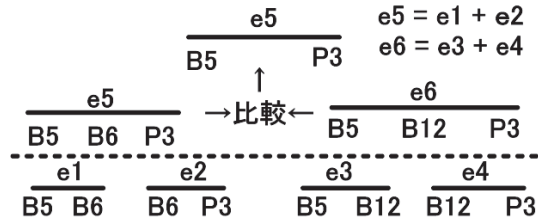


図 23 B5-P3をつなぐ経路の生成

Fig. 23 Compare route costs in Lv.1 for generating a route in Lv.2.

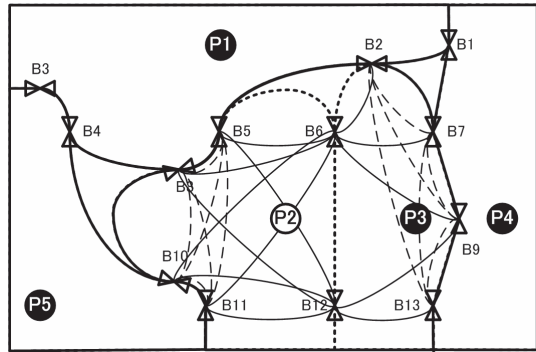


図 24 母点領域の統合 (2') (Level2)

Fig. 24 Route generation among new combination of boundaries in region merging (Level2).

それぞれの経路は以下の式で表すことができる。

$$Path(B_{i2}^5, B_{i2}^7) = Path(B_{i1}^5, B_{i1}^6) + Path(B_{i1}^6, B_{i1}^7) \quad (8)$$

$$Path(B_{i2}^5, B_{i2}^7) = Path(B_{i1}^5, B_{i1}^{12}) + Path(B_{i1}^{12}, B_{i1}^7) \quad (9)$$

上記の経路から短い方の経路を B5・B7 間の経路とする。他の新規境界点 B8, B10, B11 に対しても同様に最短経路を求める [Algorithm5 後半部]。

統合前と統合後で同じ境界点の組合せ間の経路は計算しなくてよい。図 24 で、左の境界点 B5, B8, B10, B11 相互の境界点間経路 (左の破線部) と、右の境界点 B2, B7, B9, B13 相互の境界点間経路 (右の破線部) にあたる。

[注] 上記の処理は新たに統合された領域を上位層で 1 つの領域として扱うための処理・準備である。また 3 つ以上の領域が統合する場合は、母点となる母点領域に接している領域が

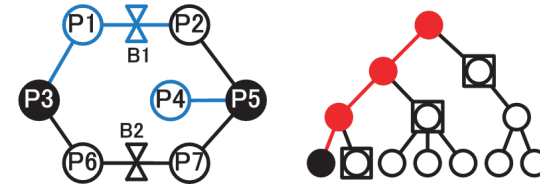


図 25 原地図更新に対する空間索引木の更新

Fig. 25 Modification of spatial index tree for updation of base map data.

ら、順に母点領域を拡大していく。統合の順序は任意に決定してよいが、統合の結果は一意に決定される。

4.4 空間索引木の更新

原地図の更新は、道路片や交差点の追加・削除の場合が考えられる。このとき、追加・削除のいずれの場合も、ノード NV 分割を実行して、空間索引木の修正する部分を見つける必要がある。ノード NV 分割の母点として与えるノードは、更新前の G_{i1} でのノード集合である。このとき次の状況がある。

状況 1 ノード削除により母点ノードが消失する (図 25 左 P3 削除の場合)

これは、ノード NV 分割する以前に確認できる。次に、NV 分割を実行した結果として、次の 4 通りの状況が考えられる。

状況 2 ノードエッジの追加により境界点が発生する (図 25 左 P1 追加の場合)

状況 3 ノードエッジの追加により境界点が発生しない (図 25 左 P4 追加の場合)

状況 4 ノードエッジの削除により境界点消失する (図 25 左 P1 削除の場合)

状況 5 ノードエッジの削除により境界点消失しない (図 25 左 P4 削除の場合)

5 つのパターンで、母点ノードが消失する場合 (状況 1) は、さらに 2 つのパターンに分けられる。レベル 1 で母点が所属ノードを持つ場合 (図 26 左) と、所属ノードを持たない場合 (図 26 右) がある。これら 2 つの場合で処理が異なる。図 26 では、削除されるものを黒破線で、変更されるものを赤で示している。

