

経路途中の変更可能性を考慮した確率的 shortest 経路案内アルゴリズムの開発*

Development of probabilistic shortest path algorithm considering on-trip rerouting*

安部遼祐**・山本俊行***

By Ryosuke ABE**・Toshiyuki YAMAMOTO***

1. はじめに

既存の渋滞情報提供を大きく超え、各ドライバーの目的地とリアルタイムの交通状況に基づいた、「動的経路誘導システム」が今後の道路交通マネジメントの鍵となる。しかしながら、現状で動的経路誘導に用いられている最短経路探索アルゴリズムは必ずしも入手可能な交通情報を十分に生かしているとは限らない。これは一つには動的経路誘導を実施するメディアの有力なデバイスであるカーナビゲーションシステムの計算能力が入手可能な情報量を十分に活かし切れていないことによる。

所要時間が不確実性を持つ場合、最適な経路は1つではなく、最適となる可能性を持つ複数の経路が存在することとなる。出発時には、これらの複数の経路を最適経路集合ととらえ、経路途中で追加的に得られる交通情報に基づいてそれらの経路集合から最適な経路を逐次的に選択することが可能である。これまでに提案されている複数経路の探索アルゴリズムはn番目最短経路探索アルゴリズムと複数独立経路探索アルゴリズムに大別される。しかしながら、これらのアルゴリズムはOD間の最短経路探索を複数回繰り返す必要があるなど、計算負荷が高くなり、応答時間が重要な要素であるカーナビゲーションシステムへの適用は困難である。

近年、複数の最適経路集合を1回の探索で実行するアルゴリズムがBell¹⁾により提案されている。このアルゴリズムは計算負荷が高くなく、カーナビゲーションシステムへの適用も可能である。しかしながら、本アルゴリズムは鉄道駅での待ち時間の不確実性を考慮したSpiess and Florianアルゴリズム²⁾を適用したものであり、駅に到着した段階では、最適な経路集合に含まれる列車のうち、最初に来た列車を選択することを意味している。

これは、交差点等の分岐点でそれぞれの進路に対して情報が得られると考えられる道路網上での遅れ時間とは若干性質が異なる。そのため、探索された経路は真に最適な経路集合とは限らない。

本研究では、Bellでの暗黙の仮定を修正し、分岐ノー

ドに到着した時点でそのノードから出発する全てのリンクに関して情報が得られ、それら各リンクの遅れ時間は確定するという仮定を置き、出発時点でどの経路集合を最適経路集合として案内するのが最も期待所要時間が短くなるか、という観点から経路探索を行うアルゴリズムを構築する。

2. アルゴリズムの概要

本研究では、Bell¹⁾によるHyperstarアルゴリズムをより道路ネットワーク上での仮定に対応した形へと拡張することにより、複数のアルゴリズムを構築する。

(1) 最大所要時間を考慮したHyperstarアルゴリズム

Hyperstarアルゴリズムでは最大遅れ時間のみを用いて、リンク選択確率を決定していた。ここでは、リンク所要時間も考慮できるようにHyperstarアルゴリズムの拡張を行う。

図1のIOD - 2リンクの例を用いてアルゴリズムの方針を示す。図中のかっこは[リンク所要時間 c_i , リンクの最大遅れ時間 d_i]を表している。まず、Hyperstarでの結果は、リンクの選択確率はそれぞれのリンクの最大遅れ時間を用いて、リンクcの選択確率は $p_c = (1/4)/(1/4 + 1/2) = 1/3$, リンクdの選択確率は $p_d = (1/2)/(1/4 + 1/2) = 2/3$, 期待所要時間は $u_0 = p_c \times 5 + p_d \times 8 + 1/(1/4 + 1/2) = 8.3$ となる。リンク選択確率は最大遅れを最小にするという観点からは妥当であるが、リンク所要時間の長い経路に多くの選択確率を与えている、という点では不自然な選択確率とも言える。

ここで構築するアルゴリズムは、リンク選択確率を決める際に、リンク所要時間と最大遅れの両者を含めた合計の所要時間を用いる。これにより、 $p_c = (8 + 2)/(8 + 2 + 5 + 4) = 10/19$, $p_d = (5 + 4)/(8 + 2 + 5 + 4) = 9/19$ となる。また、 $u_0 = p_c \times 5 + p_d \times 8 + 10 \times 4/19 = 8.52$ となる。

