情報処理学会論文誌　Vol.54　No.4　1632–1640 (Apr. 2013)

経路の特徴を反映した評価値の更新をともなう木探索

芦田 昌也[1,a)]　瀧 寛和[2]

受付日 2012年3月29日, 採録日 2013年1月11日

概要：根から葉節点に向かい，順次大きいコストを持つ辺で作られた木構造における探索手法を提案する．提案手法は，A*アルゴリズムをベースとして，直近に観測したコストの真値を利用し，所与のコストの推定値を更新しながら探索を進めるものである．コストの推定値は，その真値を超えないように更新されるため，対象とする問題空間において，提案手法は最適解を発見することが保証される．また，コストの推定値を更新することにより，その真値に近づけることができる．そのため，解を発見するまでに提案手法が展開する節点数は，同一の問題空間に適用したA*アルゴリズムによって展開される節点数以下であることが保証される．さらに，完全二分木を対象としたシミュレーションの結果からは，推定コストの初期値の精度にかかわらず，問題空間のいずれの深さにおいても，提案手法による有効分岐因子の値は，A*アルゴリズムより小さいことが示される．提案手法は，A*アルゴリズムの特性を継承しつつ，コストの分布に特徴がある問題空間において有効な手法であると考えられる．

キーワード：木探索，発見的探索，A*アルゴリズム，有効分岐因子，コストの偏り

A Tree Search Method with Improvement of Search Cost Estimation Based on Characteristics of Cost Allocation

Masaya Ashida[1,a)]　Hirokazu Taki[2]

Received: March 29, 2012, Accepted: January 11, 2013

Abstract: This article proposes a search method applied to a tree structure that each link cost is larger than the cost of its precedent link along the path from root node to each leaf node. The proposed method searches the tree structure, updating the estimation of link cost based on the actual value of its precedent link cost. It is proved that the proposed method is able to find an optimal solution. And it is also proved that the number of expanded nodes by the proposed method in finding the solution is less than or equal to the number of expanded nodes by A* algorithm. Results of simulation using the proposed method on complete binary trees show that EBF (effective branching factor) by the proposed method is smaller than that by A* algorithm on each depth regardless of the accuracy of the initial value of estimated cost. The proposed method inherits some of known properties of A* algorithm and is effective to the tree structure which is assigned the characteristic cost.

Keywords: tree search, heuristic search, A* algorithm, effective branching factor, bias of cost

1. はじめに

発見的探索法は，対象となる問題空間に対する経験的知識を反映した評価関数を利用して，探索効率を向上させる手法である [1]．経路探索問題では，目的地までの距離や所要時間などをコストと見なし，最短経路としてコスト最小の経路を求めることが行われる．探索開始前には実際のコ

1　和歌山大学経済学部
Faculty of Economics, Wakayama University, Wakayama 640–8510, Japan
2　和歌山大学システム工学部
Faculty of Systems Engineering, Wakayama University, Wakayama 640–8510, Japan
a)　ashida@eco.wakayama-u.ac.jp

スト（以下，実コスト）は未知であるから，探索はその推定値を利用して進められる．代表的な手法である A*アルゴリズムでは，推定値が実コスト以下であり，より実コストに近いほど探索効率の向上が期待できる [2]．

問題空間の性質によっては，実コストの分布に偏りが生じる場合がある．たとえば，道路網では交通量が増加すると，移動距離あたりの所要時間は長くなり [3]，結果として渋滞が発生する．また，鉄道網では乗客の乗降に想定以上の時間を要していったん遅延が発生すると，その影響は後続列車におよび，次第に拡大する [4]．いったん渋滞や遅延が発生すると，その影響により新たな渋滞や遅延の起点が生じ，周辺地域や関連路線に拡大することもある [5]．このとき，渋滞や遅延の起点に近い区間ほど遅延時間が長い状況に陥っている．この状態は，遅延時間を区間に付与されるコストととらえると，その渋滞や遅延の起点に近づくほど，実コストが大きくなるという偏りのある問題空間と見なせる．渋滞や遅延の程度は状況によって異なるため，所与のコストが必ずしも有効ではない場合も生じる [6]．

本研究は，このような性質のある問題空間においては，実コストの偏りの程度を反映するように推定値を更新しながら探索すると，探索効率が高められる可能性があることを示すものである．この問題空間に適した探索手法として，A*アルゴリズムに推定値の更新方法を埋め込んだ探索手法を提案する．