前者の場合 (図 26 左) では、母点が消失したとしても、その領域のすべての情報は、引き続き保持される必要がある。このとき、新たに代替の母点を所属するノードの中から選択し、レベル 1 での母点境界点間経路を更新する。また境界点間経路も削除される母点を經由していた場合はつなぎ直す。それより上の層でも、母点境界点間経路と境界点間経路をつなぎ直す必要がある。この場合の更新範囲は、図 25 右の赤丸である。

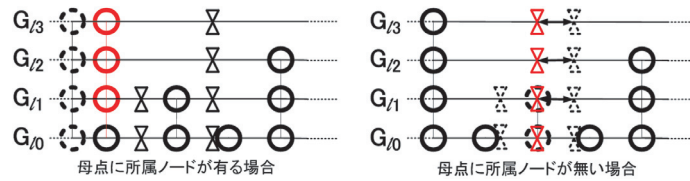


図 26 母点削除による空間索引木の更新範囲

Fig. 26 Updation scope in spatial index tree due to generator removal.

後者の場合（図 26 右）では，前者の場合のように母点領域の情報の更新が行えないため，空間索引木全体の再構成を行う必要がある．これは以下の理由による．更新の際に，削除される母点領域を無作為に隣接母点領域に統合すると，各層において境界点の位置にずれができ，正しい経路探索を行えない．また隣接領域情報も大きく変化するのでその更新も必要となり，最初から空間索引木を構成する手間と変わらないからである．

しかし，後者の場合は，母点の選択確率にも依存するが，実用上では，限りなく起こりたいたいと考えられる．

残りの 4 つの中で境界点が発生しない場合（状況 3）・消失しない場合（状況 5）では，NV 分割の結果により修正される領域を木構造に沿って根方向に向かって更新する．このとき更新されるのは，図 17 での G_{11} での node 情報と更新される範囲での母点境界点間経路と境界点間経路の経路情報となる．図 25 右で，黒丸が更新された領域とすると，赤丸の部分に影響を及ぼす．

境界点が発生する場合（状況 2）・消失する場合（状況 4）では，境界点は，隣接する領域にも影響するので，修正する範囲は隣接する領域（図 25 右 領域）も含み，順次階層を上って，その上位層でも隣接する領域を更新する．隣接しない領域は更新する必要がない．該当する領域を図 17 での構造化情報の部分を更新する必要がある．

5. 空間索引木を用いた最短経路探索

ここでは，空間索引木を用いた 2 つの最短経路探索手法について提案する．

1 つ目は，木の階層性と領域の包含関係を用いて，始点と終点の間にある領域の統合を繰り返して，経路を絞っていく手法である．

2 つ目は，各母点領域での距離見積りを用いて，領域の隣接関係を求め，探索候補を絞り，本探索の高速化を行う手法である．

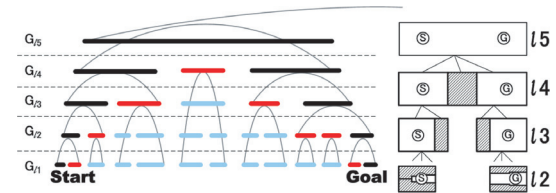


図 27 節の包含関係に着目した木構造

Fig. 27 Tree with respect to inclusion relations among generator regions.

5.1 部分木の抽出

まず両探索法において必要となる，部分木の抽出を行う．部分木の抽出は以下の工程によりできる．

1. [始点と終点をともに含む母点領域の決定] 始点と終点が，原地図上のノードとして与えられたとする．

空間索引木において，始点と終点をともに含む葉に近い節（母点領域）を，包含関係を用いて，葉から順に上方に探索する．その結果，始点と終点が同一の母点に含まれる階層（第 i 層とする）と節（母点領域）が判明する．

図 27 は， G_{15} 層のあるグラフが指定した始点と終点をともに含む領域であることを示す．

2. [母点の各階層での所属関係の抽出] 発見した第 i 層での始点と終点が同一となる節（母点領域）より，第 $i-1$ 層での母点の所属関係を抽出する．始点と終点を含む母点は，所属関係の抽出を繰り返して，含まないノード（図 27 右網掛け部）は，母点領域の構成情報を取り出す．

