D-020

道路網距離に基づく連続旅行計画
Continuous Trip Planning Query Based on Road Network Distance

Aye Su Yee Win[*] **大沢　裕**[*]
Aye Su Yee Win Yutaka Ohsawa

1. まえがき

現在位置から旅行の最終目的地に達する途中で，予め与えられたいくつかの施設（例えば，レストラン，ガソリンスタンドなど）を一つずつ訪れる際の最短経路を求める問題は，一般に旅行計画と呼ばれる．この問題は，途中で訪れる施設の種類が増えると急速に計算複雑さが増大するため，従来ユークリッド距離での近似解を求めるアルゴリズムが提案されてきた．

一方，実用的な観点からは道路網上を移動する際の最適，または準最適解が求められる．しかし，従来方式の内の多くは，この目的に適していない．そこで筆者らは以前に，まずユークリッド距離での候補経路を探し，それを道路網距離で検証する枠組み，IER(incremental Euclidean restriction) を採用することにより，道路網距離での旅行計画を高速に求める方式を提案した [1]．この方式は，道路網距離での旅行計画路を従来方式に比して高速に求めることができた．しかし，この方式では，途中で訪れる施設（以下 POI: point of interest と呼ぶ）が密度高く存在する場合において特に膨大な処理時間を必要とする問題があった．

そこで，本稿では処理時間を制限することにより，準最適な旅行計画路を指定時間内に算出して提示する方式を提案する．更に，移動体が最初の最適巡回路から逸れて移動した場合に，新たな現在位置からの旅行計画を高速に求める方式 (COSR) を提案する．実験により，1 秒程度の演算時間で準最適な経路が算出されることを示す．また，COSR では新たな最適巡回路が高速に検索できることを示す．本稿では，旅行計画の中で OSR(optimal sequenced route) 検索を中心に述べる．しかし，本稿で述べる方式は他の旅行計画にも直接適用可能である．

2.IER に基づく準最適解の検索

道路網距離に基づく旅行計画の高速な演算方式として，筆者らは IER の枠組みに基づく方式を既に提案した [1]．この方式では，まずユークリッド距離での OSR 路を距離が短いものから順にインクリメンタルに求め（ここで求まる経路を EOSR(Euclidean OSR) と呼ぶ），併せてそれらの道路網上での距離を求める．処理途中で，道路網距離での最短 OSR 路の長さが，ユークリド距離で次に短い OSR 路の長さより短くなった時点で，最短の OSR 路が決定するというものである．この方式は，筆者らの知る限りでは，高速に道路網距離での OSR 路の厳密解が得られる唯一の方式である．

一方，この方式では次の問題が生じる．途中に経由するそれぞれの POI カテゴリー中の要素数が多い場合，ユークリッド距離での探索において経路長の似た OSR が多数生成され，それらを道路網距離で検証するために膨大な処理時間を要するという問題である．一般に候補 OSR 路の道路網距離を求める演算はコストが高い．この問題を緩和するために，2 つの提案を行った．1 つは，同じ POI 間の道路網距離計算を省略する工夫であり，他の 1 つは POI カテゴリー間の密度が大きく異なるとき，密度の低い POI 種から探索する工夫である．前者については VPG(visited POI graph) と呼ぶグラフを作成しつつ OSR 路探索を行い，ある 2 点間の OSR 路が求まった時，その経路長を VPG に登録していき，もし既に VPG に登録されている 2 点間距離が必要になったときには，VPG から得る．また，後者により POI 種を訪れる順番によらず安定した旅行計画路の探索を行える．

しかし，依然として道路網距離での最短 OSR 路の探索には処理時間を要する．一方，IER による検索は常にユークリド距離での経路長が最短のものから順に求めることができるため，例えばある処理時間の上限を指定して，それを超えたときには処理を打ち切ることができる．ユークリッド距離が短い経路が常に道路網距離でも最短である保証はないが，両者の距離の間には強い相関が認められる．従って，検索を打ち切ることにより得られる経路は最適解ではないが，許容された時間内での準最適解となる．