以下では，まず，対象とする問題空間におけるコストの偏りに対する前提と，推定値の更新方法を示す．次に，探索手法を示し，対象とする問題空間での探索時に現れる特性について述べる．そこでは，探索手法が最適解を発見することと，解を発見するまでの探索範囲が推定値を更新しない場合と同じであるか，または狭くなることを理論的に保証する．最後に，シミュレーションを用いて，初期状態において精度の高いコストの推定値が与えられた場合でも，A*アルゴリズムより狭い探索範囲で解が発見されることを実験的に示す．

2.　関連研究

発見的探索法は，木構造やグラフ構造で表現される状態空間において，求解に有効な発見的知識をコストの評価関数に反映させ，効率的に解を発見しようとする手法である．代表的な手法の 1 つに A*アルゴリズムがある．対象となる状態空間が木構造として表されるとき，現在の状態 x から目標状態に至る経路の推定コスト $h(x)$ が真のコスト $h^*(x)$ を超えない範囲で精度が高いほど効率良く最適解を発見することができる．対象となる状態空間には，通常，推定コストが事前に割り当てられるが，真のコストと大きく異なることがある．このような場合に，推定コストを更新しながら探索する手法として，実時間探索の代表的な手法である RTA*や LRTA* [7] がある．LRTA*は現在の状態 x から隣接する状態の 1 つである y への状態遷移で観測された推定コスト $h(y)$ を用いて，もとの状態 x の推定コスト $h(x)$ を更新する．LRTA*(k) [8] や LRTA*$_{LS}$(k) [9] は，現在の状態 x から次に遷移する状態 y の決定に先立ち，$h(x)$ とそれまでの経路に現れている状態 p_i における推定コスト $h(p_i)$ を更新する．LRTA*と同様に x や p_i の隣接状態の 1 つへ移動し，そこで観測された推定コストを利用して $h(x)$ や $h(p_i)$ を更新する．

LRTA*や LRTA*(k) のような実時間探索では，推定コストの更新をともなう複数回の同一状態への遷移を繰り返す．これにより，各状態の推定コストが真のコストに収束し最適解に到達する．一方，A*アルゴリズムでは，推定コストが適格性を満たす限り最適解に到達する．ただし，推定コストの精度が低い場合には探索範囲が広くなる．扱う問題の性質によっては，推定コストが真のコストに収束するまでに要する時間や，広範囲を探索するために要する時間を許容できない場合がある．そのため，同一状態への遷移を回避しつつ推定コストの精度を高めながら，解を発見する手法が望まれる．

3.　提案手法

3.1　問題空間

問題空間は**図 1** に示すような木構造とする．ある節点 v の子節点の 1 つが w であるとき，この間の辺を (v, w) で表す．特に，初期節点を S，葉節点を G_i（$i = 1, \ldots, n$）と表記する．また，v を根と見なした部分木を $T_s(v)$ で表す．辺 $(v_i, v_{i+1}), (v_{i+1}, v_{i+2}), \cdots$ を経由して，v_i から到達可能な節点 v_j へ至る経路は，$(v_i, v_{i+1}, \cdots, v_j)$ で表す．節点間の各辺には正の値を持つコストが付与される．辺 (v, w) の真のコストを $c^*(v, w)$ とし，その推定値（以後，推定コストと記す）を $c(v, w)$ で表す．また，推定コストの所与の値（以後，初期値と記す）を $c_0(v, w)$ とする．対象とする木構造について，次の前提と仮定をおく．

前提 1　初期節点から葉節点に向かう経路に沿って，辺の真のコストは非減少である．すなわち，節点 u とその子節点 v，さらに v の子節点 w からなる経路 (u, v, w) の 2 つの構成辺 (u, v)，(v, w) の真のコストについて

$$c^*(u, v) \leq c^*(v, w) \tag{1}$$

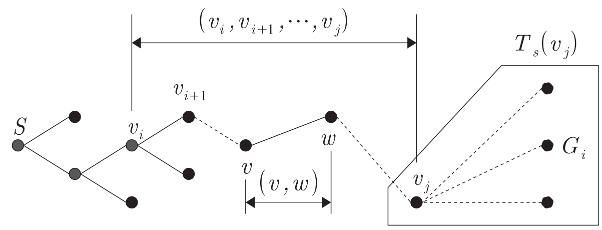

図 1　問題空間の表現
Fig. 1　Notation of a search tree.