これにより，空間索引木の部分木を限定する．この木を T_0 とする（図 27）．図 27 中の赤線の部分が図 28 での各領域にあたる．図 27 の G_{14} の赤線部分が，図 28 での 5_{14} にあたり， G_{13} の赤線部分が， 4_{13} と 6_{13} にあたる．

5.2 統合を用いた最短経路探索

4 つの工程に分けることができる．

1. [始点と終点をともに含む母点領域の決定] 5.1 節の部分木の抽出を行う．
2. [始点終点の領域以外の探索] T_0 は G_{14} で複数の節（母点領域） $\{5_{14},$ 始点を含む母点領域，終点を含む母点領域 $\}$ から構成されている．そのうち，図 28 では 1 つの領域 5_{14} となって始点と終点を含む母点領域以外の部分があればそれらを統合する．複数ある場合でも一意に統合することができる．

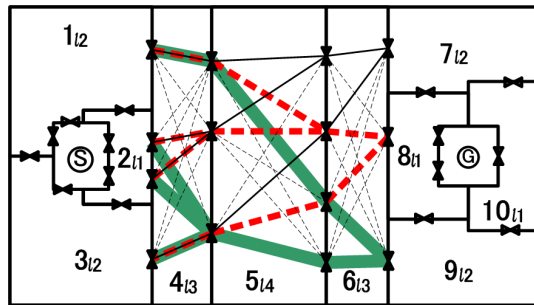


図 28 統合を用いた最短経路探索
Fig. 28 Shortest path search by use of subregions integration.

このとき得られた領域の境界点間経路の中で、始点を持つ母点領域の境界点から、終点を持つ母点領域の境界点を結ぶ経路集合を解候補経路の集合と呼ぶ。統合により、境界点間経路は一意に決定され、図 28 中の破線部分の経路は見えなくなり、赤 4 本、黒 4 本、計 12 本の線が解候補経路となっている。

3. [逐次的差分探索] T_0 の階層を 1 つ下げ、始点と終点を含む母点領域のそれぞれのサイズを縮小し、始点と終点を含まない母点領域の部分を統合して拡大する。上位層と下位層での、始点と終点を含む母点領域の差の部分（図 27 の G_{l3} 赤線）と上位層で統合した領域（図 27 の G_{l3} 青線）を統合し、下位層での解候補経路を生成する（差分探索）。 G_{l3} での解候補経路を図 28 中 赤・黒・緑実線で示す。

以下、逐次的に階層を下げ、 G_{l1} まで解経路候補を生成する。

4. [G_{l0} での探索] 第 1 層での解候補経路の生成が終了したとき、構造化情報を利用できなくなるため、始点と終点を含む母点領域の境界点まで、始点と終点から、 G_{l0} （原地図）上で経路探索を行う。

図 27 G_{l1} での黒線の部分にあたり、図 28 中で S と G を含む図上での最小の矩形領域にあたる部分を探索する。

[結果の合成] G_{l0} での探索の結果と G_{l1} での経路候補を合計し、その組合せの中から、重みが最小となる経路の組合せを選択することで、最短経路を求めることができる。

[例外] 階層を下げていくうちに、始点終点のいずれかが母点となった場合、その母点領域では、レベルを 1 つ下げ、母点領域を縮小させ、できた差分の部分を上位レベルでの経路候補と統合する必要はない。

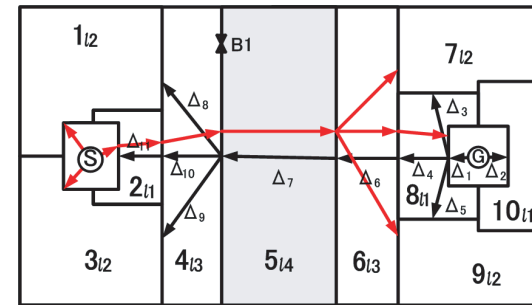


図 29 見積りを用いた最短経路探索
Fig. 29 Shortest path search by use of distance estimation.

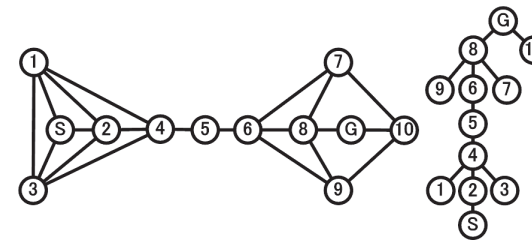


図 30 T_0 から抽出した領域の隣接関係と大きさの指標を用いて探索した木構造
Fig. 30 Adjacent relations of subregions extracted from T_0 and tree structure by using spatial index Δ .

