

道路網における最短寄り道経路検索

大沢 裕, 藤野 和久, 油井 真斗

A Shortest Detour Path Search Method on Road Network

Yutaka OHSAWA, Kazuhisa FUJINO, Masato YUI

Abstract: This paper proposes a new type of path search algorithm for LBS (location based services). Car or human navigation systems usually search a cost minimum path connecting a start point and a destination. However, sometimes we are apt to find a detour path dropping in some places, for example restaurant or amusement park during a travel, but the total path length being minimum. This paper names such kinds of search k-SD (k-shortest detour) path, then proposes an algorithm, and evaluates it.

Keywords: 位置情報サービス (location based service), 経路探索 (path finding), POI (point of interest), k-SD 検索 (k-shortest detour search)

1. はじめに

カーナビの普及は目覚ましい。カーナビは GPS により現在位置を取得でき、道路網の地図を備えており、現在位置から目的地までの最短経路探索を行える。一部では、携帯電話等の通信機能との連動により、Web 端末としての利用も可能になっている (例えば、G-BOOK)。カーナビは携帯電話と比較して大きな画面を有するため、将来的には情報ブラウザ端末としての役割も重要となるものと予測される。本稿では、カーナビを情報端末として利用することを前提とした、LBS (location based service) における新しい POI の探索方式について述べる。

いま、カーナビに予め目的地を入力して走行中

りたいとする。あるいは、時間的な余裕があり、目的地までの途中で美術館や映画館、名所などを訪れたいとする。これらの施設等を本稿では POI (point of interest) と呼ぶことにする。ここでカーナビに立ち寄りたい場所の種類や条件を入力し、寄り道を含めた総計のコスト (時間や距離) が最少となる POI を探したい。指定した条件を満たすものが複数ある場合もあり、コストが小さなものから k 個選んでカーナビ上に表示し、その中から選択したい。本稿では、このような検索を k-SD (shortest detour) 探索と呼ぶことにする。

次章で述べるように、従来 LBS 分野では、POI を対象とした k-NN (nearest neighbor) 検索のアルゴリズムが各種提案されてきた。これは、道路ネットワーク上を移動する制約の下に、現在位置から最も近い POI を探索するものである。本稿で扱う問題は、「現在位置から条件を満たす複数の POI の内の 1 つを経由して目的地に至る最短経路を与える POI を k 個コスト順に求める」検索であり、従来の k-NN 検索とは本質的に異なるものである。

2. 関連研究

大沢 : 〒338-8570 埼玉県さいたま市桜区

下大久保 225 埼玉大学大学院理工学研究科

TEL(FAX):048-858-3717

email:ohsawa@mail.saitama-u.ac.jp

であり、途中でレストランや、コンビニに立ち寄

GIS で必要になる空間検索法として、範囲検索、k-NN 検索、空間ジョイン、最近接ペア検索などを対象として多くのアルゴリズムが提案されている。しかし、その多くはユークリッド距離をベースとしたものである。一方、LBS ではしばしば道路ネットワーク上での移動が前提とされる。その場合、道路ネットワーク上での距離を対象とした上記の各種検索が必要となる[4]。

カーナビやマンナビで必要となる検索に、道路ネットワークを対象として、ネットワーク上の 2 点間を結ぶ最短経路を求める検索があるが、これに関しては、Dijkstra のアルゴリズム[2]や A* アルゴリズム[3]の各提案をはじめとして、古くから多くの研究が行われている。また 2000 年代に入り、道路ネットワーク上での距離を対象とした k-NN 検索や最近接ペア検索のアルゴリズムが各種提案されている。

LBS では 2 種類の情報が扱われる。1 つは、道路などの基盤を記述したネットワーク情報であり、他の 1 つは POI に関する情報である。前者は比較的变化が少なく安定している。一方、後者の変更頻度は前者に比べて高い。この性質から、両者を独立して管理するのが都合が良い。そこで、Papadias ら[4]は POI と道路ネットワークを分離して管理する環境で、道路に沿って近接の POI を求めるアルゴリズムを提案している。本稿でも、道路ネットワークと POI 情報は独立して管理されている状況を前提とする。

3. 問題に対する考察

最初に、問題を単純化するために経路探索の出発点 N_s と目的地 N_e を道路ネットワークにおけるノード（交差点）に限定する。また求める POI の個数を 1 個に限定する。即ち 1-SD を求めるアルゴリズムについて述べる。但し、後にこの制約は解消される。

まず 1-SD 探索問題について明確にするために、図 1 を用いて段階的に解法を考える。このアルゴリズムを以下に示す。

【アルゴリズム：1-SD を求める素朴な方法】

[Step1] N_s と N_e を結ぶ最短経路 P_1 を求める。

[Step2] P_1 上の全てのノード群を開始点とする Dijkstra 法により徐々に探索範囲を拡大しつつリンク上の POI(q)を求める。

[Step3] N_s から q を経由して N_e に至る最短経路 P_2 を求め、その経路長を D_2 とする。

[Step4] 再び、 N_s から N_e までの経路周辺のリンクを終了条件を緩めた（即ち最短経路が求まった後もノード展開を続ける）最短経路探索法により探索しつつ、他の POI を探す。より経路長が短い POI が見つかったときは解をその POI で更新する。

