2009－AL－122（11）
2009／1／30

多数のランドマークを用いるための ALTアルゴリズム拡張

松永拓[†] 平手勇宇[††] 山名早人[†††,††††]

近年，汎用的なグラフ構造に対しての最短経路探索の高速化手法として，ALTアルゴリズムが提案されている．ALTアルゴリズムでは，ランドマークと呼ばれるノードと他のノードの距離を保存しておくことにより，A*探索における距離推定のためのヒューリスティック関数を提供する．しかし，ALTアルゴリズムでは，事前保存領域がランドマーク数増加に対して線形に増加し，多くのランドマークを設定することは，多くの事前保存領域を必要としてしまう． そこで本稿では，ALTアルゴリズムにおけるヒューリスティック関数を2ランドマークを用いて推定するように拡張し，ランドマークの追加による事前保存領域が線形には増加しない手法を提案する．道路ネットワークを用いた実験の結果，提案手法においてランドマークをランダムに選択した場合，ALTアルゴリズムにおいてランドマークをランダムに選択した場合よりも，少ない事前保存領域で，平均して少ない探索空間で最短経路を得るができることを確認した．

Extending ALT algorithm to use multiple landmarks

TAKU MATSUNAGA[†] YU HIRATE[††]
HAYATO YAMANA[†††,††††]

Recently, the ALT algorithm is proposed as a speed-up algorithm to compute shortest paths in general graph structures. The ALT algorithm offers a landmark based heuristic function to estimate distance in A* search. Before computing shortest paths, the ALT algorithm computes distances between all nodes and landmarks, and stores them to prepared memory or storage space. However, as the number of landmarks increases, the required prepared space increases linearly. To solve this problem, in this paper, we propose a novel heuristic function for computing shortest paths in general graph structures. Our approach for providing heuristic function uses two landmarks to decrease the size of prepared space so that prepared space will not increase linearly as the number of landmarks increases. For evaluation, we applied a road network graph structure to our proposed algorithm and the ALT algorithm. Our evaluation shows that our approach with random landmark selection requires less prepared space and less search space than the ALT algorithm with random landmark selection.

1. はじめに

最短経路探索は，道路ネットワークを応用対象として適用することが多く，高速化手法も主に地理上の特徴を用いた高速化手法が提案されてきた．汎用的なグラフ構造に対しての最短経路探索の高速化手法としてはGoldbergらがALTアルゴリズムを提案している[1][2]．ALTアルゴリズムでは，あらかじめ指定されたランドマークと呼ばれるノードと他のノードの距離を，予め計算し保存しておくことにより，A*探索[3]におけるヒューリスティック関数を提供する．ここで，A*探索におけるヒューリスティック関数は2点間の最短距離を推定し，下限値を返す関数である．しかし，ALTアルゴリズムでは，ランドマークと他のノードの距離を保存する事前保存領域がランドマーク数増加に対して線形に増加し，多くのランドマークを設定することは，多くの事前保存領域を必要とする問題点がある． そこで本稿では，ALTアルゴリズムにおけるヒューリスティック関数を提供する際，2ランドマークを用いて推定を行うように拡張し，ランドマークの追加による事前保存領域が線形には増加しない手法を提案する．

本稿では，以下，次のような構成をとる．第2節に最短経路探索アルゴリズムについて示す．第3節に提案手法について示す．第4節において実験条件を示し，第5節に実験結果を示す．第6節はまとめと今後の課題を述べる．

2. 最短経路探索アルゴリズム

本節では最短経路探索アルゴリズムである，Dijkstra探索[4]，双方向探索[5][6][7][8]，A*探索[3]，双方向A*探索[9]，ALTアルゴリズム[1][2]について示す．

† 早稲田大学大学院基幹理工学研究科
Graduate School of Fundamental Science and Engineering, Waseda University.
†† 早稲田大学メディアネットワークセンター
Media Network Center, Waseda University
††† 早稲田大学理工学術院
National Institute of Informatics
†††† 国立情報学研究所
National Institute of Informatics