これは、以下の理由による。各母点領域は、母点-境界点間経路を所持し、この経路は、生成時の条件付けより最短である。ともに母点となった場合、母点-境界点間経路と、間にある解候補経路を合計し、重みが最小となる経路を最短経路とすればよい。

5.3 距離見積りを利用した最短経路探索

[事前探索]

1. [領域部分木の抽出] 5.1 節により、部分木の抽出を行う。始点終点を含まない領域（図 27 中の赤線）をすべての階層で取り出したとき、それらを境界点で連結することができ、 G_{l1} での始点と終点を含む領域を追加すると、一枚の平面図として得ることができる（図 29）。各領域をノードとして、表すと図 30 左のグラフが得られる。各ノードの番号は、図 29 の領域番号に対応している。

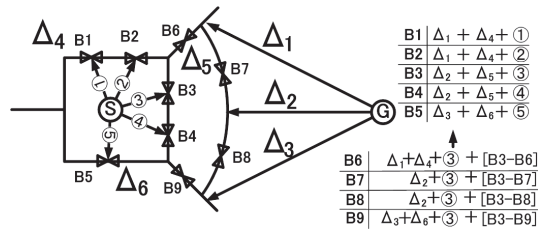


図 31 始点の領域での経路の置き換え
Fig. 31 Element replacement of path list in the start region.

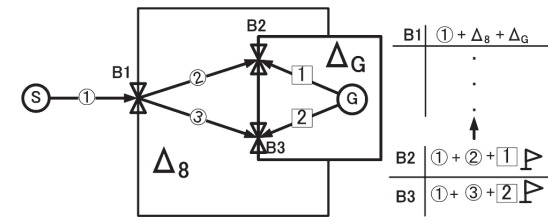


図 32 終点の領域での経路の置き換え
Fig. 32 Element replacement of path list in the goal region.

2. [大きさの指標を用いたゴールからのダイクストラ探索] 1 で得られたグラフに対して終点を含む領域からすべてのノードに対して、各ノードの大きさの指標の値を用いてダイクストラ探索法を行って、目的地を根ノードとした木構造を構成する (図 30 右). 図 29 では、黒矢印で構成された木構造に対応する .

大きさの指標の値を用いてダイクストラ探索を実行することにより、各領域は、終点を持つ母点領域までの経路と見積りの距離を保持することができる . たとえば、ノード 3 (平面図では 3_{12} に対応) に対しては、目的地 G までの見積り経路 4, 5, 6, 8 と見積り距離 Δ_9 , Δ_7 , Δ_6 , Δ_4 , Δ_1 を持つ .

3. [G から G の領域境界点までの距離の算出] 終点から終点の母点領域の持つ境界点に対して、経路探索を行う . この作業は、G の位置を原地図から図 29 に母点と同様に扱うための作業である . 終点が母点の場合、既知であるので探索を行わない .

[本探索]

次に、事前探索によって求められた始点から終点の探索木 (図 30 右) を用いて始点 S から終点 G に向かい、領域をつなぐ経路を大きさの指標を用いた経路から、境界点の情報をを用いた経路に置き換えていく .

図 29 の S と G を含む領域を拡大したものが図 31 と図 32 である .

たとえば、図 31 において、事前探索の結果、S を含む領域と隣接する領域への見積り距離 $\Delta_1 - \Delta_3$ までと、隣接領域の大きさの指標 $\Delta_1 - \Delta_6$ が求められている . S を含む領域の境界点 B1 から B5 を経由したときの G までの距離は、S を含む領域の境界点までの実測距離と、境界点までの見積り距離の和となる (図 31 右表) .

このため、以下の工程を行う .

1. [S から S の領域境界点までの距離の算出] 図 31 で境界点 B1 から B5 までの経路と距

離を算出する . 始点が母点の場合、既知であるので探索を行う必要がない .

2. [境界点での距離計算] 境界点ごとに見積り距離と実測距離を計算しリスト化する (図 31 右表) .

- (1) リスト中で最も見積り距離と実測距離の合計が少ない境界点を決定する .
- (2) リストから対象の境界点を削除する .
- (3) 対象の境界点から、展開できる境界点を展開する .
- (4) 展開した境界点の見積り距離と実測距離をリストに追加する .

