6L-01

漸進的な集約ネットワーク距離計算と最短経路キャッシングに基づく地理情報検索アルゴリズム

中山　泰宏†　　佐藤　秀樹‡

†大同大学大学院　情報学研究科

1. はじめに

複数の集約ネットワーク距離[1]を評価関数の集合 F として、道路網上での地理オブジェクト集合 D に対するスカイライン集合[2]を求める問題を検討する。①の ψ(F,D)は、問題の解集合を定義する。②で定義される関係$\succ_F$は、F に関する優越関係である。$f_i(p,Q_i)$は、地理オブジェクト p と問合せ点集合 Q_i 間の集約ネットワーク距離である。一般性を失うことなく、f_i に関して小さい値が大きい値より優越していることとする。

$\psi(F,D)=\{p\in D|\nexists p'\in D, p' \succ_F p\}$　　・・・①

$p \succ_F p' \Leftrightarrow (\exists f_i\in F,\ f_i(p,Q_i) < f_i(p',Q_i)) \wedge (\forall f_j\in F, f_j(p,Q_j)\leqq f_j(p',Q_j))$　・・・②

BatchLBC アルゴリズム[3]は、LBC アルゴリズム[4]を拡張した求解法である。本稿では、BatchLBC アルゴリズムの処理性能の改善手法を検討し、その有効性を実験により確認する。

2. BatchLBC アルゴリズム

BatchLBC アルゴリズムは、任意の評価関数をソース関数 $f_{src}(\in F)$として選択する。非ソース関数 $f_i(\in F, i\neq src)$に対して、集約ネットワーク距離の昇順に既存スカイライン・オブジェクトを整列するソートリストが維持される。f_{src} の集約ネットワーク最近傍 p が求められると、各非ソース関数のソートリスト上の集約ユークリッド距離 $d^E_{agg}(p,Q_i)$の位置に p が置かれる。ここで、既存のスカイライン・オブジェクトに優越されれば、p は破棄される。そうでなければ、順に p の非ソース関数の集約ネットワーク距離$d^N_{agg}(p,Q_i)$が計算され、ソースリスト上の位置を移動する。この過程で、既存のスカイライン・オブジェクトに優越されれば、即時に p は破棄される。全ての非ソース関数の集約ネットワーク距離$d^N_{agg}(p,Q_i)$が計算され、破棄されなければ、p がスカイライン・オブジェクトとして判定される。

3. 漸進的 LBC アルゴリズム

地理オブジェクト p と問合せ点集合 Q_i との間の集約距離は集約ユークリッド距離を下限とし、p と $q_{ij}(\in Q_i)$との間のユークリッド距離のネットワーク距離への置換により単調増加し、最終的に集約ネットワーク距離となる。このため、非ソース関数の集約距離計算過程で既存のスカイライン・オブジェクトに優越されることが判明した場合、それ以降の p のネットワーク距離計算を削減できる。この性質に基づき、BatchLBC アルゴリズムの改良アルゴリズムを示す。

3.1 IncrementalLocalLBC アルゴリズム

図 1 は、IncrementalLocalLBC アルゴリズムにおけるソース関数のネットワーク最近傍 p に対する、非ソース関数に関する既存スカイライン・オブジェクトとの間の優越関係の判定箇所である。ソースリスト上の p の位置が上位である非ソース関数 f_i の順に問合せ点 q_{ij} と p との間のネットワーク距離が ShortestPathQuery により計算され、漸進的に集約距離が更新され(3 行目)、ソートリスト上の位置が更新され(4 行目)、p が既存スカイライン・オブジェクトに優越される場合(5 行目)、p は即時に破棄され(7, 10 行目)、それ以降のネットワーク距離計算が回避される。

```
1.  for(fi(∈F) ordered by position in fi.sortList){
2.    for (qij(∈Qi)){
3.      <p,d_agg(p,Q_i)>:=ShortestPathQuery(p,qij);
4.      update p's position in fi.sortList;
5.      if (p is dominated by s(∈S)){
6.        delete p from all sortLists;
7.        break;
8.      }
9.    }
10.   if (p is dominated by s(∈S)) break;
11. }
12. if (p isn't dominated by s(∈S)) p is skyline;
```

図 1 .優越関係の判定 (IncrementalLocalLBC)

3.2 IncrementalGlobalLBC アルゴリズム

図 2 は、IncrementalGlobalLBC アルゴリズムにおけるソース関数のネットワーク最近傍 p に対する、非ソース関数に関する既存スカイライン・オブジェクトとの間の優越関係の判定箇所である。集約ネットワーク距離計算が未完了で、かつソートリスト上の p の位置が最上位である非ソース関数 f_i を選択し(2 行目)、最短ユークリッド

Geographical Information Retrieval Algorithm Based on Incremental Aggregate Network Distance Computation and Shortest Route Caching
Yasuhiro NAKAYAMA†, Hideki SATO†
†Graduate school, Daido University