3.COSR 検索

旅行計画を立て，それに沿って移動する際に，経路の見落とし，事故による通行止め，渋滞など様々な理由により当初立てた計画通りに移動できない場合がある．そこで，経路から逸れた場合に，再度現在位置からの旅行計画を立てる必要がある．しかし，再度の計算には再び長い処理時間を要する．

一方，IER の枠組みではユークリッド距離上での検索に用いる PQ の内容や，道路網距離での検証の際に得られる VPG を再利用することにより新たな旅行計画を高速に求めることができる．

初期の現在位置を s，最終目的地を d とし，m 種類の POI($C_1, \ldots, C_m$) を訪れる OSR 検索を考える．EOSR を求める際に用いられる優先順位付きキューを PQ とし，求まった EOSR は集合 F に入れられるものとする．初期検索は，F を空にして実行する．

s が別の位置 s' に移動したとき，まだ C_1 カテゴリーの POI が訪れられていないとき，COSR は次の様に実行される．

(1) PQ 中の全てのレコードに対して，s' を開始点とする $Cost$ 値を計算しなおし，新たな PQ に追加する．全てのレコードに対してこの処理が終了後，新たな PQ を EOSR の為の PQ とする．

(2) F 中の全ての経路に対して s を s' に変更して $Cost$ 値を計算しなおした経路を PQ に投入する．

図 1 は $m = 2$ の場合の例を示している．この図では F には $s \to p_1 \to p_2 \to d$ の経路が入れられている．s が s' に移動したことにより，この経路の現在位置を s' として，$Cost$ を再計算して PQ に投入する．また，POI が確定していない経路 (C1 を経由する経路，C2 を経由する経路) はまだ PQ 中に存在するが，それらについても s' からの $Cost$ を再計算する．

一方，再計算の前に既に C_1 から C_k までの k 個の POI を経由している場合には，s' を新たな現在位置とし，最終目的地 d に至る途中で C_{k+1} から C_m までの POI 種を順に訪れる経路を同様に求める．この際に，F 中の経路から C_k までの部分を取り除き，新たな現在位置を s' とした経路を PQ に投入する．また，PQ 中のレコードに対しても同様な変更を加えて新たな PQ を構築する．

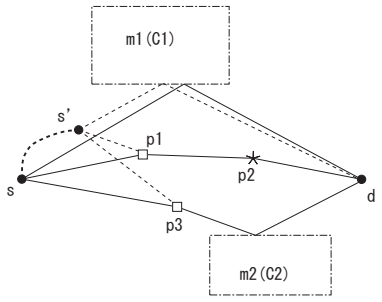

図 1: PQ レコードの $Cost$ の再計算

[*]埼玉大学

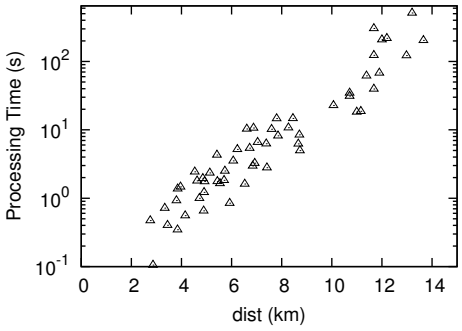

図 2: 厳密解の計算時間

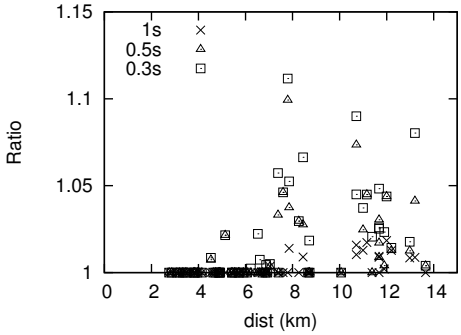

図 4: 近似 OSR 路の誤差比較

4. 性能評価

前節で述べた提案方式を Java で実現し，Intel Corei7CPU(3.2 GHz)(9GB メモリー) 上で実験を行った．使用した道路地図は，約 168km^2 の範囲の道路地図 (16,284 ノード，24,914 リンク) である．検索対象とする POI は道路リンク上に疑似乱数により発生させた．

