

# 旅行計画における IER の利用とその検証の高速化

## A Fast Verification Method for Trip Planning Query Based on Incremental Euclidean Restriction

日色 史晃\*  
Nobuaki Hiiro

トウ トウ\*  
Htoo Htoo

大沢 裕\*  
Yutaka Ohsawa

### 1. まえがき

車や人などの移動体が現在地から最終目的地に達する途中で、あらかじめ指定されたいくつかの種類の POI を訪れる際の最適経路を求める検索は、旅行計画と呼ばれる。例えば、現在地から最終目的地に至る途中で、ガソリンスタンド、レストラン、ショッピングセンター(これらを POI 種と呼ぶ)を訪れる際の最適経路を求める検索である。この際、POI 種の訪問順序が一意に指定されている場合と、訪問順序にはこだわらず全ての種類の POI を 1 つずつ訪れればよい場合とで計算の困難さが異なる。前者は Shrarifzadeh ら [1] により提案され、OSR(optimal sequenced route) 検索と名付けられている。また、後者は Li らにより提案され、TPQ(trip planning query) と名付けられている。更に、両者の中間の検索、即ち、部分的な訪問順序の規則が与えられる検索は Chen らにより提案され MRPSR(multi-rule partial sequenced route) 検索と名付けられている。本稿では、これらの全ての検索を旅行計画問題という名で総称する。

これらの旅行計画の内、計算コストが最も少ない OSR においても途中で訪れる POI の種類数が増えると多大なコストを要する演算となる。いま、 $C_1$  から  $C_m$  までの  $m$  種類の POI 種を、添え字の順序で訪れる検索を考える。また、各 POI 種にはそれぞれ  $N_i (1 \leq i \leq m)$  個の POI が含まれるものとする。現在地を  $s$  とし、最終目的地を  $d$  とするとき、まず  $s$  から最初の  $C_1$  を訪れる場合の数は  $N_1$  ある。この中から 1 つを決定して、その POI から次の  $C_2$  の POI を訪れる場合の数は  $N_2$  個ある。即ち、 $s$  から  $d$  に至る途中で各種類の POI を 1 つずつ訪れる場合の組み合わせ数は、 $\prod_{i=1}^m N_i$  となる。これが OSR の検索コストの最悪値である。このように計算困難な問題のため、OSR の解法として従来近似解を得るアルゴリズムが検討されてきた。また、道路網距離で最短 OSR を求めようとするとき、更に計算コストの高い演算となる。

本稿では、旅行計画を道路網上に移動が制約されている状況で最適解を求める方法を提案する。基本的な方針は、ユークリッド距離での検索で候補を探し、それを道路網距離で検証する枠組み、IER(incremental Euclidean restriction)[2] である。この IER を旅行計画問題に適用するためには、ユークリッド距離での候補をインクリメンタルに、即ち経路長が最小のものから順に検索できる必要がある。しかし、そのようなアルゴリズムに対する従来の提案は極わずかである。本稿では、位置に関連する情報サービス (LBS) での利用を目

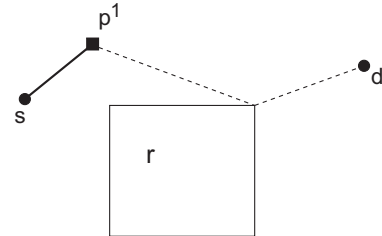


図 2: ユークリッド距離での OSR 検索

的とした、旅行計画の為の検索方式を提案する。

### 2. ユークリッド距離での OSR 検索

IER の枠組みで道路網上の距離の OSR 探索を行うためには、まずユークリッド距離での OSR をインクリメンタルに求める必要がある。しかしこの目的で用いることができる既存アルゴリズムは Sharifzadeh らによる PNEのみである。しかし、この PNE は探索空間が爆発的に増加するため  $m$  の値が小さい ( $m \leq 5$  程度) 場合にしか適用できない。そこで、筆者らは検索開始点  $s$  と最終目的地  $d$  を結ぶ線分と、POI 集合が管理されている R 木の MBR との距離を用いて最適優先探索を行うことにより、ユークリッド距離での OSR 路をインクリメンタルに求める方法を提案した [4]。