である．

仮定 1　初期状態では，各辺の推定コストの初期値が既知であり，それは真のコストを超えないものとする．すなわち，辺 (v, w) について

$$c_0(v, w) \leq c^*(v, w) \tag{2}$$

である．

3.2　評価関数

問題空間において探索を進めるうえで，次に展開する節点を選択するための評価関数を定める．

定義 1　経路の真のコスト
経路 $(v_i, v_{i+1}, \cdots, v_j)$ の真のコスト $p^*(v_i, v_j)$ は，その経路を構成する各辺の真のコストの和である．

$$p^*(v_i, v_j) = \sum_{k=i}^{j-1} c^*(v_k, v_{k+1}) \tag{3}$$

特に，初期節点 S から v_i に至る経路 $(S, v_1, \cdots, v_i)$ の真のコスト $p^*(S, v_i)$ を $g^*(v_i)$ と表記する．すなわち，$g^*(v_i) = p^*(S, v_i)$ である．

定義 2　経路の推定コスト
経路 $(v_i, v_{i+1}, \cdots, v_j)$ の推定コスト $p(v_i, v_j)$ は，その経路を構成する各辺の推定コストの和である．

$$p(v_i, v_j) = \sum_{k=i}^{j-1} c(v_k, v_{k+1}) \tag{4}$$

特に，定義 2 において推定コスト $c(v_k, v_{k+1})$ をその初期値 $c_0(v_k, v_{k+1})$ で置き換えて得られる値を経路の推定コストの初期値 $p_0(v_i, v_j)$ とする．

定義 3　目標コスト
節点 v から葉節点への目標コスト $h^*(v)$ は，v から到達可能な葉節点 $\{G_1, \cdots, G_m\}$ へ至る各経路の真のコスト $p^*(v, G_1), \cdots, p^*(v, G_m)$ の最小値である．

$$h^*(v) = \min_{k=1}^{m} \{p^*(v, G_k)\} \tag{5}$$

葉節点 $\{G_1, \cdots, G_m\}$ の中に最適解である目標節点 G_{opt} が含まれるとき，$h^*(v)$ は最適解の目標コストである．

定義 4　推定目標コスト
節点 v から葉節点への推定目標コスト $h(v)$ は，v から到達可能な葉節点 $\{G_1, \cdots, G_m\}$ へ至る各経路の推定コスト $p(v, G_1), \cdots, p(v, G_m)$ の最小値である．

$$h(v) = \min_{k=1}^{m} \{p(v, G_k)\} \tag{6}$$

特に，定義 4 において $p(v, G_k)$ をその初期値 $p_0(v, G_k)$ に置き換えて得られる値を推定目標コストの初期値 $h_0(v)$ とする．

定義 5　評価関数
初期節点 S から到達した途中の節点 u とその子節点の 1 つ v を経由し，v から到達可能な葉節点の 1 つ G_j に至る経路 $(S, \cdots, u, v, \cdots, G_j)$ の評価関数を $f(u)$ とする．$f(u)$ は，S から u に至る経路の真のコスト $g^*(u)$，辺 (u, v) の推定コスト $c(u, v)$，v からの推定目標コスト $h(v)$ の 3 項の和である．

$$f(u) = g^*(u) + c(u, v) + h(v) \tag{7}$$

図 2 は u が展開された時点の問題空間の様子と u の子節点の 1 つ v に着目した場合の評価関数 $f(u)$ の各項との対応を表した模式図である．展開した節点を黒丸で記し，生成した節点は白丸で記す．破線で記した節点は未生成の節点である．実線で表した辺は，その辺を含む経路を評価するときに，真のコストで評価される部分であり，破線で表した辺は，推定コストで評価される部分である．v を含む矩形領域は部分木 $T_s(v)$ である．

3.3　推定コストの更新

節点 u を展開したときに現れる子節点を $\{v_1, \cdots, v_n\}$ とする．各子節点を経由する経路を評価する時点では，辺 (u, v_i) 間のコストは推定コストで評価する．評価関数 $f(u)$ の最小値を与える節点 $v \in \{v_1, \cdots, v_n\}$ を選択して展開すると，辺 (u, v) の真のコスト $c^*(u, v)$ が観測される．

図 3 に示すような部分木 $T_s(v)$ 内の任意の経路 $(v, w, \cdots, u', v', w', \cdots)$ について，前提 1 に基づき，構成辺の真のコストは経路に沿って非減少である．したがって，辺 (u, v) と部分木 $T_s(v)$ 内の任意の辺 (u', v') の真のコストについて，推移的に次式が成立する．

$$c^*(u, v) \leq c^*(u', v') \tag{8}$$

これは，部分木 $T_s(v)$ 内のすべての辺の推定コスト $c(u', v')$ のうち，$c(u', v') < c^*(u, v) \leq c^*(u', v')$ であるものについては，その推定コストを真値に近づけるために，$c(u', v')$ を $c^*(u, v)$ に置き換える方がよいことを示唆している．そ