これを繰り返すことで目的地の領域の境界点まで探索できる .

たとえば、図 31 右では B3 が展開された場合、B6 から B9 が新規にリストに追加され、同時にリストから B3 が削除される .

3. [G 領域での Δ と実測値の置換] 本探索において、探索が目的地を含む境界点まで到達した場合、実測値と事前探索の 3 で求めた値を合計してリストに追加する . この目的地到達フラグを持った経路が、リストから抽出されたとき経路探索を終了し、解経路とする .

[特徴]

[1] 図 30 右の木構造の構成は、目的地までの概略の経路を与えると同時に、その領域を通ったときの目的地までの距離の見積りを与える .

[2] 図 30 左はマルチグラフの経路部分を 1 本に縮退させているので、事前探索・本探索のいずれにおいてもリストの展開数を減少させる効果がある .

6. 評 価

6.1 空間索引木の性質

シミュレーションにより空間索引木の生成処理について評価する . 生成する原地図として

25 道路網応用のための空間索引木の提案と最短経路探索への応用

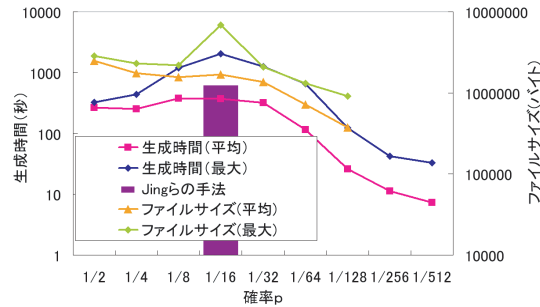


図 33 空間索引木の母点選択確率 p に対する生成時間・ファイルサイズ (1,612 ノード)

Fig. 33 Computation time and file size for spatial index tree T to selection probability p (1,612 nodes).

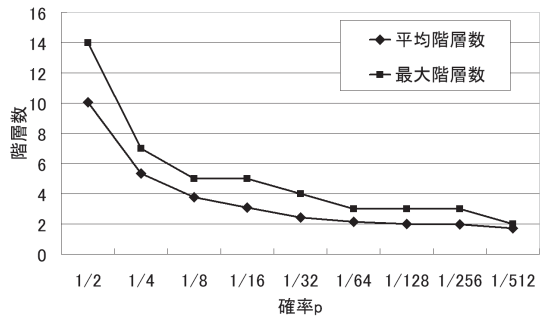


図 34 空間索引木の選択確率 p における階層数 (1,612 ノード)

Fig. 34 Number of layers for spatial index tree T to selection probability p (1,612 nodes).

国土地理院数値地図 2,500 (空間データ基盤, 1/2,500 縮尺)¹⁵⁾ の大阪府高槻市街中心部を利用する。この地図は、ノード数が 1,612 個、エッジ数は 2,077 本で構成されている。

利用したマシンは、WindowsXP, Xeon 3.72 GHz × 2, メモリ 2 GB である。すべてのノードが 1 つの領域に含まれる階層までの生成を 1 回とし、その試行を各確率において 100 回試行した。確率 p を 1/2 から 1/512 まで変化させた。

評価対象について：提案手法と比較するため、A*経路探索アルゴリズムと Jing ら^{9),10)} の手法を簡単にしたものを利用する。

A*経路探索アルゴリズムでは、ヒューリスティック関数として、求める地点の座標を (X_α, Y_α) ・目的地の座標を (X_G, Y_G) に対して、下記の数式で求めた距離を使う。

$$\sqrt{(X_G - X_\alpha)^2 + (Y_G - Y_\alpha)^2} \quad (10)$$

Jing らの手法として、空間をノードの位置座標を用いて k-d 分割し、分割された領域内のすべてのノード間の 2 地点間探索で生成した経路ビューを用いる。また領域間をまたぐ経路は、そのまま保持する。2 地点間の探索には、前記の A*経路探索アルゴリズムを利用する。各領域で保持するノードは 20 前後とした。

生成時間：空間索引木の生成時間に関する母点選択確率 p の影響の評価として、生成時間を計測した。

計測結果の平均値と最大値を図 33 左軸に示す。図より、平均生成時間は 1,612 ノードに対して、500 秒あたりに抑えられることが分かる。最大の生成時間を持つ p = 1/16 のとき