2.1 Dijkstra 探索

Dijkstra 探索[4]は，始点ノードs，終点ノードtとした場合，探索は次の手順で行う．

1. 始点となるノードsに対し,ノードを探索するためのプライオリティキューのキーとなる$k(s) = 0$を設定し，始点以外のグラフの全ノードvに対して，$k(v) = \infty$を設定する．また，探索済みのノードを管理するためのノード集合Sを空集合で初期化する．
2. Sに含まれないノードのうち$k(v)$が最小のノードvを取得し，v をSに追加する．また，s からvへの最短距離を$d(s,v) = k(v)$として確定する．もし候補となるノードがなければ,探索は失敗として，処理を終了する．
3. もし，$v = t$であれば探索は成功として，処理を終了する．
4. ノードvから接続するノードに対して，$l(v,w)$をエッジ(v,w)の重みとしたときに，$k(w) > d(s,v) + l(v,w)$ であるようなノードwがあれば，$k(w) = d(s,v) + l(v,w)$に更新する．
5. 2から4を繰り返す．

2.2 双方向探索

双方向探索[5][6][7][8]は，s，t，の両端を始点ノードとし双方向から探索を行う手法である．探索は次の手順で行う．

1. s，t，を始点ノードとして交互に Dijstra 探索を行う．探索済みのノード集合はそれぞれS，Tとする．また，最短距離の上限を管理するための変数uを$u = \infty$ で初期化する．
2. 各探索において，各エッジ(v,w)に対して，$ui = d(s,v) + l(v,w) + d(w,t)$ を求め，$ui < u$となるuiがあれば，$u = ui$に更新する．この時，(v,w)を含む経路を最短経路の候補として保存する．
3. 各探索において，探索したノードvがS，Tの探索済みノード集合両方に含まれている場合,探索を成功とし，処理を終了する．

2.3 A*探索

A*探索[3]は，Dijkstra 探索における，$k(w) = d(s,v) + l(v,w)$ の代わりに,$k(w) = d(s,v) + l(v,w) + h(w,t)$ を用いる．$h(w,t)$ は，wからtへの最短距離の推定値を返すヒューリスティック関数であり，$0 \le h(w,t) \le d(w,t)$ を満たす場合，最短経路が求められることが保証される．

2.4 双方向 A*探索

A*探索と双方向探索を組み合わせる手法として，Ikeda らは，各方向の探索において，次のヒューリスティック関数を用いることを提案している[9]．π_f，π_rはそれぞれ，正方向，逆方向のヒューリスティック関数であり，それぞれの平均値を用いて，双方向A*探索のためのヒューリスティック関数P_f, P_rを構成することにより，A*探索においても，双方向探索における終了条件を保持する．

$$P_f(v,t) = \frac{\pi_f(v,t) - \pi_r(v,s)}{2}$$

$$P_r(v,s) = \frac{\pi_r(v,s) - \pi_f(v,t)}{2}$$

Goldberg ら[1][2]は，この手法を改良し，次式を用いることを提案している．

$$P_f(v,t) = \frac{\pi_f(v,t) - \pi_r(v,s)}{2} + \frac{\pi_r(t,s)}{2}$$

$$P_r(v,s) = \frac{\pi_r(v,s) - \pi_f(v,t)}{2} + \frac{\pi_f(s,t)}{2}$$

本稿では，実験において，双方向 A*探索の式として Goldberg らの式を用いる．

2.5 ALT アルゴリズム

ALT アルゴリズム[1][2]は，あらかじめ指定されたランドマークと呼ばれるノードと他のノードの距離を，予め計算し保存しておくことにより，A*探索におけるヒューリスティック関数を提供する．

ヒューリスティック関数は任意のノードv, w, lにおいて

$$d(l,w) - d(l,v) \le d(v,w)$$
$$d(v,l) - d(w,l) \le d(v,w)$$

が成り立つことを利用する．探索時には，A*探索において，ヒューリスティック関数 $h(v,w) = d(l,w) - d(l,v)$ あるいは $h(v,w) = d(v,l) - d(w,l)$ の否負の最大値を用いて，探索を行う．

図1は，ALT アルゴリズムにおけるヒューリスティック関数の概要図である．点線はノードv, wの最短距離の下限値である．l_fとv, wへの実線は，予め計算し保存したランドマークからノードへの実際の最短距離であり，同様にl_rとv, wへの実線はノードからランドマークへの実際の最短距離である．

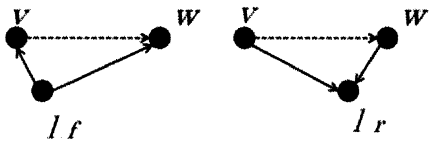

図 1. ALT におけるヒューリスティック関数

2.6 ALT アルゴリズムにおけるランドマーク選択手法

ALT アルゴリズムでは限られたランドマークのみを用いるため，ランドマークの選び方によってヒューリスティック関数の精度が変わる．したがって，良いランドマークを見つけることが重要である．Goldberg らは，いくつかのランドマーク選択手法を提案している[1][2]．本稿では，実験において ALT アルゴリズムにおけるランドマーク選択方法として，ランダム選択[1]，FarB 選択[2]，Avoid 選択[2]，MaxCover 選択[2]を比較対象として用いる．