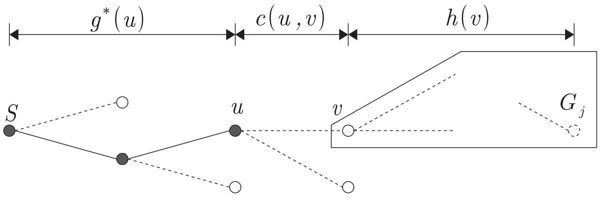

図 2　評価関数の構成
Fig. 2　Each element of an evaluation function.

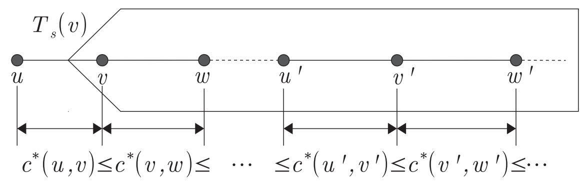

図 3　経路に沿った各辺の真のコストの関係
Fig. 3　Relation of actual cost of each link along a path.

こで，部分木内の各辺の推定コストを次の方法で更新する．k 回目の更新値を $c_k(u',v')$ で表す．

定義 6　辺の推定コストの更新方法

$$c_{k+1}(u',v') = \max\{c^*(u,v), c_k(u',v')\} \tag{9}$$

この更新方法には，次の特徴がある．

補題 1　$k = 0,1,\ldots$ において $c_k(u',v') \le c_{k+1}(u',v') \le c^*(u',v')$，すなわち，辺の推定コストは更新により減少することはなく，かつ，その真のコストを超えることもない．

証明 1（補題 1）　まず，更新後の値 $c_{k+1}(u',v')$ は，定義 6 により，観測済みの値 $c^*(u,v)$ と更新前の値 $c_k(u',v')$ のいずれか大きい方の値になるため，$k = 0,1,\ldots$ において $c_k(u',v') \le c_{k+1}(u',v')$ は自明である．
次に，初期値 $c_0(u',v')$ は，仮定 1 により，$c_0(u',v') \le c^*(u',v')$ であり，かつ，そのように初期値を設定できるから，$k=0$ において次式が成立する．

$$c_0(u',v') \le c^*(u',v') \tag{10}$$

k 回目の更新値 $c_k(u',v')$ が真のコスト $c^*(u',v')$ を超えないと仮定すると，次式が成立しなければならない．

$$c_k(u',v') \le c^*(u',v') \tag{11}$$

式 (9) による $k+1$ 回目の更新値について，式 (8) と式 (11) から次式が成立する．

$$c_{k+1}(u',v') = \max\{c^*(u,v), c_k(u',v')\} \le c^*(u',v') \tag{12}$$

式 (12) は，$c_k(u',v') \le c^*(u',v')$ を仮定した場合に，$c_{k+1}(u',v') \le c^*(u',v')$ が成立することを示している．すなわち，$k = 0,1,\ldots$ において $c_k(u',v') \le c_{k+1}(u',v') \le c^*(u',v')$ である．　□

以後，推定コストを表記する際に，特に必要がない限り更新回数の表記と初期値の区別を省略する．

補題 1 から次の補題が示される．

補題 2　経路 $(v_i, v_{i+1}, \cdots, v_j)$ の推定コスト $p(v_i,v_j)$，その初期値 $p_0(v_i,v_j)$，および真のコスト $p^*(v_i,v_j)$ について，$p_0(v_i,v_j) \le p(v_i,v_j) \le p^*(v_i,v_j)$，すなわち経路の推定コストの値はその初期値以上であり，かつ真のコストの値を超えない．

証明 2（補題 2）　経路 $(v_i, v_{i+1}, \cdots, v_j)$ の各構成辺の推定コスト $c(v_k, v_{k+1})$ は，補題 1 から，各辺について $c_0(v_k,v_{k+1}) \le c(v_k,v_{k+1}) \le c^*(v_k,v_{k+1})$ である．経路に沿って $c(v_i,v_{i+1})$ から $c(v_{j-1},v_j)$ まで辺々加え次式を得る．

$$\sum_{k=i}^{j-1} c_0(v_k,v_{k+1}) \le \sum_{k=i}^{j-1} c(v_k,v_{k+1}) \le \sum_{k=i}^{j-1} c^*(v_k,v_{k+1}) \tag{13}$$