の最大生成時間は 2,055 秒であった。この生成時間の変動は、各ノードが、上位層でも存在するか否かを、確率で選択していることに由来する。

Jing らの手法は、提案手法で 1 領域あたりのノード数が近い p = 1/16 の場合よりも 100 回生成した平均が悪かった。

各確率における生成ファイルの実データサイズ：空間索引木における木構造部分のファイルサイズから、母点選択確率 p に対する空間索引木の構成を見るために、図 33 右軸に生成されたファイルの合計ファイルサイズを示す。

各確率での平均値と最大値を示している。生成時間と関係なく緩やかに平均ファイル容量は減少している。ノードが母点となる確率が大きいときは、階層数が多く、2 境界点間をつなぐ経路が多いためこのようなグラフになると考えられる。最大値は、ファイルサイズ・生成時間ともに 1/16 が最大を示しているが、これは、境界点間経路を生成する数が多かったためにこうなったとみられる。

空間索引木の高さ：空間索引木の階層数に関する母点選択確率 p の影響の評価として、空間索引木の階層数を計測した。

計測結果の平均値と最大値を図 34 に示す。図から見てとれるように、階層数は、 $\log_p N$ (N: ノード数) に平均して収まっていることが分かる。また最大階層数も平均と比較しても大きく変化しない。

各階層における領域の広さ：空間索引木の各階層における領域の広さを示す目安として、あるサンプルにおいて各階層ごとに Δ を平均したものを図 35 に示す。

1/2 の 6 層目と 1/4 の 3 層目が大体同じ値を示している。このことから、母点になる確

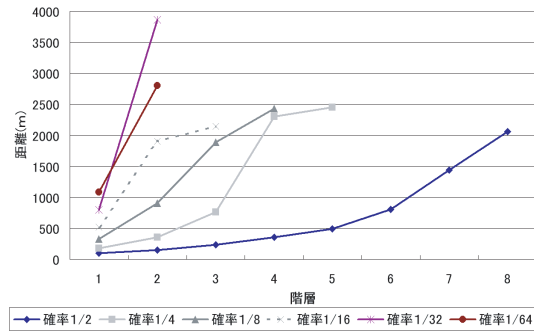


図 35 選択確率 p において空間索引 Δ を用いた領域の大きさの比較
Fig. 35 Region size in spatial index Δ to selection probability p .

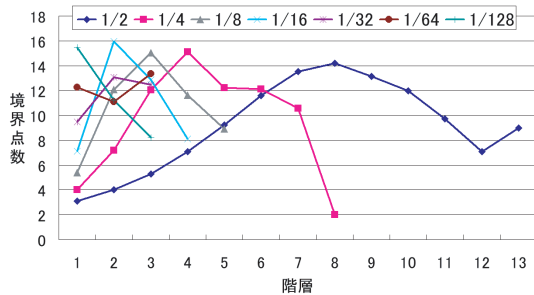


図 36 選択確率 p の各母点領域における境界点数
Fig. 36 Number of boundary in generator regions to selection probability p .

率が大きいと 1 つの領域のサイズが小さくなり、母点選択確率が、 $1/2$ になるとある領域のサイズは、約 2 倍の大きさになる。

各領域が持つ境界点：各領域が持つ境界点の数から、各領域が持つ母点境界点間経路と境界点間経路の数を計測した結果を図 36 に示す。

各確率各階層において最大で大体 1 つの母点領域は、16 の境界点を持つ。このとき、母点境界点間経路は 16 本であるし、2 境界点間経路は、 16^2 本となる。確率 $1/2$ で実行したときのある結果では、すべての階層すべての生成経路を合計した結果は、45,193 本であった。これはすべての 2 ノード間の最短経路を生成した場合に発生する $1,612^2$ の経路より大

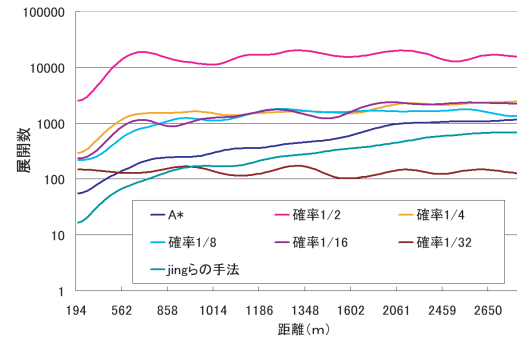


図 37 統合型経路探索と評価対象アルゴリズムの計算量の比較 (1,612 ノード)
Fig. 37 Number of list expansion in shortest path search by region integration method, A* search algorithm and path view model (1,612 nodes).