図 2 は、このアルゴリズム (EOSR) の基本的な動作を示している。現在値を  $s$  とし、最終目的地を  $d$  とするとき、 $s$  から  $d$  に至る途中で指定された POI ( $C_i, 1 \leq i \leq m$ ) の中から順番に 1 つずつ訪れる経路を探索する。検索に際しては、まず  $C_1$  の POI を管理する R 木の MBR( $r$ ) を参照して、 $s$  から  $r$  をたどり、 $d$  に至る経路長  $L_{lb}$  の下限値を求める。その後  $L_{lb}$  最小の  $r$  を 1 段ずつ下にたどり、最終的に POI( $p^1$ ) に至る。この時点で  $s$  から  $C_1$  の  $p^1$  を経由して  $d$  に至る経路候補が求まることになる。次に、 $p^1$  から  $d$  に至る経路上で  $C_2$  に含まれる POI を求める処理を開始する。図 2 はこの様子を示している。

その後、最終的に  $C_m$  に含まれる POI( $p^m$ ) を探し、この時点で 1 つの OSR 路が探索されたことになる。このアルゴリズムは、経路長が最小のものから厳密解の探索が行え、かつ徐々に距離が長くなる OSR 路をインクリメンタルに探索することができる。

### 3. 道路網上の距離での検証

ユークリッド距離で得られた候補に対して、道路網上の移動距離を求めることにより、候補が道路網上の距離による OSR であるかを検証する。この為に、まず EOSR によりユークリッド距離最短の OSR を 1 つ求める。その結果では途中で経由する POI がすでに求まっているため、道路網上の距離による検証には A\* アルゴリズムや、2 点間距離のマテリアライズ化も用いることができる。これらの方式により、距離を求めようとする 2 点が指定されれば、高速にその距離を得ることができる。

ユークリッド距離での最短 OSR 路を道路網上の距離で検証した結果の OSR 長を  $L_{min}$  とする。次に EOSR をインクリメンタルに起動して、距離が  $L_{min}$  以下の OSR を全て求める。最初に求めた OSR を含む、距離  $L_{min}$  以下の OSR の集合を  $C$  とするとき、道路網上の経路長最短の OSR は必ず  $C$  に含まれることになる。

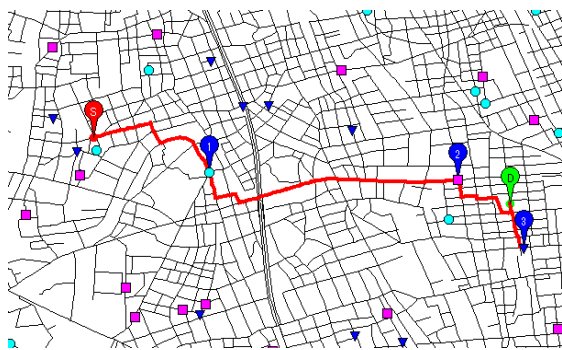


図 1: OSR 検索の例

\*埼玉大学

集合  $C$  に含まれる OSR を模式的に図 3 に示す．このように POI 間を結ぶグラフを VNG(visiting node graph) と呼ぶ．VNG は隣接する POI 間，又は POI と  $s$  や  $d$  を結ぶ道路網上の最短路をリンクとして構成される．集合  $C$  中の経路は同一のリンクを共有する．そこで，道路網距離での検証の際に，まだ VNG に存在しないリンク長は A\* アルゴリズム等により求め，既に VNG に登録されているリンク長は VNG を参照することにより得れば，道路網距離に要する計算時間を大幅に短縮することができる．

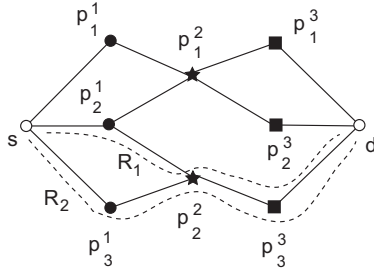


図 3: Visiting node graph

#### 4. 性能評価

ユークリッド距離での OSR 検索に関しては，文献 [3],[4] において性能評価がなされている．図 4 は，POI 数が 10,000 点からなる  $m$  個のカテゴリに対して，EOSR を求める際の実行時間と R 木ノードの参照回数を示している．この結果に見られるように，PNE では  $m$  の数が 5 程度までしか OSR 路を求められることができないが，EOSR では計算可能な  $m$  値の範囲を拡大している．

次に，ユークリッド距離により求められた候補に対する，道路網上での距離による検証結果について述べる．この実験では，道路網としてさいたま市内の道路地図 (24,914 道路セグメント) を用いた．POI セットは疑似乱数を用いて道路網上に様々な確率 (Prob) で発生させたものを用いた．ここで，Prob=0.01 とは，1 つの道路セグメント上に 0.01 個の割合で POI が存在することを表わしている．検索の始終点の距離 ( $L$ ) と Prob 値の間には密接な関係がある． $L$  が長くなれば低い Prob 値でも始終点を結ぶ最短路上に多数の POI が存在することになり，逆に  $L$  が短くなれば，高い Prob 値でも OSR 路は  $L$  に比べて長くなる．そこで，本実験では  $L$  値を約 10km に固定している．

