ネットワークボロノイ図を用いた領域分割木の提案と評価

蒲原　智也† 　大西　真晶†† 　上島　紳一†

† , †† 関西大学大学院総合情報学研究科
569-1095 大阪府高槻市霊仙寺町 2-1-1
E-mail: †{fb6m125,ueshima}@edu.kansai-u.ac.jp, ††ohnishim@nict.go.jp

あらまし　本稿では，道路網をグラフと捉え, ネットワークボロノイ図を用いて領域分割木の確率的な生成法とそれを用いた経路探査法を提案する．提案手法では，与えられたグラフに対して, ネットワークボロノイ分割を行うことで, 部分グラフに分割し, 全体領域を部分領域に分割する．次に, 各部分グラフをノードと見て, 隣接する部分グラフ同士を融合することにより, 各領域の大きさを拡張しながら, より広い部分領域のグラフを構成する．この操作を繰り返して, 階層化することで, 上位階層の領域間の包含関係を満たす領域分割木をボトムアップに生成することができる．続いて, 領域分割木を用いて, 再帰的な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, 領域分割木の持つ性質や確率的な母点選択に対する生成時間などについて検討し, 国土地理院道路網データに実際に適用して, 提案アルゴリズムを用いた領域分割木の定量的な評価を行う．提案手法により, ディジタル道路情報を有効に利用して, 道路網の密度に応じた領域分割木が構成でき, 効率的な経路探査などを行うことができる．
キーワード　GIS, ネットワークボロノイ図, スキップリスト, 経路探索

Generation of Space Partitioning Tree using Network Voronoi Diagrams

Tomoya KAMBARA†, Masaaki OHNISHI††, and Shinichi UESHIMA†

† , ††Graduate school of Informatics, Kansai University
2-1-1 Ryozenji, Takatsuki, OSAKA 569-1095, Japan
E-mail: †{fb6m125,ueshima}@edu.kansai-u.ac.jp, ††ohnishim@nict.go.jp

Abstract　The authors propose probabilistic construction of space partitioning tree using Network Voronoi Diagram considering road map as a graph. From the given graph, to generate a Network Voronoi Diagram, we partition the entire graph and generate subgraphs. Next, considering each subgraphs as nodes, we merge with adjacent subgraphs, and extend the subgraphs. This is processed continuously to construct layers, which the higher level covers the region of lower level, resultantly generating space partitioning tree in a bottom-up manner. Then, using space partitioning tree, we provide route search algorithm and discuss the efficiency of our method. To verify the efficiency, we examine the characteristics of space partitioning tree and its time length for probabilistic selection of generators. We perform numerical simulation for rspace partitioning tree on road maps from geographical survey institute using our algorithm. We use digital road maps efficiently,generating space partitioning tree with different level of details of road network, to perform efficient route search.
Key words　GIS, NetworkVoronoiDiagram, SkipList, Routing Algorithm

1. はじめに

最近, ネットワークボロノイ図を利用した研究が注目を集めている [1] [2]．連続平面の分割に用いられている通常のボロノイ図に対して, ネットワークボロノイ図は, ノードとエッジから構成されるグラフ構造を分割する図として，様々な分野へ応用が期待されている．

本稿では, ノード間の位置関係と距離関係を保持する複数の詳細度別のグラフから構成された階層型データ構造を提案する．ここでは, 道路ネットワークにおいて交差点をノード, 交差点間をつなぐ道路をエッジとしたグラフ構造とみなし, 各階層で, (i) ノードの確率的選択とネットワークボロノイ分割, (ii) 部分グラフの完全グラフ化, (iii) ネットワークボロノイ領域を単位とするグラフの作成, の処理を繰り返し, 漸進的に階層型データ構造を作成する．

提案データ構造を経路探索に用いることで, 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ことが期待できる.

本手法の有効性を評価するために, 国土地理院数値地図（空間データ基盤, 1/2500 縮尺）[3] を用いて, 提案階層型データ構造の生成時の各種パラメータについて, 評価する. 提案手法は, ディジタル地図などの種々の空間データ基盤に対して適用できるものと考えられる.

確率的なデータ構造の構成法として, Pugh [4] は, 1 次元リスト構造に格納されたノードに対する確率的なデータ構造の階層化手法を与えている. Harvey ら [5] は, ノード間の負荷分散を図る目的で, ノードのメンバーシップ関数を用いてＩＤ空間に対してスキップ構造を作成したＰ２Ｐネットワークの構造を提案している. また, Eppstein [6] らは, 空間の 4 分割に繰り返して 4 分木を構成し, 平面内のノードの位置に応じて根からの経路を圧縮した圧縮 4 分木を構成している.

本提案手法は, 平面グラフをネットワークボロノイ図を用いて領域分割と抽象化を繰り返すことで, おおまかな距離地図を計算している. 一方, 連続平面のボロノイ図の階層化は, ＣＧの領域などで LOD(Level Of Detail) を可能にするの詳細度別表示に用いられている.