定義 1 および定義 2 から，$p_0(v_i,v_j) \le p(v_i,v_j) \le p^*(v_i,v_j)$ である．　□

補題 3　推定目標コスト $h(v)$，その初期値 $h_0(v)$，および目標コスト $h^*(v)$ について，$h_0(v) \le h(v) \le h^*(v)$，すなわち推定目標コストの値はその初期値以上であり，かつ目標コストの値を超えない．

証明 3（補題 3）　節点 v から到達可能な葉節点 G_k（$k = 1,\ldots,m$）への経路 $(v,\cdots,G_k)$ について，補題 2 により $p_0(v,G_k) \le p(v,G_k) \le p^*(v,G_k)$ である．定義 3 から $h^*(v)$ を与える葉節点が G_{min} であるとすると，$p(v,G_{min}) \le p^*(v,G_{min}) = h^*(v)$ である．定義 4 により $h(v) = \min_{k=1}^{m}\{p(v,G_k)\} \le p(v,G_{min})$ であるから，次式が成立する．

$$h(v) \le h^*(v) \tag{14}$$

また，$h(v)$ を与える葉節点が G'_{min} であるとすると，$p_0(v,G'_{min}) \le p(v,G'_{min})$ だから，定義 4 により $h_0(v) = \min_{k=1}^{m}\{p_0(v,G_k)\} \le p_0(v,G'_{min})$ である．これより次式が成立する．

$$h_0(v) \le h(v) \tag{15}$$

ゆえに，$h_0(v) \le h(v) \le h^*(v)$ である．　□

3.4　探索手続

推定コストの更新を含む探索手続の概要を**図 4** に示す．この探索手続は A*アルゴリズムに推定コストの更新処理を付加したものである．図中の行頭の数字は行番号である．節点オブジェクトには，**表 1** に示すデータフィールドとメソッドが定義されているものとする．節点オブジェクトのリスト構造を構成するリストオブジェクトには，**表 2** に示すメソッドが定義されているものとする．

```
S.parent ← S;  g*(S) ← 0;  c*(S,S) ← 0;
open.append(S);
while open ≠ ∅ do
    v ← open.pop();
    u ← v.parent;
    v.fvalue ← g*(u) + c*(u,v) + h(v);
    if v.isGoal() then
        return v;
    end if
    for all (u',v') ∈ v.EdgeSet() do
        c(u',v') ← max{c*(u,v), c(u',v')};
    end for
    for all w ∈ v.SuccSet() do
        w.parent ← v;
        w.fvalue ← g*(v) + c(v,w) + h(w);
        open.append(w);
    end for
    open.sort();
end while
return null;
```

図 4　提案手法の探索手続き
Fig. 4　Search procedure of the proposed method.

表 1 節点オブジェクトに関するデータフィールドとメソッド
Table 1 Data fields and methods in an object for a tree node.

データフィールド	格納するデータ
$parent$	親節点のオブジェクト
$fvalue$	この節点の評価関数値
メソッド	機能
isGoal()	目標節点であれば True, そうでなければ False を返す
SuccSet()	子節点の集合を返す
EdgeSet()	部分木 $T_s(v)$ の辺の集合を返す

表 2 リストオブジェクトのメソッド
Table 2 Methods in an object for a list structure.

メソッド	機能
pop()	先頭要素を返す
append(v)	節点オブジェクト v を要素として リストに追加する
sort()	要素を評価関数値の昇順にソートする

S は初期節点，u は v の親節点，v は展開対象となる節点，w は v の子節点にそれぞれ対応する節点オブジェクトである．$open$ は展開対象となる節点を評価関数値の昇順に格納するリストオブジェクトである．木構造のすべての辺には，推定コストの初期値が割り当てられているものとする．