Step1 では道路ネットワーク上で N_s と N_e を結ぶ最短経路を（例えば）Dijkstra 法により探索し、見つかった最短経路の構成点を

$$P_1 = \{n_0 = N_s, n_1, \dots, n_{k-1}, n_k = N_e\}$$

とする。また、その経路上での距離を D_1 とする。

もし、探索しようとする POI が経路 P_1 上に存在するとき、それが 1-SD となる。1-SD を求めれば十分な場合には、これで処理を終了する。一方、POI が P_1 上に存在しない場合には、Step2 で P_1 から道路ネットワーク上で等距離となる領域を徐々に拡大しつつ、道路セグメント上に存在する POI(これを q とする)を求める処理を行う。これは、 P_1 上の全ての点を開始点とする場合の Dijkstra 法と等価である。

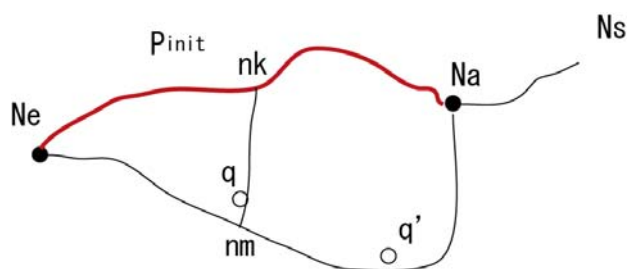


図 1 寄り道検索の例

ここで、POI は道路ネットワークと密に結合せず、それぞれを独立に管理することが望ましいと考える。通常道路ネットワークは多目的に用いられる情報であり、アプリケーションに依存した情報とは独立に管理されるべきである。また POI はユーザの興味対象により検索時にそのレイヤ

が指定される情報であり、また生成消滅の頻度が高いことから両者は独立していることが望ましい。

このため、本稿では道路ネットワークとPOIがそれぞれ独立の空間インデックスで管理されているものとして扱う。道路網上で隣接する2つのノード n_i と n_j を直接結ぶリンクを $n_i n_j$ とし、あるPOIを q したとき、双方の空間インデックス間で次の2種類の検索が行えるものとする。

- $n_i n_j = \text{find_segment}(q)$: POIの位置が与えられ、そのPOIが存在する道路セグメントを求める演算。
- $S_{\text{POI}} = \text{find_entities}(n_i n_j)$: ある道路セグメント上に存在するPOIの集合 S_{POI} を求める(有無確認も含む)演算。

また、ある道路セグメントの端点 n において、その端点に直接接続している道路セグメントの集合 S_n を容易に求められる構造で、道路セグメントが管理されているものとする[1]。

本稿では、`find_entities` 関数により、各道路セグメント上にPOIが存在するか否かを確認し、もし存在する場合にはそのPOIが得られるものとする。

Step3の処理が必要な状況を、図1を用いて説明する。この図で、太線は N_s と N_e を結ぶ最短経路 P_1 示している。Step2の処理により N_s , n_k , q , n_k , N_e という寄り道経路が求まることになるが、実は N_s , n_k , q , n_m , N_e という経路の方が短いかもしれない。つまり、見つかったPOIを経由する最短経路を求めるのがStep3の役割である。

更に、図1では、 N_s , N_a , n_m , q , n_m , N_e という経路がStep3の経路より短い場合も考えられる。このような経路におけるPOIを探索するのが、Step4の役割である。

[定理] Step4で N_s からの距離が D_2 を超えるまで最短経路探索法を適用することにより、 D_2 以下の距離の寄り道に存在するPOIが見つかる。

[証明] 最短経路を求めるアルゴリズムとして、Dijkstra法を用いるものとする。また、そのアル

ゴリズムで次に展開すべきノードを決定するために、 N_s からノードまでの距離を優先度とした優先順位付きキュー、 PQ を用いるものとする。Step4で最後に展開されるノードとして PQ から n が取り出された時、その N_s からの距離は D_2 を超えている。つまり、この時点で N_s から D_2 以下の距離で到達できるノードまでのパス上に存在する全てのPOIは検出されていることになる。

従って、 D_2 以下の距離の寄り道に存在する全てのPOIが見つかることになる。

4. 提案アルゴリズム

前章の準備の下に、 k -SD 検索のアルゴリズムを提案する。

前章で述べた処理の流れの内、Step2の役割はほとんどない。このステップが意味をもつ状況としては、探そうとする種類のPOIが非常に密度高く存在しており、Step1で見つかった最短経路そのもの、またはごく少ない距離の寄り道をする事により目的とするPOIが存在する状況である。一方、POIの密度が低い場合には、このStep2は意味をなさない。そこで、以下で述べるアルゴリズムからはStep2を省略する。