また, Shahabi ら [7] は, 同様にネットワークボロノイ図を用いた 2 層のデータ構造を用いて, kNN 手法に適用している. Shahabi らの方法は, 店舗や郵便局などの固定的な対象に対してデータ構造を準備しておくため, 異なる対象に対して多数のデータ構造を準備させる必要がある.

以下, 2 章では, ネットワークボロノイ図の特徴について説明し, 提案データ構造の基本的な考え方について説明し，探索手法の提案をおこなう. 3 章で提案データ構造の構成法について述べ, 4 章で国土地理院のデータを利用し実空間において提案階層構造の生成における評価をおこなう.

2. 提案手法

2.1 基本的な考え方

本稿では，探索時間の圧縮を達成する試みとして，Shahabi らの手法に平面グラフの SkipList の考え方を応用した，確率的なノード選択と，地図ネットワークの階層化手法を提案する. ここでは，その基本的な考え方について説明する.

(i) **ネットワークボロノイ分割**：道路ネットワークに対してネットワークボロノイ図を用いることにより, 領域間の位置的な隣接関係を構成し, 各ネットワークボロノイ領域に含まれるノード間において大まかな位置関係と距離を調べている. つまり, ネットワークボ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近傍ノードを分割している. 言い換えれば, 原道路ネットワークの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仮想的に構成することに相当する.

(ii) **部分グラフの完全グラフ化**：(i) を行うことにより二つの隣接する領域に存在する境界を利用して，領域内を経由する場合の距離を算出しておくことで探索時間の短縮ならびに (iii) のための準備をしている.

(iii) **ネットワークボロノイ領域を単位とするグラフの作成**：(i) を行うことにより領域間の隣接関係とおおまかな距離がわかり, (ii) を行うことで領域をひとつのノードと見ることができる. この二つの処理から，ノードは領域，エッジを隣接関係と距離とみた新しいグラフを構成することができる.

また, (i)-(iii) を繰り返して, ノードを選択して部分グラフを作成し, それに対しネットワークボロノイ図を構成することで, 段階的に詳細度の異なるグラフを作成することができる.

すなわち, この手順を繰り返すことで, 与えられた道路ネットワークに対して段階的に詳細度の粗い距離地図を階層的に積み上げてデータ構造を作成できる.

構成した詳細度の異なる距離地図を利用して，長い距離の探索の場合に，粗い距離地図を利用し間にあるノードの数を減らすことで，探索時間の圧縮を図る.

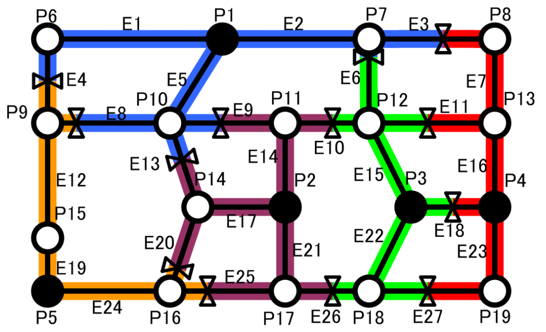

図 1　ネットワークボロノイ図

Fig. 1　Network Voronoi Diagram

母点	所属ノード	所属エッジ
P1	P6, P7, P10	E1, E2, E5
P2	P11, P14, P17	E14, E17, E21
P3	P12, P18	E15, E22
P4	P8, P13, P19	E13, E16, E23
P5	P9, P15, P16	E12, E19, E24

表 1　図 1 における所属母点

Table 1　Belonging of Generator in Network Voronoi Diagrams

2.2 ネットワークボロノイ図

ノード集合 N とエッジ集合 E で構成されたグラフ $G = (N, E)$ を考える. エッジは非負値を長さとして持つものとする.G 上で, 複数のノードを母点として与えた時, G 上の母点以外の各ノードから他のどの母点よりも近い母点をノードの所属する母点という. これにより, G のすべてのノードは, 所属する母点を一意に決定できる. つまり, 与えられた母点に対して, G のすべてのノード集合を一意に分割できる. 分割した図を**ノードネットワークボロノイ図**という.

図 1 にネットワークボロノイ図を示す. 図中の黒点は母点集合 {P1, P2, P3, P4, P5} であり, 各ノードの所属母点を表 1 に示す. 全てのノードの数は 19, エッジの数は 27 である. 以降全ての図において黒点は，母点を指し，白点はノードを指す.

