

デジタル地図における多段最短路探索アルゴリズムの提案

3T-9

徳永稔* 瀬戸洋一* 浜田ちぐさ* 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求めるには、デジタル地図上の道路データを用いて、通行時間をコストとした最短路探索を行えばよい。最短路探索法のひとつに Dijkstra のアルゴリズム [1] がある。これは精度は完全であるが、計算時間がかかる問題がある。それに対処するために以前提案した高速経路探索アルゴリズム [2] は、Dijkstra のアルゴリズムに比べて計算時間は短い、精度が落ちる。そこで本稿では、これら二つのアルゴリズムを組み合わせた、高精度で高速処理が可能な多段最短路探索アルゴリズムを提案し、計算時間と精度の評価を行う。

2 多段最短路探索アルゴリズム

道路網として図1のような無向ネットワークを考える。図1において、○はノード、ノードとノードを結ぶ直線はリンクを表し、ノード番号を○の中に数字で、ノード i とノード j を結ぶリンクの通行時間を $t(i, j)$ で示す。

すでに提案した高速経路探索アルゴリズムは、Dijkstra のアルゴリズムの最短路の判定処理を省略することにより、計算時間の短縮化を図っている。これは、始点ノードから順次隣接ノードを探索していく際に、始点ノードからの通行時間は一度計算した時点で確定し、別の経路からの計算を行わない、というものである。このため、高速経路探索アルゴリズムは Dijkstra のアルゴリズムに比べて、計算時間はかなり短い、精度が落ちる。

そこで、計算時間は Dijkstra のアルゴリズムより短く、かつ、精度は高速経路探索アルゴリズムより高く

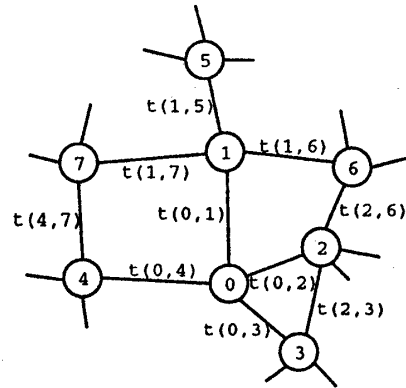


図1 無向ネットワーク

することを目的として、これら二つのアルゴリズムを組み合わせた多段最短路探索アルゴリズムを提案する。

多段最短路探索アルゴリズムは、あるノードに適用するアルゴリズムを、始点ノードからそのノードまでの通行時間 t_s によって切り替える。ここでは、制限通行時間 t_l での最短路探索を行う場合の、つぎの二つを考える。

- (1) Dijkstra のアルゴリズム→高速経路探索アルゴリズム (提案法 A)
 - (i) $0 \leq t_s < t_l \cdot \alpha$ のノードでは Dijkstra のアルゴリズムを適用
 - (ii) $t_l \cdot \alpha \leq t_s < t_l$ のノードでは高速経路探索アルゴリズムを適用
- (2) 高速経路探索アルゴリズム→Dijkstra のアルゴリズム (提案法 B)
 - (i) $0 \leq t_s < t_l \cdot \alpha$ のノードでは高速経路探索アルゴリズムを適用
 - (ii) $t_l \cdot \alpha \leq t_s < t_l$ のノードでは Dijkstra のアルゴリズムを適用

ただし α は、はじめのアルゴリズムを適用するノードの t_s の上限の、 t_l に対する比率である。また、 $0.0 \leq \alpha \leq 1.0$ であり、提案法 A は $\alpha = 0.0$ のときは高速経路探索アルゴリズムに、 $\alpha = 1.0$ のときは Dijkstra のアルゴリズムに相当し、提案法 B は $\alpha = 0.0$ のときは

A Proposal of Multistage Shortest Path Search Algorithm on a Digital Map

Minoru TOKUNAGA*, Yoichi SETO*, Chigusa HAMADA*, Shuji KITAZAWA**

*Systems Development Laboratory, HITACHI, Ltd.