図 2 は，途中に経由する POI 種の数 (m) を 3 に，各 POI の存在確率を 0.01 に設定した際に，最適な OSR 路を得る際の処理時間を示したものである．ここで，POI の存在確率は，1 道路セグメント当たりに POI が存在する確率を示している．横軸は求まった OSR 路の長さを，縦軸は処理時間を示している．この図にみられるように，処理時間は OSR 路長が長くなるに従い急速に増大する．

図 3 は，図 2 を求める際に発生させたユークリッド距離での OSR 候補数を示したものである．処理時間が増大する理由は，ここに見られるように OSR 路が長くなるに従い膨大な数の OSR 路をユークリッド距離で求め，それを道路網距離で検証する必要がある為である．

POI が密度高く存在する場合に，多数の OSR 路候補を生成し，その距離を検証する必要があるが，逆に密度高い場合にはユークリッド距離最短の経路と道路網距離最短の経路は似たものとなる．即ち，IER の枠組みでの生成・検証を一定時間で打ち切っても，その時点で得られる OSR 路長は最短 OSR 長に比して大きく離れてはいないものと推測できる．図 4 は，演算時間を 0.3s，0.5s，1.0s で打ち切った場合に，その時点での最短路と真の最短路との長さの比を示している．ここに見られるように，処理時間を 0.3s に設定した場合には，最適路の 1.1 倍程度以下に，また 1.0s に設定した場合には 1.02 倍程度以下の経路が求められていることが分かる．例えば，12km 程度の長さの最適 OSR 路を求める際には 10s から 100s 程度の処理時間を要しているが，わずかな経路長の増大を許せば 1 秒以下の処理時間で準最適解が得られることが分かる．

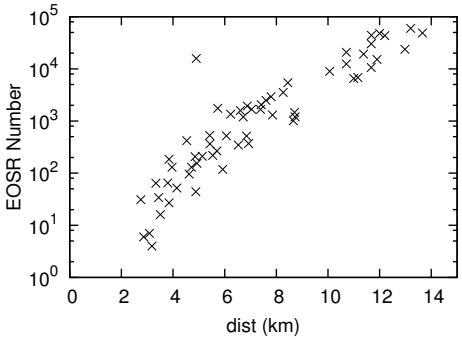

図 3: 発生された候補数

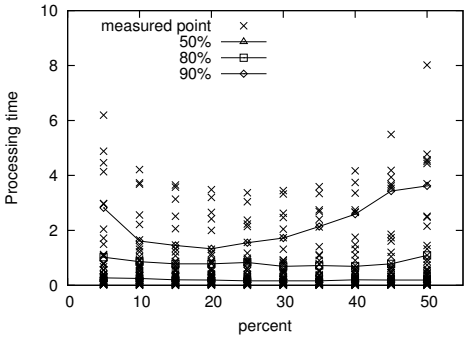

図 5: COSR 検索の処理時間

図 5 は，現在位置 s から最終目的地 d への OSR 路の探索を行った後で，$s-d$ 間の最短路上を p%進み，そこで再度 OSR 路の探索を行う際の処理時間を示している．横軸は p%を縦軸は処理時間を示している．また，×印は個々の $s-d$ 対の処理時間を示し，また処理時間が短いものからそれぞれ 50%,80%,90%の累積値を示すラインを示している．ここに見られるように，COSR 路の探索には最長 8 秒程度かかる場合もあるが，p の値に関わらず 80%のサンプルにおいて 1 秒程度の処理時間で新たな OSR 路が探索できていることが分かる．

5. まとめ

本稿では，道路網上での旅行計画問題を対象として，IER の枠組みで近似解を高速に求める方式を提案した．更に，最初に検索された経路を無視して進行した後で，再度旅行計画路を求める際に，最初の検索で得られた情報を用いて高速に再計算を行う方式（COSR 検索）を提案した．

準最適路の検索については，最適値を求める際の処理時間に比して 1/10 から 1/100 程度の処理時間で最適値より 2%以下の距離増加の準最適解が求められることを示した．また初回の OSR 探索時のデータを保持することにより，提案経路から逸れた際の OSR 路を高速に再探索できることを示した．

参考文献

[1] 大沢, トウ, 曽根原, 坂内：“道路網距離での旅行計画方式の為のインクリメンタル検索方式”, DBSJ Journal, **11**, 2, pp. 1 –6 (2012).