また, エッジについても, エッジ上の任意の地点からの母点への距離を考えれば, 各点の所属母点を一意に決定することができる. この場合, 1 つのエッジすべてが同じ母点に所属する場

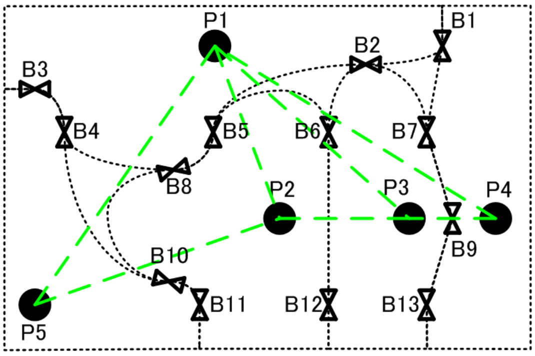

図 2　母点間の距離地図

Fig. 2　Distance Map between Generator Point

合と 1 つのエッジの途中で所属母点が変る場合がある.1 つのエッジの途中で所属母点が変る点を**境界点**と呼ぶ．境界点を持つエッジを**境界エッジ**という. エッジの母点への所属から G のすべてのエッジ集合を分割した図を**エッジネットワークボロノイ図**という.

図 1 では, ⧖が, 境界点を示している. 境界エッジの本数は 13 本. 表 1 には境界点をもたないエッジの所属を表した.

ネットワークボロノイ図を用いて, 道路やその周辺における地形の制約を考慮した, 道路網上の複数地点からの勢力関係を得ることができる. たとえば, 通学の距離を考慮した学区の決定, 消防署の管轄範囲, 店舗の商圏問題などに用いられている. [1], [2]

2.3　ネットワークボロノイ図の階層化と利用

2.3.1　階層化手順

境界点と母点間のエッジは, 母点の位置関係を維持した**距離地図**の生成に当たる．この距離地図は，ネットワークボロノイ図におけるボロノイ図と双対関係にある母点ノード間で作るドロネー図と同じである．図 2 では, 点線はネットワークボロノイ図におけるボロノイ領域に相当する領域を示し, 破線は下位レベルでの距離地図のエッジを示す．これは, 境界点は 2 つの母点をもち，同じ 2 つの母点の組み合わせの境界点から, 最小距離をもつ境界点を発見することで, 最短の組み合わせで距離地図を生成することができる.

ネットワークボロノイ分割を行うと，部分領域を単位としたグラフと見なすことができる．このことよりより提案手法では，距離地図（図 2）内の P1 の持つ領域を上位レベルにおいては P1 と等価として扱っている.

同時に各領域でもつ境界点の全てを n としたとき $n(n-1)/2$ の組み合わせを接続する．この境界点間のエッジは, 上位の階層を生成するときに, 母点から新しく母点の配下となったノードの境界点までの計算を新規にレベル 0 上で探索する必要がない．他にも領域を経由するときにそのエッジを利用することで領域内の計算を簡略化することができる．このグラフを指して本稿では，**完全グラフ**と呼ぶ.

距離地図と完全グラフは全てのレベルで生成でき，この距離地図と完全グラフを同一のものとしてみたものを**完全地図**と呼ぶ．以降原地図をレベル 0 と呼び，下位のレベルより上位のレベルにかけてレベル 1・2・3… とする.

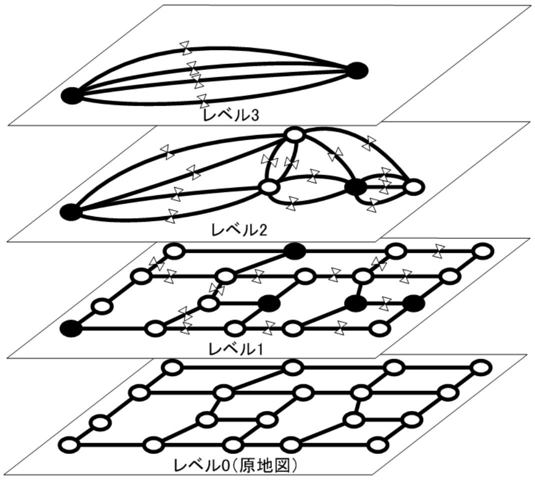

図 3　階 層 構 造

Fig. 3　Hierarchical Network

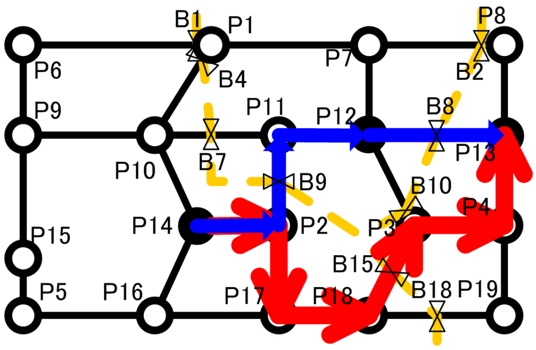

図 4　ある状況における完全地図

距離地図の生成を行い，完全グラフを生成する．生成できた完全地図上でネットワークボロノイ図を生成することで，粗い距離地図が生成できる．これを繰り返すことで，領域分割木を生成することができる (図 3).

2.3.2　探 索 手 法

前節で構築した完全地図上で，任意の 2 地点を始点と終点とした場合の経路探索を行う手法を提案する．大きく分けて三つの手順に分けることができる.

手順 1：始点と終点の関係を知る

生成時の情報より，始点と終点の所属する母点を全てのレベルにおいて，知ることができる.