**Systems Engineering Div., HITACHI, Ltd.

Dijkstra のアルゴリズムに、 $\alpha = 1.0$ のときは高速経路探索アルゴリズムに相当する。

3 評価

提案法 A および提案法 B について以下の方法で計算時間と精度の評価を行った。

使用したデジタル地図データは、(財)日本デジタル道路地図協会の全国デジタル道路地図データベースから、神奈川県の基本データを経路探索用に抽出したもの(ベクター形式)である。ノード数は 26773 個、無向リンク数は 36267 本である。リンク通行時間はリンク長と規制速度から計算して求めた。

計算には、メインメモリ 48MB を実装した日立のワークステーション 3050/R(57MIPS)を使用した。

これらの条件のもと、提案法 A および提案法 B について、同じ始点ノードで制限通行時間を 30 分として、 α を 0.0 から 1.0 まで 0.1 ずつ変化させたときの計算時間と精度を求めた。なお、計算時間は、100 回同じシミュレーションを行ったときの 1 回あたりの平均値(有効数字 2 桁)であり、精度は、提案法 A、提案法 B でそれぞれ $\alpha = 1.0$ 、 $\alpha = 0.0$ のときの総探索リンク長における比率(有効数字 2 桁)である。

4 実験結果および検討

図 2 に、評価の結果を精度と計算時間の関係として示す。計算時間については Dijkstra のアルゴリズム(提案法 A では $\alpha = 1.0$ のときに、提案法 B では $\alpha = 0.0$ のときに相当)より小さく、精度については高速経路探索アルゴリズム(提案法 A では $\alpha = 0.0$ のときに、提案法 B では $\alpha = 1.0$ のときに相当)より優れた結果が得られた。このことから、多段最短路探索アルゴリズムは高速かつ高精度なアルゴリズムとして有効であるといえる。

また、図 2 から、同じ精度を得るために必要な計算時間は提案法 A の方が提案法 B より短く、同じ計算時間で得られる精度は提案法 A の方が提案法 B より高いことがわかり、提案法 A は提案法 B より優れているといえる。そして提案法 A で、計算時間と精度のどちらをどれほど重視するかにもよるが、高速経路探索アルゴリズムとの計算時間の差と Dijkstra のアルゴリズムとの精度の差が同時になるべく小さくなるような α を選

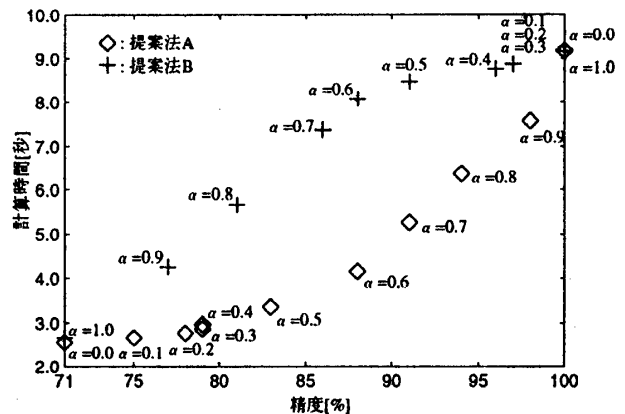


図 2 α を変化させたときの精度と計算時間の関係

択するのが最適である。

以上から、つぎのことが考えられる。始点ノード付近に高精度な Dijkstra のアルゴリズムを用いることにより、周辺部に波及する精度誤差はない。このとき、Dijkstra のアルゴリズムによって探索されるリンク数が総探索リンク数に占める割合は小さいため、計算時間はそれほど大きくなる。また、始点ノード付近に比べてリンク数が増える周辺部には計算時間が短い高速経路探索アルゴリズムを用いることにより、最短路探索に要する時間は抑えられる。そして、ここで生じる精度誤差は他に大きく影響しない。したがって、計算時間をそれほど増加させずに精度の向上が可能になり、精度をそれほど劣化させずに計算時間の短縮が可能になる。

5 むすび

Dijkstra のアルゴリズムと高速経路探索アルゴリズムを組み合わせた多段最短路探索アルゴリズムを提案し、計算時間と精度の評価を行った。多段最短路探索アルゴリズムの計算時間は Dijkstra のアルゴリズムより小さく、かつ、精度は高速経路探索アルゴリズムより優れており、その有効性を確認できた。

参考文献

- [1] 石畑: “アルゴリズムとデータ構造”, 岩波書店, 1989.
- [2] 浜田, 瀬戸, 萩原, 北澤: “デジタル地図における経路探索法の高速化の検討と評価”, 情報処理学会第 48 回全国大会, 2T-5, 1994.