2.6.1 ランダム選択

ランダム選択[1]は，ノード集合からランダムにランドマークを選択する手法である．

2.6.2 FarB 選択

FarB 選択[2]は，すでに選択したランドマーク集合に含まれる全ノードから遠いホップ数を持つノードを選択する．

2.6.3 Avoid 選択

Avoid 選択[2]は，各ノードvに対して，重み$s(v)$を計測し，ランドマークを選択する．次の 1-4 の手順を必要なランドマーク数が満たされるまで繰り返す．

1. ランダムにノードrを選び，rをルートとする最短経路木Trを構築する．
2. 最短経路木Trのすべての部分木Tvがランドマークを含んでいれば，vの重みとして，$s(v) = 0$,とする． 含んでいなければ，Tvの子ノード集合Cに含まれるノード各ノードcの$s(c)$の合計を，$s(v)$とする．
3. すべてのノードの$s(v)$を計測した後，ルートノードrから，常に一番$s(v)$が大きい子ノードに移動するように探索を行う．
4. 探索がリーフにたどり着いたら，探索を終了し，リーフノードをランドマークとして選択する．

2.6.4 MaxCover

選択したランドマーク集合を用いたヒューリスティック関数が，各エッジ(v,w) に対して $h(v,w) = d(v,w)$ となるようなエッジの合計数をカバー数と呼び，局所探索を用いてカバー数が大きくなるような組み合わせを，選び出す手法である．候補集合の生成は，Avoid 選択によって行う．

3. 提案手法

本節では，提案手法について述べる．

ALT アルゴリズムでは，ランドマークと他のノードの距離を保存する事前保存領域がランドマーク数増加に対して線形に増加し，多くのランドマークを設定することは，多くの事前保存領域を必要とする問題点がある．そこで本稿では，ALT アルゴリズムにおけるヒューリスティック関数を拡張し，2 ランドマークを用いて推定することにより，ランドマークの追加による事前保存領域が線形には増加しない手法を提案する．

3.1 提案手法におけるヒューリスティック関数

提案手法におけるヒューリスティック関数について示す．

$d(l_f, v)$ として距離を保存したランドマーク集合をL_f ，$d(v, l_r)$として距離を保存したランドマーク集合をL_r とする．

提案手法では，ヒューリスティック関数として

$$\pi(s,t) = d(\mathrm{l_f}, \mathrm{l_r}) - d(\mathrm{l_f}, s) - d(t, \mathrm{l_r})$$

の最大値を用いる．ここで，$\pi(s,t) \leq d(s,t)$ であるので，ヒューリスティック関数として妥当である．

図 3 は，提案手法におけるヒューリスティック関数の概要図である．点線はノードv, wの最短距離の下限値である．l_fとv, wへの実線は，予め計算し保存したランドマークからノードへの実際の最短距離であり，同様にl_rとv, wへの実線はノードからランドマークへの実際の最短距離である．また，l_fとl_r間の実線は，l_fからl_rへの最短距離である．

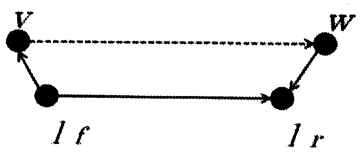

図 3. 提案手法におけるヒューリスティック関数

提案手法では，各ランドマークは，ヒューリスティック関数が最大となるのに必要なだけのノードとの距離を保存する．

3.2. ランドマーク，ノード間の距離保存

ランドマーク，ノード間の距離を保存するための事前処理の手順は次のように行う．

1. ランドマークとなるノードを選択する．
2. ランドマークとなるノードから，他の全ノードへの距離を計算する．
3. 2で求めた結果をもとに，ランドマーク間の距離を保存する．
4. $\pi(s,t) = d(l_f, l_r) - d(l_f, s) - d(t, l_r)$ が最大となるのに必要なだけのノード間の距離だけを保存する．

4 は次のように求める．$l_r \in L_r$ に対し，$d(l_f, l_r) - d(l_f, v)$が最大となるl_fをひとつ求め，$d(l_f, v)$を保存する．$l_f \in L_f$ に対しても同様に保存する．