異なる母点になった場合，境界点の情報から母点の関係がわかり関係は次の 2 通りである．それは，隣接しているか，隣接していないかである.

隣接している場合，レベルを下げる．どの位置で隣接しているかわからないためでその境界点を利用すると大幅に遠回りの可能性があるからである. 図 4 の場合，P14 から P13 に到達する最短経路は，P14・P2・B9・P11・P12・B8・P13（図中青線）であるが，隣接しているので隣接を示す境界点を経由すると，P14・P2・P17・P18・B15・P3・P4・P13（図中赤線）となり，最短経路と比較して約 1. 3 倍の経路となる.

経路探索において A*などの経路探索手法を利用し，利用する

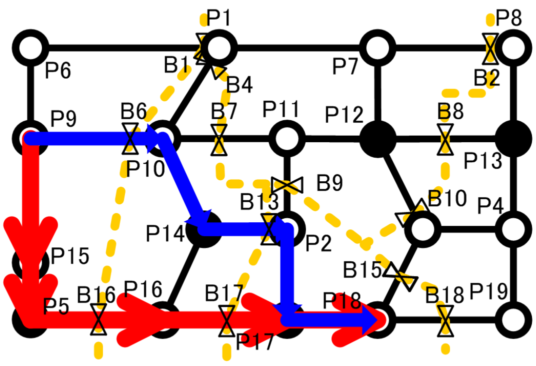

図 5　あるレベルにおける完全地図

境界点の組み合わせを決定していかない理由は，上位レベルにおいて始点終点が母点でない限り，最短経路であるとはいえないからである．図 5 は，あるレベルにおいて P9 を始点とし，P18 を終点とする経路探索を考える．始点の母点 P5 から終点の母点 P17 に対して A*経路探索を行った場合，利用する経路は，P9・P15・P5・B16・P16・B17・P17・P18 を通るものになるが，実際に始点から終点に至る最短経路は，P9・B6・P10・P14・B13・P2・P17・P18 であり，実際の最短経路とは異なる．

手順 2：離れた領域間を組み合わせによる経路を生成する

隣接していない場合，始点の母点領域の境界点から終点の境界点まで全ての組み合わせにおいて，完全グラフ上で経路探索をおこなう．

逐次的にレベルを下げ，境界点の数を変化させる．境界点数の変化により組み合わせの数が変化する．上位レベルで探索した部分と新しく探索する部分を組み合わせ，最小の組み合わせのみに残す．また，母点のもつ領域内を経由する経路は削除する．この組み合わせの数は，始点の母点が持つ境界点の数×終点の母点が持つ境界点の数以下になる．

図 6 の S は始点を，G は終点を示している．右図は，上位階層と下位階層での探索範囲の違いと，探索範囲の変化を示している．左図は，組み合わせの結果生成できたエッジを示している．図 6 では，上位レベルで始点の所属する母点の持つ境界点 {B1, B4, B11} と終点の所属する母点のもつ境界点 {B3，B7，B9，B10，B12} をつなぐ経路は無数に存在するが，距離が最短となる経路は，15 通り生成することができる．

下位レベルでの組み合わせには，上位レベルで生成した経路と上位レベルで同じ母点の配下に存在したノードの領域を利用する．始点の所属する母点の持つ境界点 {B1，B4，B13} と終点の所属する母点の持つ境界点 {B9，B10，B12，B14} を最短でつなぐ経路 12 通りを生成する．このとき右図の斜線部を通る経路は，上位レベルでの結果である 15 通りの中から選択することにより探索の時間の圧縮が図ることができる．左下図の点線は，組み合わせの結果最短とならなかった始点側境界点と終点側境界点をつなぐ経路である．

レベルを下げた場合に，始点か終点のどちらか，または，両方が母点となる場合がある．このとき，全てのレベルで生成時に，母点と境界点をつなぐ経路を生成している．この経路を利用して，探索時間の圧縮を図る．図 6 下位レベルでは，終点が母点となっている．このとき太線のエッジを利用し境界点 {B9，B10，B12，B14} に至る経路を知ることができる．母点となった側は，これ以上レベルを下げないが，母点とならなかった側は，さらにレベルを下げ，組み合わせを継続する．

レベルが 1 のとき始点か終点が母点ではなかった場合，そのノードは，完全地図上に存在していない．このときそのノードが所属する母点の持つ境界点に，レベル 0 の地図上で経路探索を行う．例えば，図 6 の下のレベルがレベル 1 のときの完全地図だった場合，始点の所属する母点の持つ境界点 {B1，B4，B13} に対して経路探索を行うことで，境界点までの距離を算出することができる．

手順 3：始点から終点までの経路を生成する

手順 2 で算出した経路 {始点から境界点までの経路・始点の境界点から終点の境界点間の経路・終点の境界点から終点までの経路} を組み合わせ，最終的に最も経路長が短い経路を解として出力する．