まず，初期節点 S に関する初期化として，S の形式的な親節点，S から S への真のコストを代入する（行 1）．次に，リスト $open$ に初期節点を追加する（行 2）．以後，$open$ から展開する節点の取り出しと追加を $open$ が空になるまで繰り返す．節点 v を選択するときに観測される $c^*(u,v)$ を用いて評価値を再計算する（行 3–6）．v が目標節点であるかどうか終了判定を行う．v の評価値は再計算されているため，v が葉節点であり，かつ，$open$ の先頭の節点の評価値が $v.fvalue$ より大きければ，v.isGoal() は True を返し，v を解として探索を終了する．v が葉節点であり，かつ，$open$ の先頭の節点の評価値が $v.fvalue$ 以下であれば，v.isGoal() は，v を $open$ に再度追加し False を返す．また，v が葉節点ではない場合は False を返す（行 7–9）．v が目標節点でない場合には，$c^*(u,v)$ を利用し，部分木 $T_s(v)$ 内のすべての辺 (u',v') の推定コストを更新する．ただし，$v=S$ のとき，形式的に $c^*(S,S)=0$ としたので，推定コストの実質的な更新はない（行 10–12）．v を展開して得られるすべての子節点 w のそれぞれについて，親節点を記録したのち，w を経由する経路の評価関数値を算出し，リスト $open$ に加える（行 13–17）．リスト $open$ の要素を評価関数値の昇順に整列する（行 18）．

3.5 手法の特性

定理 1 提案手法は最適解を発見する．

証明 4（定理 1） 評価関数が $f_{A*}(v)=g_{A*}(v)+h_{A*}(v)$ である A*アルゴリズムを木構造に適用するとき，目標節点までの推定コストを表す $h_{A*}(v)$ がその真のコストを超えないならば，A*アルゴリズムは最適解を発見して停止することが知られている [1]．提案手法は評価関数が $f(u)=g^*(u)+c(u,v)+h(v)$ であるから，次式が成立すれば A*アルゴリズムと同様に提案手法について最適解の発見を保証できる．

$$c(u,v)+h(v) \le c^*(u,v)+h^*(v) \tag{16}$$

まず，補題 1 により，各辺の推定コスト $c(u,v)$ はその真のコスト $c^*(u,v)$ を超えないから，次式が成立する．

$$c(u,v) \le c^*(u,v) \tag{17}$$

次に，補題 3 により，推定目標コスト $h(v)$ は目標コスト $h^*(v)$ を超えないから，次式が成立する．

$$h(v) \le h^*(v) \tag{18}$$

式 (17) と式 (18) の辺々を加え次式が成立する．

$$c(u,v)+h(v) \le c^*(u,v)+h^*(v) \tag{19}$$

これは，式 (16) であり，u からどの子節点 v を経由する経路の推定コストも，その真のコストを超えないことを示している．したがって，提案手法は最適解を発見して停止する． □

定理 2 ある問題空間において提案手法が解を発見するまでに展開する節点数は，同一の問題空間に適用された A*アルゴリズムが解を発見するまでに展開する節点数以下である．

証明 5（定理 2） 提案手法と A*アルゴリズムのいずれも最適解を発見して停止する．最適解に至る経路の真のコストを f^* とするとき，提案手法では $f(v) \le f^*$ となる v が Open リストにあれば，v を必ず展開する．A*アルゴリズムもまた，$f_{A*}(v) \le f^*$ となる v が Open リストにあれば v を必ず展開する．A*アルゴリズムにおける v から目標節点に至る経路の推定コスト $h_{A*}(v)$ は，更新されないため，提案手法の $h(v)$ 以下である．よって，v が提案手法で展開されるときには，$f_{A*}(v) \le f(v) \le f^*$ が成立する．すなわち，提案手法で展開される節点は，A*アルゴリズムでも展開される． □

4. シミュレーション

4.1 実行内容

前提 1 を満たす深さ $D \in \{4,6,8,10,12,14,16\}$ の完全二分木を問題空間とする．いずれの深さにおいても，値の範囲が 0 から上限 $X \in \{100, 100000\}$ である一様乱数 R を用いて，各辺の真のコストの値と推定コストの初期値を付与する．推定コストの初期値 E は，経路に沿って非減少とは

限らない割当て方法 M_1 と，真のコストと同様に経路に沿って非減少となる割当て方法 M_2 の 2 通り $E \in \{M_1, M_2\}$ とする．$(d, x, e) \in D \times X \times E$ である問題空間 $T(d, x, e)$ をそれぞれ 100 例生成する．同一の問題空間に対照手法と提案手法を適用し，それぞれの手法が解を発見するまでに展開する節点数と，生成する節点数を記録する．対照手法は A*アルゴリズムであり，推定コストの初期値を使用して経路を評価する．A*アルゴリズムでは推定コストの更新は行われない．