また，提案手法においては，各ランドマーク間の距離を保存する必要がある．

3.2 提案手法におけるランドマーク選択

本稿では，提案手法におけるランドマーク選択手法として，ランダム選択を用いる．

4. 実験

4.1 実験データ

実験データは，9th DIMACS Implementation Challenge[10]が提供している，全米道路ネットワークデータのうちNYのデータを用いる．ノード数は，264,346，エッジ数は733,846である．

4.2 実験条件

事前にランダムに選択した 100 のs,tペアをテストセットとして生成し，各実験においては同じテストセットを用いて実験を行い，平均探索済みノード数，最大探索済みノード数をそれぞれ求めた．探索時において，探索済みノード数が少なければ，読み込む必要のあるノード数を減少させ，高速に探索を行える．

5. 実験結果

表1に提案手法の実験結果，表2に比較対象としてALTアルゴリズムの実験結果を示す．また，図4に，ALTアルゴリズムの各ランドマーク手法と提案手法のランダム選択の比較のために，事前保存領域に対する平均探索ノード数のグラフを示す．

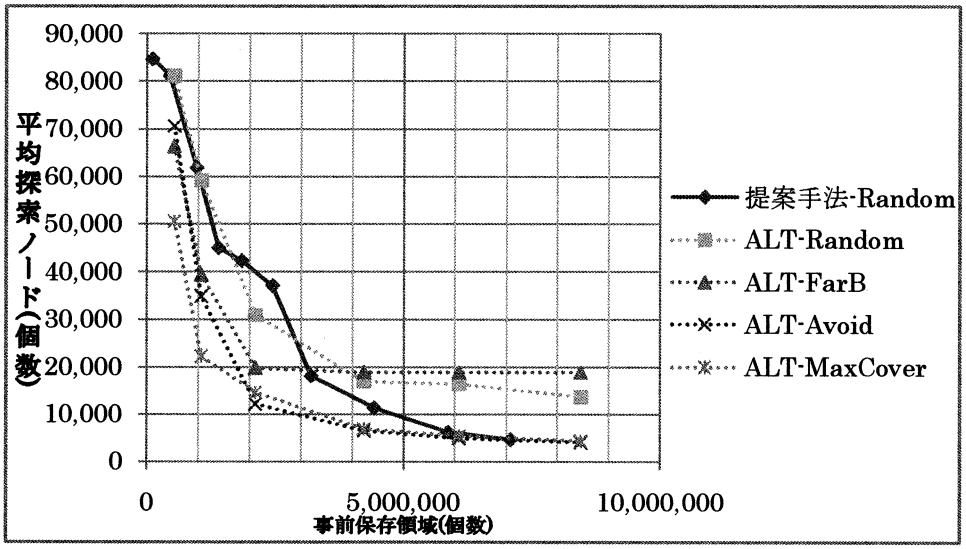

図4. ALTアルゴリズムの各ランドマーク手法の結果と提案手法のランダム選択手法の結果の比較

表 1. 提案手法の実験結果

ランドマーク選択方式	ランドマーク数	事前保存領域へのランドマーク間距離保存数	事前保存領域へのノード距離保存数	事前保存領域合計	平均探索済みノード数	最大探索済みノード数
Random	2	1	117,247	117,248	84,655	181,539
Random	4	4	448,667	448,671	81,098	185,961
Random	8	16	976,575	976,591	61,820	185,963
Random	16	64	1,406,303	1,406,367	45,047	171,685
Random	32	256	1,847,575	1,847,831	42,236	173,809
Random	64	1,024	2,456,268	2,457,292	36,982	168,743
Random	128	4,096	3,201,237	3,205,333	17,985	98,141
Random	256	16,384	4,418,592	4,434,976	11,274	80,143
Random	512	65,536	5,798,568	5,864,104	6,176	81,344
Random	1,024	261,121	6,832,727	7,093,848	4,528	30,461

表 2. ALT アルゴリズムの実験結果

ランドマーク選択方式	ランドマーク数	事前保存領域へのノード距離保存数	平均探索済みノード数	最大探索済みノード数
Random	2	528,692	81,085	191,708
Random	4	1,057,384	59,111	196,050
Random	8	2,114,768	30,784	118,373
Random	16	4,229,536	16,837	62,896
Random	23	6,079,958	16,331	62,838
Random	32	8,459,072	13,654	64,174
Farb	2	528,692	66,274	186,984
Farb	4	1,057,384	39,300	157,966
Farb	8	2,114,768	19,712	112,011
Farb	16	4,229,536	18,881	111,468
Farb	23	6,079,958	18,876	111,468
Farb	32	8,459,072	18,847	111,468
Avoid	2	528,692	70,615	212,495
Avoid	4	1,057,384	34,833	156,074
Avoid	8	2,114,768	12,157	55,034
Avoid	16	4,229,536	6,439	43,440
Avoid	23	6,079,958	4,821	29,851
Avoid	32	8,459,072	4,061	29,929
MaxCover	2	528,692	50,563	165,814
MaxCover	4	1,057,384	22,280	90,715
MaxCover	8	2,114,768	14,541	67,792
MaxCover	16	4,229,536	6,904	43,956
MaxCover	23	6,079,958	5,369	43,784
MaxCover	32	8,459,072	4,273	29,551