2.4　特　　徴

確率的なノード選択

ネットワークのノードの疎密に依存せず，一様に母点と配下ノードを構成できるように，確率的に母点をノード集合から選択する．この確率的にノード選択を行うことにより，地図上のノードの疎密に応じた母点配置を得ることができる．また，ノードを限定する条件をつけないのでディジタル地図に適用しやすいなどの利点を挙げることができる．

ボトムアップ生成

ボトムアップに生成することにより，トップダウンに生成する手法とくらべて，領域分割木の深さが，ノードの密集により一部の枝のみが深くなるような問題が発生しない．

上位レベルでの下位レベルの領域の完全な包含

下位レベルで生成した母点の持つ領域単位で合併を行い，上位レベルでの領域を生成していくことで，上位レベルでの領域は下位レベルでの領域を完全に包含することができる．

合併による抽象度の異なるグラフの生成

レベル 1 で生成する境界点は，レベル 1 での母点の領域を示している．この境界点を元に，母点の持つ領域単位で抽象ノードを生成できる．境界点は，母点の隣接関係を示している．この隣接関係から，抽象化したエッジを生成することができる．この抽象ノードと抽象エッジから，抽象グラフを生成できる．

境界点の再利用

レベル 2 以降の抽象グラフの生成では，エッジネットワークボロノイ分割を実行して，抽象エッジ上に境界点を新規に生成しない．これは，抽象エッジは複数のレベル 0 の地図上のエッジで構成されている，この複数のエッジ上のどこかにレベル 2 以降の境界点を生成するには，レベル 0 の地図で最探索する必要がある．レベルごとに境界点を生成すると，抽象エッジの増加が考えられ，抽象グラフが複雑になると考えられる．

本稿では，レベル 1 で生成した，境界点を再利用することを提案する．ノードネットワークボロノイ分割を実行して，母点ごとに抽象グラフを分割する．母点への所属関係から，境界点の所属関係の変化も分かり，ノードの持つ領域の端である，境

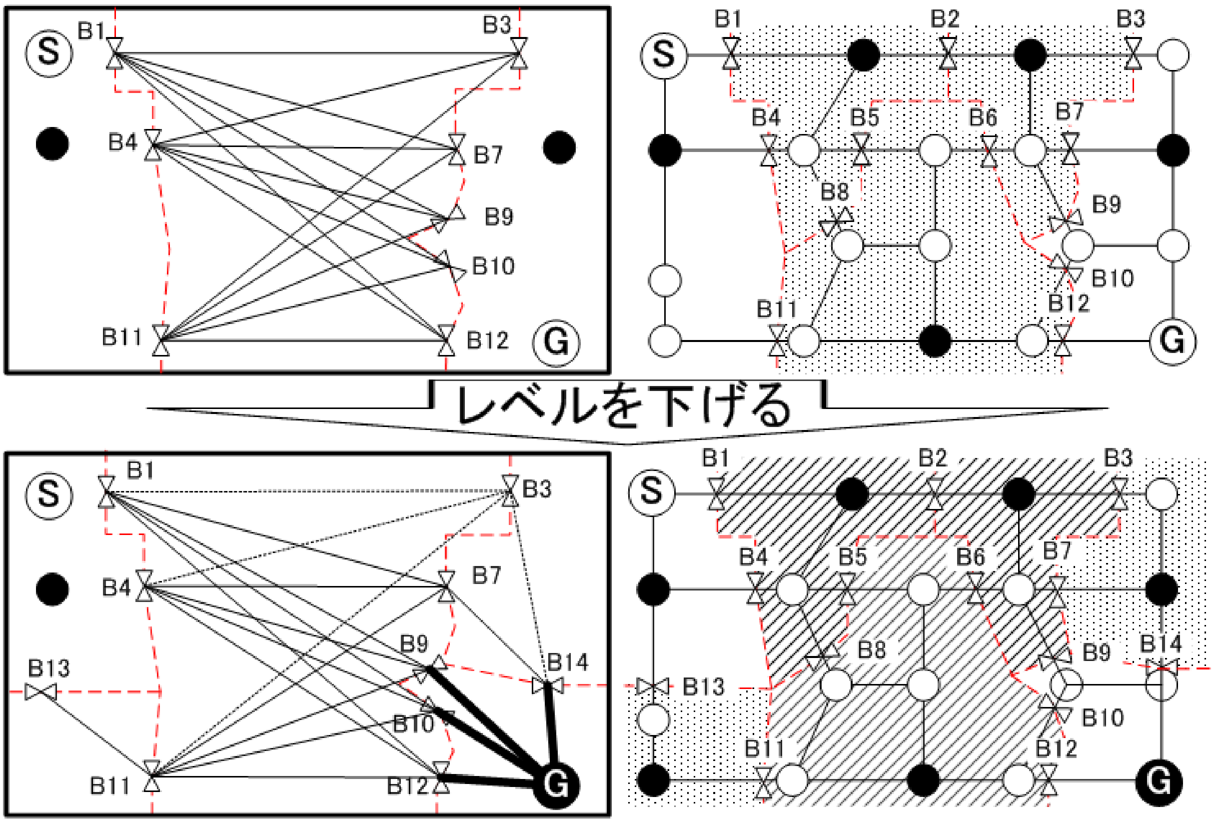

図 6　レベルに応じた探索範囲の変化

界点の所属を変化することによって，領域も所属を母点とすることができる．

母点の領域の端の変化により，抽象ノードの持つ領域の規模が拡大する．境界点の接続の変化から，抽象エッジのエッジ長が増加する．この抽象ノードと抽象エッジを用いて，下位レベルでの抽象グラフより，さらに詳細度の低い上位レベルの抽象エッジを生成することができる．

