ダイクストラ法の拡張による経路探索の効率化に関する研究

能登研究室　　佐藤 弘章 (46244)

1　はじめに

経路探索は，現在地から目的地まで与えられた条件下で最適経路を求める問題である．現在，道路網などのネットワークに使われ，カーナビゲーションなどにも応用されている．しかし，実際のカーナビゲーションで求めた経路は最適なものとは限らない．そこで，本研究では経路探索で最適解を求める事のできるダイクストラ法を基とし，それを拡張する事で，高速かつより最適解に近い経路を求める手法を提案する．

2　ダイクストラ法

現在，経路探索で利用，研究されているアルゴリズムにはダイクストラ法，A^*アルゴリズム，遺伝的アルゴリズムなどがある．

まず，道路の交差点を節点 (node)，節点と節点をつなぐ道を経路 (route)，その道の長さや通りやすさなどを移動コスト (cost) とする．

ダイクストラ法は出発点から順に最も移動コストが小さい節点を探していくので，最適解 (経路) を求める事ができる．アルゴリズムを以下に示す．

アルゴリズム

- Step0：出発点に印を付ける．
- Step1：出発点ににつながっている節点の，出発点–節点間の移動コストを求め，最小の値を持つ節点に印を付ける．
- Step2：印が付いた節点につながる節点の出発点からの移動コストを求め，最小の値を持つ節点に印を付ける．
- Step3：Step2 を目的地に印が付くまで繰り返す．

ここで得られた値が目的地までの最小コストである．また，節点に印を付けるときに，前の節点を記憶させておく事により，最短経路を求められる．

3　ダイクストラ法の拡張

従来のダイクストラ法は，一般的に，探索領域は同心円状に広がってしまうので，一回の探索で調べる節点の数は多くなってしまい探索時間が長くなってしまうのが欠点である．

そこで，出発点と目的地の両方からダイクストラ法を適用し拡張する事を提案する．このことにより，探索領域が同心円状に広がる事が押えられ，探索する節点の数を減らすことができる．以下に，ダイクストラ法を拡張したアルゴリズムを示す．

アルゴリズム

- Step0：出発点，目的地に印を付ける．
- Step1：出発点につながる節点なら出発点から，目的地につながる節点なら目的地からの移動コストを求め，最小の節点に印を付ける．
- Step2：印の付いた節点とつながる印の付いていない節点から出発点又は目的地までの移動コストの中で最も小さい節点に印を付ける．
- Step3：Step2 を繰り返し，出発点からの探索と目的地かの探索が重なりあった時終了する．

この方法で経路探索する場合にも探索範囲が広すぎるとダイクストラ法を基にしているので時間がかかってしまう．そこで探索時間または探索範囲がある一定値を越えた場合，そこでこの探索を終了させ，残った探索領域を近似して，そこに遺伝的アルゴリズムを用いることにする．近似の仕方としては，残った探索領域の両端をそろえ，そこに探索済領域の外側の節点 (出発点からと目的地から) を列べる．

4　シミュレーション

従来のダイクストラ法と本研究で提案した拡張ダイクストラ法との比較を行う．探索領域としては碁盤目状の経路を作成し，各節点間の移動コストはランダムで決定する．

シミュレーションの結果，ダイクストラ法を拡張することにより探索する節点の数が約半分まで減り，その結果，探索時間も減少した．シミュレーションで得られた探索時間の比較を図 1 に示す．

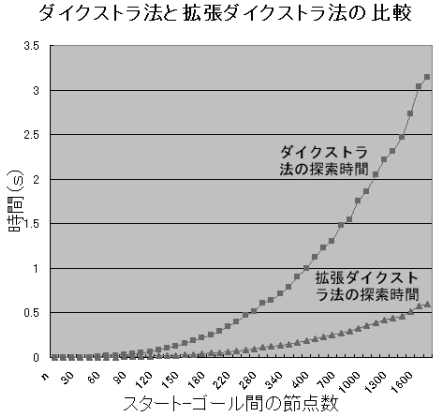

図 1: シミュレーション結果

5　おわりに

本研究で提案した拡張ダイクストラ法を用いることにより経路発見までに調べた節点の数は従来のダイクストラ法と比べ少なくなった．移動コストとしては，拡張ダイクストラ法では，必ずしも最適解が求まらなかったが，探索時間としては，経路にもよるが短縮された．

出発点と目的地からの探索の際，重なりあっているか否かという判定を行うため作業が複雑化してしまっているので，それを解決できればさらなる時間の短縮が見込めるものと考えられる．

また，出発点と目的地の間に通過点がある場合，そこからもダイクストラ法を適用することができるので，より効果的な探索が期待できる．