5.1 ランドマークをランダム選択したALTアルゴリズムとの比較

ランダム選択を用いた提案手法を，ランダム選択を用いたALTアルゴリズムと比較すると，提案手法で128ランドマーク以上を用いた結果においては，ALTアルゴリズムよりも小さい事前保存領域で，平均して少ない探索空間で探索可能になっている．すなわち，ランダム選択においては，提案手法はALTアルゴリズムに対して，保存領域に対して効率の良い性能を得ることができる．

5.2 ランドマークを他の手法において選択したALTアルゴリズムとの比較

FarB選択を用いたALTアルゴリズムとの比較においては，ランダム選択と同様の結果が得られた．

より性能の良いランドマークを選択する手法であるAoid選択，MaxCover選択を用いたALTアルゴリズムとの比較においては，128ランドマーク以上においてもALTアルゴリズムが保存領域に対する効率が良いものの，512ランドマーク以上を用いた結果においては，ほぼ同等の結果となった．

6. まとめと今後の課題

本稿では，ALTアルゴリズムを基に事前保存領域を抑える試みとして，ヒューリスティック関数を2ランドマークを用いて推定を行うように拡張し，ランドマークの追加による事前保存領域が線形には増加しない手法を提案した．道路ネットワークを用いた実験の結果，提案手法においてランドマークをランダムに選択した場合，ALTアルゴリズムにおいてランドマークをランダムに選択した場合よりも，少ない事前保存領域で，平均して少ない探索空間で最短経路を得るができることを確認した．

より性能の良いランドマークを選択する手法であるAoid選択，MaxCover選択を用いたALTアルゴリズムとの比較実験の結果から，今後は，提案手法においても，ランダム選択よりも性能を得ることができるランドマーク選択手法を開発することが必要であり，課題としたい．

謝辞

本研究の一部は，科学研究費補助金「情報爆発に対応する高度にスケーラブルなモニタリングアーキテクチャ」によるものである．

参考文献

[1] A. V. Goldberg and C. Harrelson, "Computing the Shortest Path: A* Search Meets Graph Theory," In Proc. 16th ACM-SIAM Symposium on Discrete Algorithms, pp.156-165, 2005.

[2] A. V. Goldberg and R. F. Werneck, "Computing Point-to-Point Shortest Paths from External Memory," In Proc. 7th International Workshop on Algorithm Engineering and Experiments, pp. 26-40, SIAM, 2005.

[3] P. E. Hart, N. J. Nilsson, and B. Raphael , "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on System Science and Cybernetics, SSC-4(2), pp. 100-107,1968.

[4] E. W. Dijkstra,, "A Note on Two Problems in Connexion with Graphs," Numerische Mathematik, vol.1, pp.269-271 , 1959

[5] G. B. Dantzig, " Linear Programming and Extensions. Princeton Univ. Press," Princeton, NJ, 1962.

[6] T. A. J. Nicholson, " Finding the Shortest Route Between Two Points in a Network," Computer J., vol.9, pp.275-280, 1966.

[7] D. Dreyfus, " An Appraisal of Some Shortest Path Algorithms," Technical Report RM-5433, Rand Corporation, Santa Monica, CA, 1967.

[8] I. Pohl, " Bi-directional Search. In Machine," Intelligence, vol..6, pp.124-140. Edinburgh Univ. Press, Edinburgh, 1971.

[9] T. Ikeda, Min-Yao Hsu, H. Imai, S. Nishimura, H. Shimoura,T. Hashimoto, K. Tenmoku, and K. Mitoh. , "A Fast Algorithm for Finding Better Routes by AI Search Techniques," In Proc. Vehicle Navigation and Information Systems Conference. IEEE, 1994.

[10] "9th DIMACS Implementation Challenge - Shortest Paths,"
http://www.dis.uniroma1.it/~challenge9/index.shtml