図 5 に，その結果を示す．“w/o VNG” は EOSR により求めた OSR 候補を構成する全ての経路に対して A\* アルゴリズムにより探索を行った際のノード展開回数と計算時間

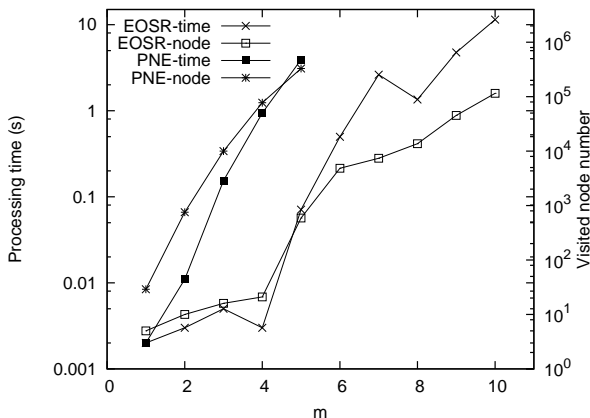


図 4: ユークリッド距離での実験結果

を，“VNG” は重複するリンクの距離を既に計算された値を用いた結果である．この図にみられるように，POI 密度が高く存在する場合，VNG を構成することにより，大幅に処理時間を短縮することができる．

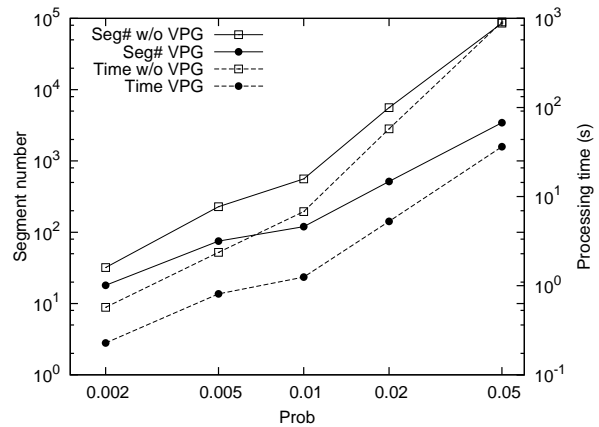


図 5: セグメント数

#### 5. まとめ

本稿では，道路網上の移動を想定した旅行計画を IER の枠組みで実行する方式を提案した．まず，ユークリッド距離上での OSR をインクリメンタルに検索可能なアルゴリズム EOSR を提案した．従来方式の内，インクリメンタルな検索が可能な PNE との比較により，旅行途中で経由する POI の種類が多い場合，及び POI が密度高く存在する状況で提案方式は PNE に大幅に優れることを示した．

次に，道路網上での距離による検証方式を検討した．POI が密度高く存在する状況では，ユークリッド距離上での検索により得られた経路間に重複が存在する．即ち，ある 2 つの POI 間の経路が多数の候補で重複する．道路網上での 2 点間の距離算出には時間を要するため，重複する 2 点間の距離の算出を抑制する方式を提案した．

本論文では，経路長が最短の 1 つの経路を求める場合のアルゴリズムについて示した．一般には距離が短い経路から  $k$  個 (任意個) の経路の提示が求められる場合も多い．本稿で示したユークリッド距離での候補生成アルゴリズムはインクリメンタルに適用可能である．また，道路網上での距離算出も基本的な考え方は  $k$  個の OSR を求める為に適用可能である．更に，OSR 以外の旅行計画，TPQ や MRPSR 検索に際しても本稿で提案した方式は適用可能である．

#### 参考文献

- [1] M. Sharifzadeh, M. Kalahdouzan and C. Shahabi: “The optimal sequenced route query”, Technical report, Computer Science Department, University of Southern California (2005).
- [2] D. Papadias, J. Zhang, N. Mamoulis and Y. Tao: “Query processing in spatial network databases”, Proc. 29th VLDB, pp. 790–801 (2003).
- [3] トウ, 大沢: “ユークリッド距離での制約をベースとする単純な旅行計画の為に高速アルゴリズム”, DEIM Forum 2012, pp. D8–1 (2012).
- [4] Y. Ohsawa, H. Htoo, N. Sonehara and M. Sakauchi: “Sequenced route query in road network distance based on incremental euclidean restriction”, DEXA 2012 (2012). (to appear).