幅に少ない。

6.2 最短経路探索

統合型経路探索：確率 p を $1/2$ から $1/32$ まで変化させ、提案空間索引木を 1 度ずつ構成する。構成できた木構造において原地図にあるノードからランダムに選択した始点と終点の組合せ最短経路を求めた。このとき、すべての階層で、始点の所属する母点領域の境界点と、終点の所属する母点領域の境界点の組合せの合計を、距離順にならべたものを図 37 に示す。

各確率において距離の長さにかかわらず、組合せの合計がそれほど変化しないことが分かる。A*経路探索アルゴリズムを利用して探索した場合と比較して、A*経路探索アルゴリズムでは、距離が遠くなるにつれ、展開ノード数が増大しているが、提案探索手法では増大しない。提案探索手法は長距離になるにつれ、A*探索アルゴリズムよりも効率的であると考えられる。

また Jing らの手法と比較してみると、A*探索アルゴリズムと同様に距離に対して展開数が増加しているが A*探索アルゴリズムほどではない。また長距離になるにつれ、増大しているの、よりノード数が増えると統合型経路探索が効率的になると考えられる。

見積り型経路探索：統合型経路探索と同じように展開数について同様のグラフを図 38 に示す。同じ A*経路探索アルゴリズム・jing らの手法の動きに対して見積り型経路探索は、統合型経路探索と同様に、距離にかかわらず展開数が一定であるが。また統合型経路探索と比較しても、低い展開数で処理されていることが分かる。確率が小さくなるにつれ、展開数

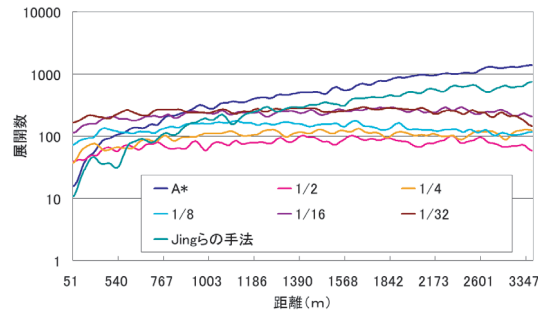


図 38 見積り型経路探索と評価対象アルゴリズムの展開数の比較 (1,612 ノード)

Fig. 38 Number of list expansion in shortest path search by distance estimation method, A* search algorithm and path view model (1,612 nodes).

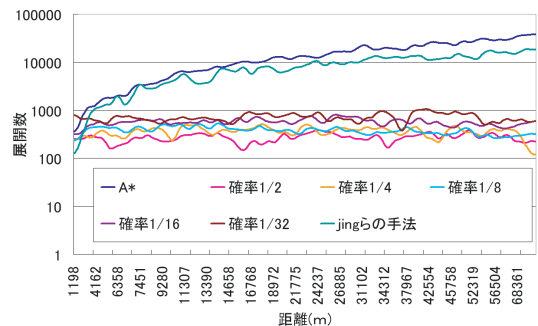


図 39 見積り型経路探索と評価対象アルゴリズムの比較 (40,323 ノード)

Fig. 39 Number of list expansion in shortest path search by distance estimation method, A* search algorithm and path view model (40,323 nodes).

が増大している。これは、原地図でダイクストラ探索を行う領域の拡大により大きくなっているものと考えられる。

大規模地図への適用：次に、ノード数 40,323・エッジ数 54,234 (和歌山市周辺)での結果を見る。ここまでの評価で、見積り型経路探索と統合型経路探索では、見積り型の方が有効だったのでここでは、見積り型を利用し評価を行う。

図 39 に各確率において生成したファイルを使い、探索を実行した結果を示す。距離が増大するにつれ、A*経路探索アルゴリズムは、遠距離になるにつれ増大していき最終的には、

展開数が 1 万を超えた。jingra の手法では、A*探索手法と比較して改善されている。しかし、遠距離になると A*探索手法と同様に増大している。見積り型経路探索では、確率ごとに平均的な展開数でおさまっていることが確認できる。

7. おわりに

本稿では、原地図に対して空間索引木の構成方法を提案し、最短経路探索への応用について議論した。空間索引木は、境界点を結ぶエッジの包含関係と長さに基づいて生成される特徴を持つため、木の生成と最短経路探索が境界点エッジのリスト統合操作に帰着されることを示した。同時にシミュレーションにおいて提案手法の評価を行った。