実験 1 は $T(D, 100000, M_1)$，実験 2 は $T(D, 100, M_1)$ を対象として，それぞれ提案手法と対照手法を比較する．また，実験 1 と実験 2 からコストの上限値の差に起因する相違の有無を検討する．実験 3 は $T(D, 100000, M_2)$，実験 4 は $T(D, 100, M_2)$ を対象として，それぞれ提案手法と対照手法を比較する．また，実験 1 と実験 3，実験 2 と実験 4 のそれぞれにおいて，推定コストの初期値の違いに起因する相違の有無を検討する．

4.2　問題空間の生成

前提 1 が成立する問題空間になるように，次の方法で各辺にそれぞれの値を付与する．

まず，初期節点 S とその子節点 $v \in \{v_1, v_2\}$ の間の各辺 (S, v) に，一様乱数 R を用いて真のコストを割り当てる．

次に，再帰的に，すでに真のコストが割り当てられた辺 (u, v) について，v とその子節点 $w \in \{w_1, w_2\}$ の間の各辺 (v, w) に対し，一様乱数 R を用いて真のコストを割り当てる．ただし，一様乱数 R で生成される値が，辺 (u, v) に割り当てられている値 r より小さい間は破棄し，最初に現れた r 以上の値を採用する．

すべての辺に真のコストを割り当て終えたら，最後に，すべての辺に推定コストの初期値を割り当てる．実験 1，実験 2 では，各辺の $U = c^*(v, w)$ を上限値として生成した一様乱数の値をその辺 $(v.w)$ の推定コストの初期値として割り当てる．実験 3，実験 4 では，辺 (v, w) の先行辺 (u, v) の推定コスト $L = c(u, v)$ を下限値，真のコスト $U = c^*(v, w)$ を上限値として生成した一様乱数の値を辺 (v, w) の推定コストの初期値ととして割り当てる．

4.3　結果と考察

4.3.1　展開節点数の比較

問題空間は，対照手法による展開節点数 N_A と，提案手法による展開節点数 N_P の値の組 (N_A, N_P) により特徴づけられる．横軸に N_A，縦軸に N_P の値をとる座標平面上にプロットすると，プロットされた点が 1 つの問題空間 $T(d, x, e)$ に対応する．特に，深さ $D = 8$ の場合について，実験 1 の $T(8, 100000, M_1)$ の結果を**図 5** に，実験 3 の $T(8, 100000, M_2)$ の結果を**図 6** にそれぞれ図示する．

いずれの図においても図中の原点を通る直線は，対照手法と提案手法の展開節点数が等しくなる境界線である．この境界線上とその下側の領域は，提案手法の展開節点数が対照手法の展開節点数以下となる領域である．いずれの図においても 100 例すべての問題空間が，境界線上かまたはその下側の領域に存在する．これは定理 2 を実験的に示したものである．図 6 に見られるように，両手法とも推定コストの初期値が経路に沿って非減少である場合には，それとは限らない場合と比較して，展開節点数は平均的には減少する．特に対照手法の展開節点数の減少は顕著であり，提案手法と同数となる場合がある．なお，実験 2 の $T(8, 100, M_1)$，実験 4 の $T(8, 100, M_2)$ についても，それぞれ図 5，図 6 と同様の傾向を示した．

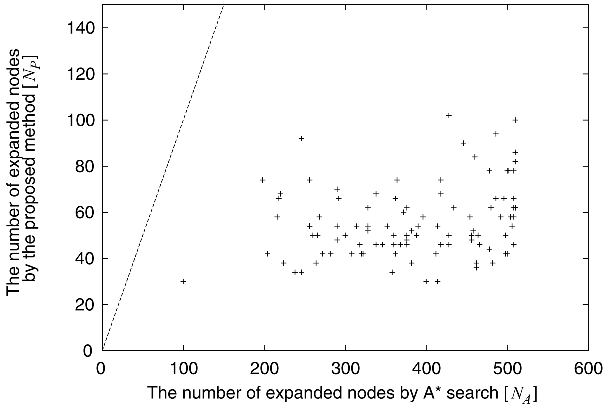

図 5　実験 1：展開節点数の比較（$T(8, 100000, M_1)$）

Fig. 5　The comparison of the number of expanded nodes by the proposed method and A* search on $T(8, 100000, M_1)$.