詳細度の異なるグラフ

詳細度の異なるグラフを利用することにより，長い距離を探索する場合は，上位レベルの粗いグラフを利用し，途中の経由するエッジの探索を抑え，短い距離を探索する場合は，下位レベルでの詳細なグラフを利用することにより，詳細な探索を行うことができる．

3. 構成アルゴリズム

3.1 並列 Dijkstra

並列 Dijkstra とは，ネットワークボロノイ図生成における，ノードネットワークボロノイ分割を行うアルゴリズムである．Dijkstra 法の変形で，母点となるノード集合の上に重み 0 の root を生成し，Dijkstra 法を行うことで，ネットワーク上の全てのノードの所属を分割することが出来る．(Algorithm1)

Algorithm 1 *PD*: 並列 Dijkstra

```
for all p ∈ N do
  if p ∈ K then
    V(p)←p;D(p)←0;P(p)←root;M(p)←true;insert(h, p)
  else
    V(p) ← null; D(p) ← ∞; P(p) ← null; M(p) ← false
  end if
end for
for all outgoing_edge(p, w) with M(w) = false do
  Δ ← D(p) + cost(p, w)
  if D(w) = ∞ then
    V(w) ← V(p); D(w) ← Δ; P(w) ← p; insert(h, w)
  else if D(w) < ∞ and Δ < D(w) then
    V(w)←V(p);D(w)←Δ;P(w)←p;delete(h, w);insert(h, w)
  end if
end for
while h is not empty do
  p ← min(h); delete(h, p); M(p) ← true; expand_childnode
end while
```

3.2 レベル 1 生成

3.2.1 境界点生成

前述の並列 Dijkstra 法では，ノードの所属は判明するが，エッジの所属・境界エッジの数・境界点の位置などは分からない．この過程をエッジネットワークボロノイ分割といい，生成結果をエッジネットワークボロノイ図という．

全てのエッジの両端ノードを調べる，このとき以下の二つの可能性がある．

- 両ノードの所属が同じ
- 両ノードの所属が異なる

両ノードの所属が同じ場合，そのエッジの所属母点はノードの所属母点と同じである．両ノードの所属が異なる場合，そのエッジは，境界エッジであり，境界点が存在する．境界点から母点までの重みは，(ノード a から母点 A までの重み + 境界エッジの重み + ノード b から母点 B までの重み)/2 で判明する．

3.2.2 境界点と母点をつなぐ経路の生成

境界点と母点をつなぐ経路は，境界エッジの端ノードの所属母点はひとつであり，ノードネットワークボロノイ分割を実行

Algorithm 2 境界と母点をつなぐ（*make_path*）

add $e(b)$ to $PathEdgeList(e(b))$
while $p1 \neq V(p1)$ **do**
  add $Edge(p1, P(p1))$ to PathEdgeList($e(b)$);$p1 \leftarrow P(p1)$
end while
while $p2 \neq V(p2)$ **do**
  add $Edge(N2, P(N2))$ to PathEdgeList($e(b)$);$p2 \leftarrow P(p2)$
end while

時に算出される親ノードは, 母点につながるノードであり, 母点までの距離は, ノードの情報として持つ. 境界エッジの端ノードから境界点の距離を算出し, 母点までの距離を足すと, 境界点と母点をつなぐ経路と距離となる.（Algorithm2）

3.2.3　境界点間をつなぐ経路の生成

境界点間をつなぐ経路は, 最大 2 本算出することが出来る.2 本存在する場合は, 境界点のどちらも同じ母点の組み合わせの場合のみである. しかし母点ごとに完全グラフのエッジを持つので同一の結果を持つことはない. 母点の組み合わせが異なれば, 他の境界点を経由することになり, 多数の候補があげられるので, 直接つなぐ経路は 1 本になる. どちらの場合も, 同様の手段で算出する. 片方の境界エッジの端ノードからもう一方の境界エッジの端ノードに探索をし, その間の経路・距離を記録することでなされる.

3.3　レベル 2 以降生成

完成した距離地図上で並列 Dijkstra 法を実行する. ネットワークボロノイ図を生成する母点は, 距離地図上のノードに確率 p で選択する. ノードネットワークボロノイ分割により, ノードの母点所属が判明する.