今後の課題として、本稿で示した有向グラフへの適用法の大規模地図での総合的な数値評価などによる有効性の確認、空間索引木の並列生成法の検討、最短経路探索以外の様々な探索での応用、アドホックな環境を想定した空間索引木の生成と利用などが考えられる。

謝辞 貴重なご指摘を賜った査読者に衷心より感謝の意を表す。

参考文献

- 1) Samet, H.: *Foundations of Multidimensional and Metric Data Structures*, Morgan Kaufmann (2006).
- 2) Sankaranarayanan, J., Alborzi, H. and Samet, H.: Efficient Query Processing on Spatial Networks, *Proc. 13th ACM International Workshop on Geographic Information Systems*, pp.200–209 (2005).
- 3) Kriegel, H.-P., Kroger, P., Kunath, P., Renz, M. and Schmidt, T.: Proximity Queries in Large Traffic Networks, *Proc. 15th ACM International Symposium on Advances in Geographic Information Systems* (2007).
- 4) Hu, H., Lee, D.L. and Lee, V.C.S.: Distance Indexing on Road Networks, *Proc. 32nd International Conference on Very Large Data Bases*, Vol.06, pp.894–895 (2006).
- 5) Kolahdouzan, M. and Shahabi, C.: Voronoi-based K Nearest Neighbor Search for Spatial Network Databases, *Proc. 30th International Conference on Very Large Data Bases*, pp.849–851 (2004).
- 6) Erwig, M.: The Graph Voronoi Diagram with Applications, *Networks*, Vol.36, No.3, pp.156–163 (2000).
- 7) Okabe, A., Boots, B., Sugihara, K. and Chin, S.N.: *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd edition, pp.218–224, John Wiley & Sons Ltd. (2000).
- 8) Graf, M. and Winter, S.: Network Voronoi Diagrams, *Österreichische Zeitschrift*

für Vermessung und Geoinformation, Vol.91, No.3, pp.166–174 (2003).

- 9) Jing, N., Huang, Y.-W. and Rundensteiner, E.A.: Hierarchical Encoded Path Views for Path Query Processing: An Optimal Model and Its Performance Evaluation, *IEEE Trans. Knowledge and Data Engineering*, Vol.10, No.3, pp.409–432 (1998).
- 10) Huang, Y.-W., Jing, N. and Rundensteiner, E.A.: Effective Graph Clustering for Path Queries in Digital Map Databases, *Proc. 5th International Conference on Information and Knowledge Management (CIKM'96)*, pp.215–222 (1996).
- 11) Pugh, W.: Skip Lists: A Probabilistic Alternative to Balanced Trees, *Comm. ACM*, Vol.33, No.6, pp.668–676 (1990).
- 12) Hjaltason, G.R. and Samet, H.: Properties of Embedding Methods for Similarity Searching in Metric Spaces, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.25, No.5, pp.530–549 (2003).
- 13) Eppstein, D., Goodrich, M.T. and Sun, J.Z.: The Skip Quadtree: A Simple Dynamic Data Structure for Multidimensional Data, *Proc. 21st ACM Symposium on Computational Geometry*, pp.296–305 (2005).
- 14) Gold, C.M. and Angel, P.: Voronoi Hierarchies, *GIScience 2006*, pp.99–111 (2006).
- 15) 国土地理院：数値地図（空間データ基盤）. <http://sdf.gsi.go.jp/>

(平成 20 年 12 月 19 日受付)

(平成 21 年 4 月 13 日採録)

(担当編集委員 角谷 和俊)



蒲原 智也（学生会員）

昭和 57 年生．平成 18 年阪南大学経営情報学部卒業．平成 20 年関西大学総合情報学研究科博士前期課程修了．現在，博士後期課程在学中．経路探索手法に関する研究に従事．日本データベース学会学生会員．



上島 紳一（正会員）

昭和 30 年生．昭和 53 年京都大学工学部数理工学科卒業．昭和 58 年同大学大学院工学研究科博士課程単位取得退学（京都大学工学博士）．現在，関西大学総合情報学部教授．マルチメディア情報システム，柔軟な情報ベースに関する研究に従事．電子情報通信学会，ACM，IEEE 等の会員．