

デジタル地図における経路探索法的高速化の検討と評価

2T-5

浜田ちぐさ* 瀬戸洋一* 萩原義裕* 北澤修司**

* (株) 日立製作所システム開発研究所 ** (株) 日立製作所システム事業部

1 まえがき

近年、カーナビゲーションなど地図上でシミュレーションを行うニーズが高まっている。道路網のある地点から一定時間内に到達することのできる地域を経路探索によって求める場合、従来のDijkstraのアルゴリズムでは、計算時間がかかる。そこで高速経路探索アルゴリズムの検討を行った。提案アルゴリズムはDijkstraのアルゴリズムにおけるノードの最短ルートの判定処理を省略することにより高速経路探索を可能にし、また、精度の劣化を抑えるため、隣接ノードデータを通過時間の順にソーティング処理を行う。そこで、提案アルゴリズムを開発し、速度と精度の評価を行う。

2 高速経路探索アルゴリズム

(1) 概要と問題点

高速アルゴリズムはDijkstraのアルゴリズムのノードにおける最短ルートの判定処理を省略することにより高速化を図る。本アルゴリズムについて図1を道路ネットワークの例を用いて説明する。図1で、○はノード、ノードとノードを結ぶ直線はリンクを表し、ノード n_i と n_j を結ぶリンクの通過時間を $d(i,j)$ で示す。始点ノードを n_0 とし、制限時間内での経路探索を行う場合、初めに、始点ノード n_0 までの通過時間の和を0とする。次に、隣接ノード n_1, n_2, n_3, n_4 について始点ノード n_0 からの通過時間の和を求める。さらに、 n_1 の隣接ノード n_5, n_6, n_7 について始点ノード n_0 からの通過時間の和を求める。このようにして、始

点ノードから順次隣接ノードを検索して、通過時間の和を求める。既に通過時間の和を求めているノードを検索した場合は通過時間の和の計算を行わない。

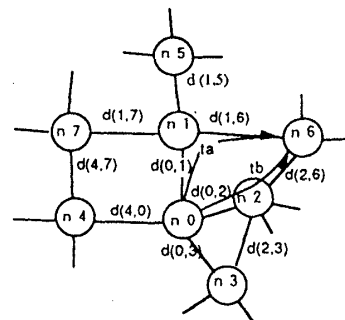


図1 道路ネットワークの例

この高速経路探索アルゴリズムでは始点ノードから順次機械的にノードの通過時間の和を求めるので、先に求めた通過時間の和が必ず優先され、精度劣化がおこる。例えば図1で、ノード n_6 の通過時間の和 t は、ノード n_1 を経由する場合 t_a とノード n_2 を経由する場合 t_b が考えられるが、ノード n_1 を経由する経路で通過時間の和を求めるので、通過時間 t は

$$t = t_a (= d(0,1) + d(1,6)) \quad (1)$$

となる。しかし、

$$t_a > t_b (= d(0,2) + d(2,6)) \quad (2)$$

であるとき、ノード n_2 を経由した方が短時間で到達可能であり、ここに精度の劣化が生じる。

(2) 精度の改善

上述のアルゴリズムでの精度劣化の原因は、必ずしも最短経路を求めていることにある。そこで、最短経路を求めるために、以下の処理を行う。図2でノード n_0 の隣接ノード $n_1, n_2,$

The Examination and Evaluation a fast Route Search algorithm on Digital Map
 Chigusa HAMADA, Yoichi SETO, Yoshihiro HAGIHARA, Syuji KITAZAWA
 * Systems Development Laboratory, HITACHI, Ltd
 ** Public Systems Dept. Systems Engineering Div. HITACHI, Ltd

n_3 への通過時間が、

$$d(0,2) > d(0,3) > d(0,1) \quad (3)$$

であるとき、隣接ノードの順序を n_3, n_1, n_2 の順序に並び換える。この隣接ノードのソーティング処理を全てのノードに対して行うことにより精度の向上を図る。

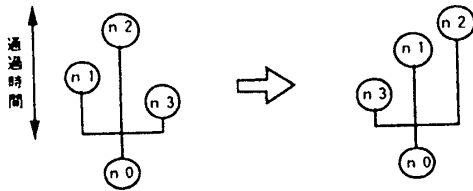


図2 データソーティングの例

3 評価方法

対象とするネットワークは、財団法人日本デジタル地図協会の全国デジタル道路地図データベースから神奈川県周辺の基本ノードデータを経路探索用に抽出したものである。ノード数は40241個、無向リンク数は74552本である。提案法である高速経路探索アルゴリズムとDijkstraのアルゴリズムを評価した。道路通行速度は毎時30kmとして行った。コンピュータには日立の3050R(57MIPS)のワークステーションを用いた。

4 実験結果および検討

図4にリンク数に対する計算時間の評価を示す。提案法の計算時間は、測定したデータの平均で、Dijkstraのアルゴリズムの約6分の1であった。

また、各手法による精度評価結果を表1に示す。制限時間20分での経路探索を神奈川県内3か所で行った結果の平均の探索リンク長の総和から精度を求めた。Dijkstraのアルゴリズムによって求めた探索リンク長を100%とした時の探索リンク長の割合を精度とした。高速ア

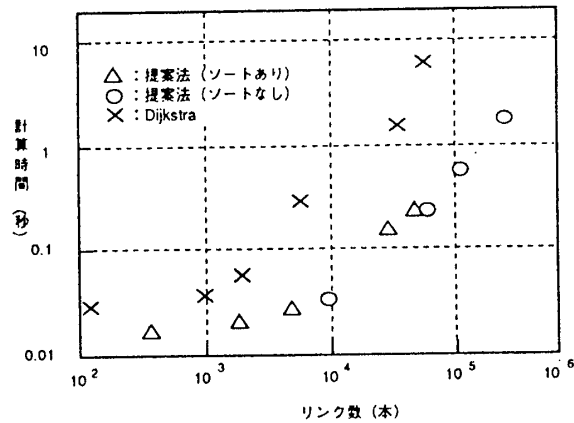


図3 リンク数と計算時間の関係

表1: 各手法による精度評価

方法	リンク数	リンク長	精度 (%)
提案法 (ソートあり)	5628	102819	77.6
提案法 (ソートなし)	5554	101182	71.3
Dijkstra	7528	141966	100.0

ルゴリズムだけで経路探索を行った結果、71.3%の精度であったが、ソーティング処理を施すことによって77.6%まで精度を上げることができた。

5 むすび

隣接ノードデータを通過時間の順にソーティング処理を行う高速経路探索アルゴリズムデータを提案した。本アルゴリズムとDijkstraのアルゴリズムを比較すると、計算時間は約6分の1、精度は77.6%を得ることができた。

参考文献

[1] 石畑清, アルゴリズムとデータ構造, 4章, 岩波書店, (1989).