また、SD 経路を求める演算が実際に用いられる状況を考えると、寄り道の経路長は、最短経路長と比較してある程度の倍率以内であることが求められるであろう。いくら距離が長くなってもかまわないという状況は少ないであろう。以下では、寄り道の距離増加が最短経路の f 倍以下となる経路を求めている。

更に、以下のアルゴリズムでは、3つの集合 U , V を用いる。 U は、Step1で N_s から開始するDijkstra法により見つけたPOIを入れる集合であり、 V は N_s および N_e からの距離が確定したPOIを入れる集合である。

[k -SD アルゴリズム]

[Step1] Dijkstra法により、 N_s から N_e に至る最短経路を求める。その最短経路の距離を D_{sp} とする。Dijkstra法でノード展開を行う際に、各リンク上

でのPOIの存在を確認し、もしPOIが存在すれば、Nsからの経路とともにその距離を記録したレコードを集合Uに入れる。Neまでの最短経路が求まった後、Dijkstra法によるノードの展開をNsからの距離が $f \times D_{SP}$ となるまで続ける。

[Step2] NeからDijkstra法を適用しつつ、リンク上のPOIを探索する。もし、POIが見つければ、そのPOIが集合Uに含まれるか調べる。もし、Uに含まれれば、そのPOIを含む寄り道長が確定した。その結果をVに入れる。Uに含まれない場合は、そのPOIを経由する寄り道の距離が $f \times D_{SP}$ 以上になるため、棄却する。

[Step3] Vに含まれる要素を経路長が短いものから順にk個提示する。

4. 実験

前節で述べたアルゴリズムを実現して、k-SD経路の探索実験を行った。POIは疑似乱数を用いて、道路セグメントに確率 p で存在するように発生させた。図2,3では p を0.2としており、POIを青点で示している。

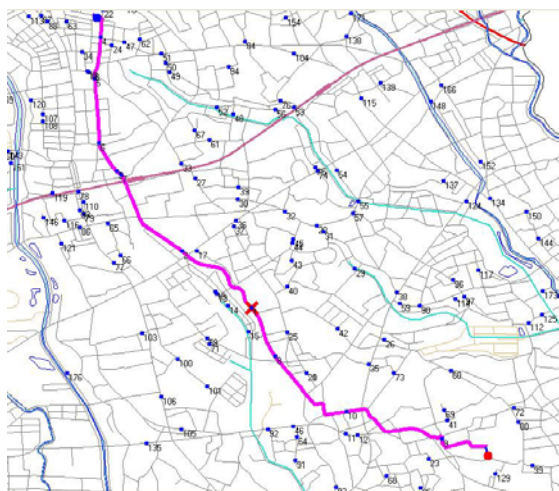


図2 1-SDの検索結果

図2の太線は1-SDの経路を示している。またここで見つかったPOIを×印で示している。ここでは、POIの密度が高く、最短路上に1-SDが存在している。また図3は32番目のSDを示している。この寄り道経路の長さは最短路に比して3.2%程度長くなっている。

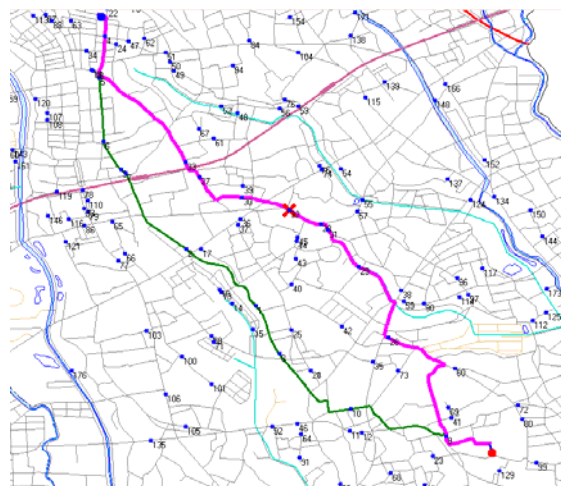


図3 32番目のSD経路

5. まとめ

本稿では、LBSでの利用を目的に、POIへの寄り道を含む場合の最短経路探索アルゴリズム(SD: shortest detour)について提案した。本稿で述べたアルゴリズムにはDijkstra法を用いているため、展開ノード数が多くなる欠点がある。本方式にA*アルゴリズムを用いるように改良することは容易であり、その際に処理効率の向上が期待できる。これらの検討は今後の課題である。

参考文献

- [1] 油井真斗, 大沢 裕, 「道路ネットワークに沿った最近接検索の効率化」, 第17回地理情報システム学会講演論文集, 2008
- [2] E.W.Dijkstra, “A note on two problems in connections with graphs”, *Numeriche Mathematik*, Vol.1, pp.269-271, 1959
- [3] P.E.Hart, N.J.Nilsson, B.Raphael, “A formal basic for the heuristic determination of minimum cost paths”, *IEEE Trans. Systems Science and Cybernetics*, Vol.SSC-4, No.2, pp.100-107, 1968
- [4] D. Papadias, J. Zhang, N. Mamoulis and Y. Tao. “Query Processing in Spatial Network Databases”, *Proceeding of the 29th VLDB Conference*, 2003.