*キーワード：経路選択，交通情報，ITS

**学生員，東京大学大学院工学系研究科

***正員，博(工)，名古屋大学大学院工学研究科

(名古屋市千種区不老町，Tel: 052-789-4636, Fax: 052-789-5728)

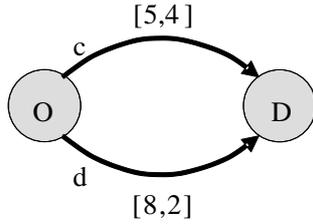


図1 IOD-2 リンクの例

このアルゴリズムは以下のようになる。

```

0. Initialisation
  ui ← ∞, i ∈ I - {s}, us ← 0
  f'a ← 1/(ca + da)
  f'i ← 0, i ∈ I
  di ← 0, i ∈ I
  yi ← 0, i ∈ I - {r}, yr ← 1
  L ← A
  H ← ∅
  Hi ← ∅

1. Select link a
  Find a = (i, j) ∈ L with minimum hi + uj + ca
  L ← L - {a}

2. Update node i
  If ui ≥ uj + ca then
    Hi ← Hi + {a}

    di ← maxa'=(i,j') ∈ Hi  $\frac{d_{a'} f'_{a'}}{f'_i + f'_{a'}}$ 

    ui ←  $\sum_{a'=(i,j') ∈ Hi  $\left\{ (u_{j'} + c_{a'}) \frac{f'_{a'}}{f'_i + f'_{a'}} \right\} + d_i$ 

    f'i ← f'i + f'a
    H ← H + {a}
  if L = ∅ or hi + uj + ca > ur go to Step3 else go to Step1

3. Loading
  For every link a ∈ A in decreasing order of hi + uj + ca
  if a ∈ H then pa ← (f'a/f'i)yi and yi ← yi + pa else pa ← 0$ 
```

文字の定義等はHyperstarと同じである。アルゴリズム1はHyperstarアルゴリズムの一部を変更したものである。Hyperstarでの $f_a = d_a$ をアルゴリズム1では $f'_a = 1/(c_a + d_a)$ としている。また、ノードiから流出するリンク $a' = (i, j')$ のうち、最適な経路集合(ハイパーパス)に含まれるリンクの数を H_i と定義する。

(2) リンク所要時間と遅れ時間の発生確率を考慮したアルゴリズム

Hyperstar, アルゴリズム1では、リンク選択確率を決める際に、常にリンクの最大遅れ時間 (+リンク所要時間) を用いていた。ここで構築するアルゴリズムは、

終点までの「リンク所要時間+最大遅れ時間」をリンク選択確率の決定に用いるようにする。またこの際に、遅れ時間に確率分布を与え、より詳細な情報を決定に組み込めるようにする。

具体的な方針をリンクが並列となる時(図1)から示す。ここでは、確率分布として、遅れが発生しない(遅れゼロ) 確率を 1/2, 最大遅れが発生する確率を 1/2 と設定する。すなわち、リンク c ではリンク所要時間が 5 となる確率を 1/2, 9 となる確率を 1/2 とする。リンク d も同様である。このとき、リンク c, d において各 2 通りの事象があるため、ノード O では表 1 のようにリンク c, d の組み合わせによって 4 通りの事象を認識することになる。各事象の発生確率は独立とすると、リンク c, d それぞれの発生確率をかけ、各事象 1/4 となる。また、4 通りの各組み合わせで実際に利用されるのは所要時間が短いリンクであることから、各事象で利用するリンクが決まる。これを元にノード O での各リンクの選択確率は $p_c = 1/4 + 1/4 + 1/4 = 3/4$, $p_d = 1/4$ となる。また、ノード O の期待所要時間は $u_o = 5 \times 1/4 + 9 \times 1/4 + 8 \times 1/4 + 10 \times 1/4 = 6.75$ となる。

表1 並列の場合

リンクc	リンクd	発生確率	利用するリンク
5	8	1/4	c
5	10	1/4	c
9	8	1/4	d
9	10	1/4	c

次に、図 2 のようにリンクが直列の場合、ノード O では表 2 のように 4 通りの事象を認識することになる。この場合、終点ノード D までの所要時間 13, 15, 17, 19 の各事象が各 1/4 の確率で起こる。そして、 $p_c = 1/4 \times 4 = 1$, $p_d = 1/4 \times 4 = 1$, $u_o = 3 \times 1/4 + 15 \times 1/4 + 17 \times 1/4 + 19 \times 1/4 = 16$ となる。