母点と配下にあるノードとは, 境界点でつなげることができる. その境界点を連結し, 配下のノードの持っていた他の母点に通じる境界点までの経路を生成する. この経路は, 合併する母点と配下ノードの完全地図を利用し, 境界点と母点間経路, 境界点間経路を生成することができる.

例として図 1 の P2 を母点として配下に P3 が所属する場合を考える. P2 は境界点を 6 個, P3 は境界点を 6 個持ち, そのうち P2・P3 と重複する境界点は 2 個である. まず P3 の持つ P2 と重複していない境界点に対して境界点と母点間をつなぐ経路を生成する. これは, P2・P3 と重複する境界点・P3 のみが所持する境界点へと完全地図を利用して生成できる. 同様に P2 のみが持つ境界点・P3 と重複する境界点・P3 のみが所持する境界点へと接続することにより境界点間経路となる.

3.4　探索アルゴリズム

始点と終点の関係を知る

組み合わせによる経路探索のために, 始点と終点の関係が必要である. 母点ノードも母点に所属していると考えたときこの始点と終点の関係とは, 次の 3 通りある. {同一の母点に所属・母点同士が隣接・母点同士が隣接していない}

同一の母点に所属しているかしていないかはノードの情報から読み取ることができる. (Algorithm7)

LV：生成された完全地図での最後のレベル, V(p,i)：レベル i でのノード p の母点, Dlv：始点と終点の母点が異なる最後のレベル, とする.

Algorithm 3 ノードの同一母点所属判定

$i = 0$
for $i \rightarrow LV$ **do**
  if $V(s,i) \neq V(g,i)$ **then**
    $Dlv \leftarrow i$
  end if
  $i \leftarrow i + 1$
end for

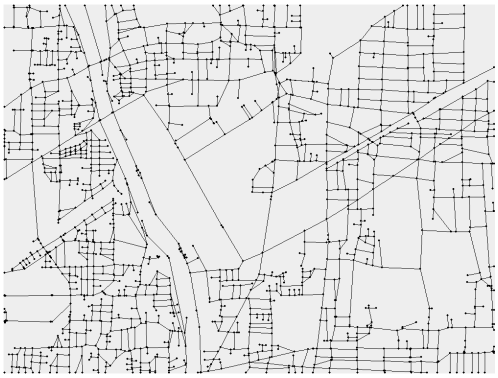

図 7　グラフ構造化した高槻市街中心部の道路網
Fig. 7　Base Road Network

隣接する母点に所属しているか, 所属している母点が隣接していないか, は境界点の情報から読み取ることができる.

経路候補となる組み合わせを生成する

始点の所属する母点の持つ境界点のリストから終点の所属する母点の持つ境界点までをすべての組み合わせを保持する. 上位のレベルで生成した経路候補に, 新規に探索した部分を付加していくことで, すべての組み合わせを網羅する. 始点か終点が母点となったレベルでは, 母点となった側はレベルを下げないが, ならなかった側は, レベルをさげ新規に探索する部分を知る.

経路候補から経路を生成する

経路候補は境界点と境界点をつなぐ経路なので, 始点と境界点をつなぐ経路, 終点と境界点をつなぐ経路とつなげ, コストが最小となる組み合わせを最短経路とする. このとき, 始点と終点が, 母点か母点でないかで対応が変わる. 母点の場合, 完全地図から, 母点と境界点をつなぐ経路をもつ. 母点でない場合, レベル 0 の地図上で, 経路探索をする必要がある. 対象の点を始点ノードとし, 対象の点の母点に所属するノードにダイクストラ法による探索を行い, 境界点の持つ同一母点の端ノードまでの探索をおこなう.

4. 評　　価

4.1　道路地図の利用

本稿では, 国土地理院数値地図 2500（空間データ基盤, 1/2500 縮尺）の大阪府高槻市街中心部の地図を利用する. この道路網のみを図 7 に示す. 黒線が道路, 黒線上の点が交差点を示す. 交差点数は 1612 個, 道路の本数は 2077 本である.