距離である問合せ点 q_{ij} と p との間のネットワーク距離が ShortestPathQuery により計算され、漸進的に集約距離が更新され(4 行目)、ソートリスト上の位置が更新され(5 行目)、p が既存スカイライン・オブジェクトに優越される場合(6 行目)、p は即時に破棄され(8 行目)、それ以降のネットワーク距離計算が回避される。以上が全ての非ソース関数の集約ネットワーク距離計算が完了するまで繰返される。

```
1.    while(∃fi(∈F) whose d^N_agg isn't calculated){
2.      select fi(∈F) whose d^N_agg isn't calculated and
          whose position in fi.sortList is least;
3.      select qij(∈Qi) such that d(p,qij) is euclid
          and least;
4.      <p,d_agg(p,Qi)>:=ShortestPathQuery(p,qij);
5.      update p's position in fi.sortList;
6.      if (p is dominated by s(∈S)){
7.        delete p from all sortLists;
8.        break;
9.      }
10.   }
11.   if (p isn't dominated by s(∈S)) p is skyline;
```

図 2 . 優越関係の判定 (IncrementalGlobalLBC)

4. 最短経路キャッシング

BatchLBC アルゴリズムでは、ある地理オブジェクトと異なる問合せ点との間の最短経路探索，逆にある問合せ点と異なる地理オブジェクトとの間の最短経路探索が多数行われる。最短経路に関しては、次の定理が成立する（証明略）。

［定理］ 道路網上の任意の 2 点間の最短経路の部分経路は最短経路である。 □

この定理に基づき、以前計算された最短経路の情報を保存し（キャッシュ化）、目的ノードを共有する別の最短経路探索において再利用することにより、異なる最短経路間で共有される部分経路計算の回避を可能とする（図 3 参照）。本稿では、目的ノードを共有する部分経路の活用を図る。

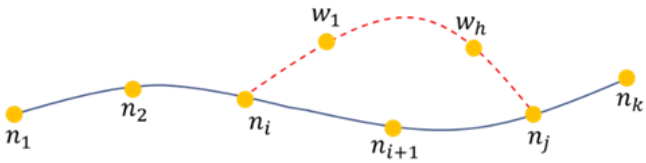

図 3 . 最短経路の部分経路の最適性

5. 評価実験

漸進的アルゴリズムと経路情報キャッシング機能を Java 言語により実装し、Intel Core i5-6600 PC(クロック周波数 3.3GHz, メモリ 8GB)を用いて評価実験を行なう。道路網データは愛知県データ[5]を、地理オブジェクト・データは賃貸物件情報(愛知県 32 万件)[6]を用いる。問合せ点数 5 の SUM 集約距離を評価関数として、名古屋駅を中心とする 500m 四方内で問合せ点の位置を無作為に生成する。図 4 は、問合せ 100 問の処理時間の平均値である。BatchLBC と比較すると、評価関数の数が 10 の場合での各アルゴリズムの処理時間は、IncrementalLocalLBC はキャッシュ無で 70%、キャッシュ有で 23%であった。IncrementalGlobalLBC はキャッシュ無で 59%、キャッシュ有で 19%であった。

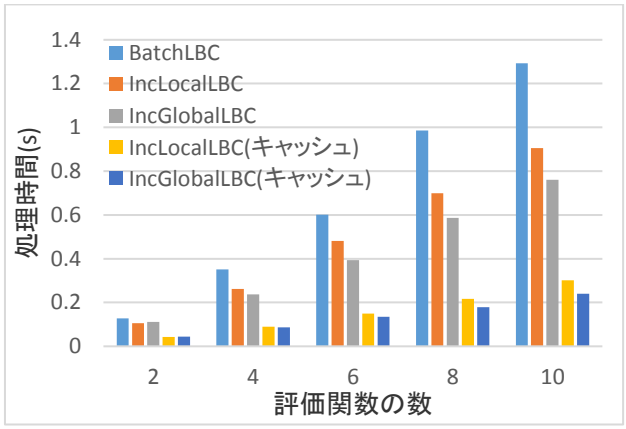

図 4 . アルゴリズムの処理時間（名古屋駅周辺）

6. おわりに

BatchLBC アルゴリズムの処理時間改善のため、漸進的計算と最短経路キャッシングを提案した。評価実験により、提案アルゴリズムの処理時間が Batch LBC アルゴリズムの 20%以下であることを確認した。

謝辞 本研究では、株式会社 LIFULL が国立情報学研究所の協力により研究目的で提供している[LIFULL HOME'S データセット]を利用させていただきました。

参考文献

[1] Yiu, et al: "Aggregate Nearest Neighbor Queries in Road Networks", IEEE TKDE, Vol.17, No.6, pp.820-833, 2005
[2] S.Borzsonyi, et al: "The Skyline Operator", ICDE, pp.421-430, 2001
[3] 中山,佐藤: "道路網上における複数評価尺度に基づく空間問合せに対する処理"平成 29 年度 電気・電子・情報関係学会 東海支部連合大会, D2-1
[4] K.Deng, et al: "Multi-source Skyline Query Processing in Road Networks", ICDE, pp.796-805, 2007
[5] 国土交通省国土地理院: 数値地図 2500(空間データ基盤)中部-2 http://www.gsi.go.jp/geoinfo/dmap/dm2500sdf/
[6] 国立情報学研究所 情報学研究データリポジトリ: LIFULL HOME'S データセット http://www.nii.ac.jp/dsc/idr/next/homes.html