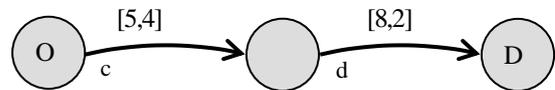


図2 直列の場合

表2 直列の場合

リンクc	リンクd	合計所要時間	発生確率	利用するリンク
5	8	13	1/4	c,d
5	10	15	1/4	c,d
9	8	17	1/4	c,d
9	10	19	1/4	c,d

以上の方針に基づいてアルゴリズムを構築する。具体

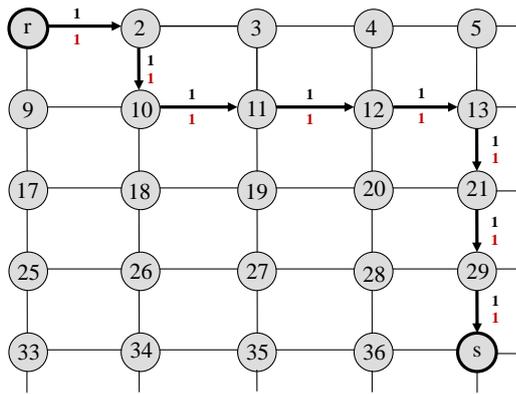


図6 $d_a=0$ (Hyperstar とアルゴリズム 1)

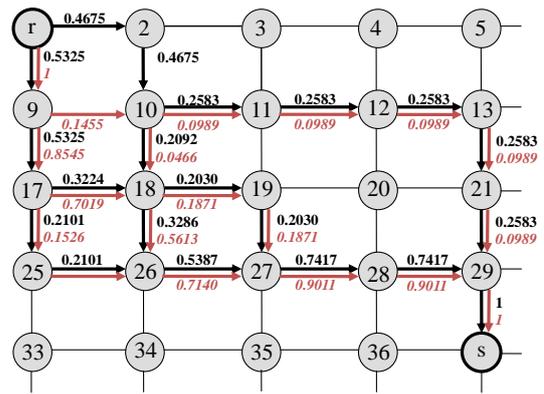


図8 $d_a=R$, (遅れゼロ, 最大遅れ)=(0.5,0.5)
(アルゴリズム 2 と 3)

表3 期待所要時間と選択リンク数
(Hyperstar とアルゴリズム 1)

最大遅れ時間	Hyperstar		アルゴリズム1	
	u_r	選択リンク数	u_r	選択リンク数
$da=R$	13.6226	98	14.4334	109
$da=0.5R$	12.4169	81	12.8898	94
$da=0.3R$	11.8649	67	11.9583	69
$da=0$	10.6994	46	10.6993	46

表3での選択リンク数とはアルゴリズムの Step1 で選択されたリンク数を示す。選択リンク数はアルゴリズムの効率性を表す指標として意味がある。Hyperstar の選択リンク数に関しては Bell (2009)の原文と一致していないが、期待所要時間、リンク選択確率は全て原文と誤差の範囲内で一致しているため、作成した Hyperstar に大きな誤りはないと思われる。

(3) アルゴリズム2, 3の結果

アルゴリズム 2, 3は全て $d_a = R$ とし、リンクの発生確率は、(遅れゼロ, 最大遅れ) = (0,1), (0.5,0.5), (0.3,0.7), (1,0)の場合で行った。リンク選択確率は上に書かれているものがアルゴリズム 2, 下がアルゴリズム 3である。

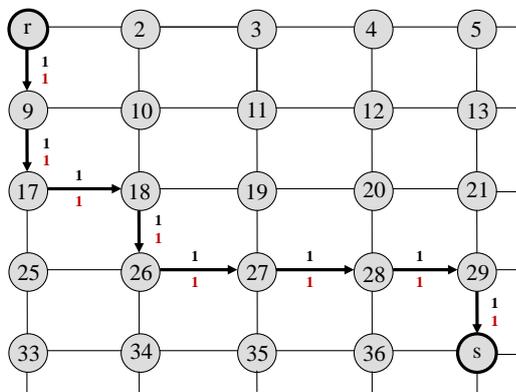


図7 $d_a=R$, (遅れゼロ, 最大遅れ)=(0,1)
(アルゴリズム 2 と 3)