4.2　階層化地図構造としての評価

ここでは, 前述した高槻市街マップを利用し, 確率 p を 1/2・

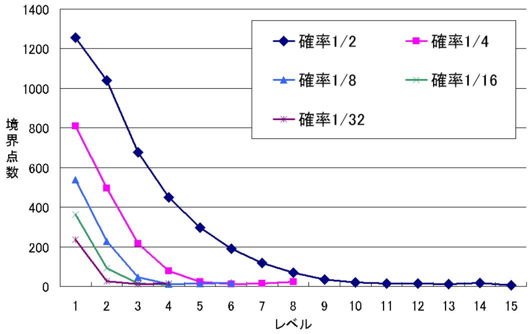

図 8　評価 1：各レベルでの境界点数の推移

Fig. 8　Transition of boundary point at each level

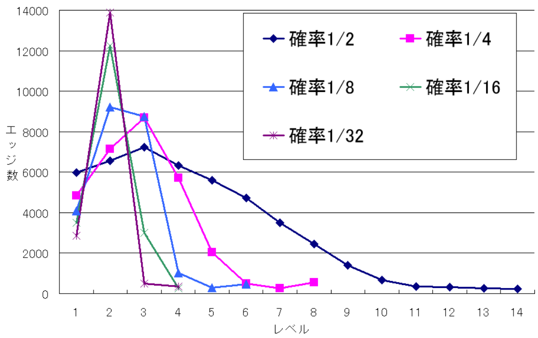

図 9　評価 2：完全地図上エッジ数の変化

Fig. 9　Change in number of Edges on Complete Map

1/4・1/8・1/16・1/32 と変化したときの各項目について評価及び考察を行う．確率が関与するので各確率において 100 回の試行を行いその平均で評価をする．

評価 1：各レベルでの全ての境界点数の推移

評価 1 の結果 (図 8)：確率の如何にかかわらず同様の境界点数の減少がみられた．確率 1/2 の場合をみると，レベル 1 から 2 にかけて母点数が半分に減少するのだが，境界点の数は約 1/4 ほどしか減少しない．また，同じレベルにおいて確率が半分になっても境界点の数が半分にならないことが見てとれる．

評価 2：完全地図上の各階層における総エッジ数の変化

評価 2 の結果 (図 9)：確率 1/2 のときレベル 3 にかけて緩やかに増大し以後減少していく．これはひとつの領域の持つ境界点の数が，レベルの変化により，吸収されグラフ上から消えていくものより，合併し新規に接続しなければならないものの方が多いからで，減少するのは，逆で吸収されグラフ上から消えていく方が多いからだと考えられる．

評価 3：完全地図を生成するために，並列 Dijkstra におけるノード総展開数と完全グラフを構成するときの組み合わせの総展開数を比較することで，事前計算量のバランスの取れた確率を考察する．

評価 3 の結果 (図 10)：並列 Dijkstra におけるノード総展開数が確率 p が小さくなるにつれて減少しているのは，生成するレベル自体が少なくなっているからだと考えられる．完全グラフを構成する組み合わせの総展開数が右肩上がりなのは，評価 2 のレベル 2 での完全グラフのエッジ数が膨大で，その組み合わせが作用しているからだと考えられる．

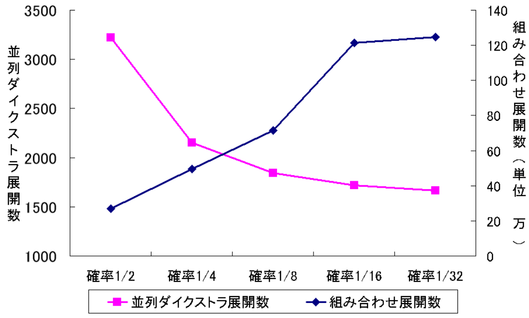

図 10　評価 3：展開数

Fig. 10　Number of Node expansion

5. おわりに

本稿ではネットワークボロノイ図を用いて，階層化した地図を生成する手法を述べた．この階層化した地図を利用した経路探索手法を提案した．ネットワークボロノイ図を用いた研究は，母点ごとの支配領域の生成に主だっているものが多いが，本稿では境界点が母点間の隣接関係を示すことに着目した階層化ネットワークを構成する．

今後の課題として，本稿では，経路探索については，提案だけに終わったが，A*最短経路探索法などと比較する必要がある．そして提案データ構造は, ランダムにノードを選択して階層化しているため, k 最近傍問題においては, 本構造 1 種類で, 種々のオブジェクトを対象とした k 最近傍問題にも適用することができると考えられるので検討する必要がある．

文　　献

[1] M.Erwig, “The Graph Voronoi Diagram with Applications”,Networks, 36(3), 156-163, 2000.

[2] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, Sung Nok Chin, “Spatial Tessrllations:Concepts and Applications of Voronoi Diagrams” “2nd edition”, pp.218-224, 2000

[3] 国土地理院, 数値地図 (空間データ基盤), http://sdf.gsi.go.jp/

[4] William Pugh. “Skip lists: A probabilistic alternativeto balanced trees.” Communications of the ACM, 33(6):668-676, June 1990.

[5] Nicholas J.A. Harvey, Michael B. Jones, Stefan Saroiu, Marvin Theimer., Alec Wolman.“SkipNet: A Scalable Overlay Network with Practical Locality Properties” the 4th USENIX Symposium on Internet Technologies and Systems (USITS ’03), vol 4, p. 9 (2003).

[6] David Eppstein, Michael T. Goodrich , Jonathan Z. Sun“The Skip Quadtree: A Simple Dynamic Data Structure for Multidimensional Data”, Proceedings of the 21st ACM Symposium on Computational Geometry 2005, pp.296-305

[7] Mohammad Kolahdouzan, Cyrus Shahabi, “Voronoi-based K Nearest Neightbor Search for Spatial Network Databases”, Proceedings of the 20th VLDB Conference, Tronto, Canada 2004, pp.840-851