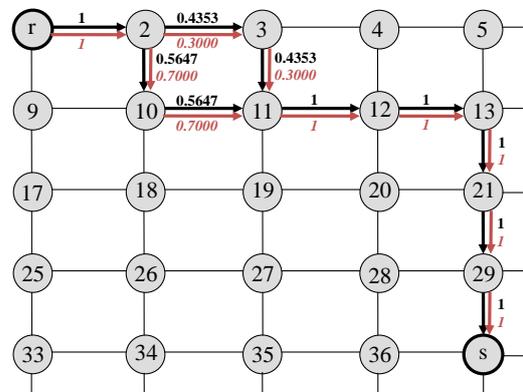


図9 $d_a=R$, (遅れゼロ, 最大遅れ)=(0.7,0.3)
(アルゴリズム 2 と 3)

$d_a = R$, (遅れゼロ, 最大遅れ) = (1,0)の場合の経路は図6と同じである。

表4 期待所要時間と選択リンク数
(アルゴリズム 2 と 3)

$da=R$	アルゴリズム2		アルゴリズム3	
(z_{a1}, z_{a2})	u_r	選択リンク数	u_r	選択リンク数
(0,1)	14.2574	109	14.2574	109
(0.5,0.5)	12.0366	72	11.5118	50
(0.7,0.3)	11.5947	60	11.2140	46
(1,0)	10.6993	46	10.6993	46

アルゴリズム 2 の(0.5,0.5)の経路集合は Hyperstar, アルゴリズム 1 の $d_a = 0.5R$ の場合に $2 \rightarrow 3 \rightarrow 11$ を通過する経路を除いて一致し、(0.7,0.3)の場合は $d_a = 0.3R$ と一致する。また、これらアルゴリズム 2 のリンク選択確率は Hyperstar よりもアルゴリズム 1 の方に近い。したがって、アルゴリズム 1 とアルゴリズム 2 は最適な経路集合、リンク選択確率に関してはアルゴリズム 1 での $d_a = xR$ の係数 x はアルゴリズム 2 での発生確率 (z_{a1}, z_{a2}) が全てのリンクで同じとき、 $z_{a1} = 1-x$, $z_{a2} = x$ とある程度対応する。(ただし(0,1), (1,0)の場合は除く。)しかし、両者の期

待所要時間に関してはあまり一致していない。

また、アルゴリズム 2 と 3 を比較したとき、経路、期待所要時間、選択リンク数に全てにおいてアルゴリズム 3 はアルゴリズム 2 の過小評価となっている。また、両者のリンク選択確率の差は大きい。アルゴリズム 3 はアルゴリズム 2 の良い近似解を与える簡略法とはなっていない。

本研究では全てのリンクの発生確率を(0.5,0.5)のように一律に設定しているが、アルゴリズム 2, 3 では実際のリンクの情報によって発生確率を各リンク自由に決めることができる。

まとめると、アルゴリズム 1 は Hyperstar に比べより道路ネットワークでの条件が満たされたアルゴリズムとなっている。また、Hyperstar 同様に計算負荷が低く実用的である。アルゴリズム 2 はアルゴリズム 1 に比べより詳細な交通情報も考慮可能なアルゴリズムであるが、計算負荷が高く実用的ではない。アルゴリズム 2 を簡略化したアルゴリズム 3 は計算負荷が低く、一応実用的であると言える。現時点ではカーナビでの利用を想定した複数経路の最短経路探索アルゴリズムとしてアルゴリズム 1, 3 が考えられる。

4. おわりに

実際には、アルゴリズム 2, 3 の方針を元に本研究で作成したアルゴリズムでは最適な全てのリンクを最適な経路集合に加えて切れていない可能性がある。したがって、期待所要時間が短縮される可能性がある。現状では、期待所要時間を短縮させることが可能なリンクを最適な経路集合に加えるという条件を完全には満たしていないが、アルゴリズムとして実用可能なものとしてアルゴリズム 2, 3 を作成した。この条件を完全に満たすようにアルゴリズム 2, 3 の作成も行ったが、このときはアルゴリズム 3 の簡略法はアルゴリズム 2 の良い近似となった。しかし、これらのアルゴリズムは現時点ではアルゴリズムとして実現可能な形ではない。

今後は、この条件に対応した形への本研究のアルゴリズムの改良を行うことが必要となる。

参考文献

- 1) Bell, M.G.H.: Hyperstar: A multi-path Astar algorithm for risk averse vehicle navigation, Transportation Research Part B, Vol. 43, pp. 97-107, 2009.
- 2) Spiess, H., Florian, M.: Optimal strategies: a new assignment model for transit networks, Transportation Research Part B, Vol.23, No. 6, pp.83-102, 1989