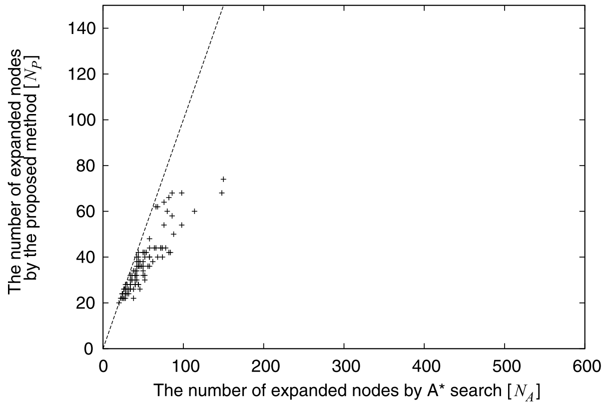

図 6　実験 3：展開節点数の比較（$T(8, 100000, M_2)$）

Fig. 6　The comparison of the number of expanded nodes by the proposed method and A* search on $T(8, 100000, M_2)$).

4.3.2　有効分岐因子の値の比較

深さ $d \in D$ の問題空間ごとに，対照手法と提案手法における 100 例の生成節点数の平均値 G を用いて，式 (20) を満たす有効分岐因子 [10] の値 B をそれぞれ算出し表に示す．

$$B + B^2 + \cdots + B^d = G \tag{20}$$

また，横軸に深さ $d \in D$，縦軸に有効分岐因子 B をとり，

表 3　実験 1：生成節点数と有効分岐因子（$T(D, 100000, M_1)$）
Table 3　The number of generated nodes and the effective branching factor by the proposed method and A* search on $T(D, 100000, M_1)$.

Search tree		A* search		Proposed method	
d	N	G	B	G	B
4	31	25.08	1.89	16.94	1.67
6	127	98.96	1.91	33.94	1.53
8	511	383.30	1.92	55.82	1.43
10	2,047	1,492.98	1.93	84.76	1.37
12	8,191	5,951.98	1.94	119.56	1.33
14	32,767	23,502.12	1.95	161.98	1.30
16	131,071	92,069.72	1.95	212.44	1.27

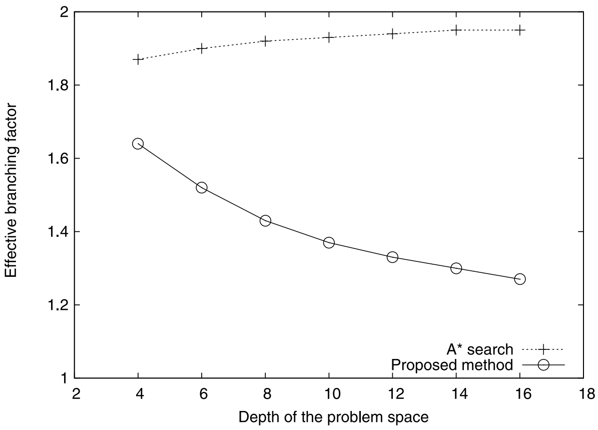

図 7　実験 1：有効分岐因子の比較（$T(D, 100000, M_1)$）
Fig. 7　The comparison of the effective branching factor by the proposed method and A* search on $T(D, 100000, M_1)$.

表 4　実験 2：生成節点数と有効分岐因子（$T(D, 100, M_1)$）
Table 4　The number of generated nodes and the effective branching factor by the proposed method and A* search on $T(D, 100, M_1)$.

Search tree		A* search		Proposed method	
d	N	G	B	G	B
4	31	26.16	1.89	17.04	1.64
6	127	102.30	1.92	32.68	1.51
8	511	394.64	1.91	52.22	1.41
10	2,047	1,569.42	1.94	76.56	1.35
12	8,191	6,227.46	1.95	105.02	1.31
14	32,767	25,106.30	1.96	131.02	1.27
16	131,071	97,630.80	1.96	165.50	1.25

D と B の関係を図示する．実験 1 の結果を**表 3** と**図 7** に，実験 2 の結果を**表 4** と**図 8** に，実験 3 の結果を**表 5** と**図 9** に，実験 4 の結果を**表 6** と**図 10** にそれぞれ示す．

実験 1（表 3・図 7）は，推定コストの初期値が経路に沿って非減少とは限らない場合の結果である．対照手法は問題空間が深くなるにつれて，有効分岐因子の値がわずかに増加する．これは対照手法では問題空間が深くなるにつれて，解を発見するまでにより広い範囲の探索が必要になることを意味する．問題空間が完全二分木であることを考慮すると，有効分岐因子の上限は 2 であるから，ほとんどの節点を生成している．一方，提案手法は，問題空間が深くなるにつれて，有効分岐因子の値が減少する．これは問題空間が深くなるにつれてより狭い範囲の探索で解が発見できることを意味する．この傾向は，実験 2 の結果（表 4・

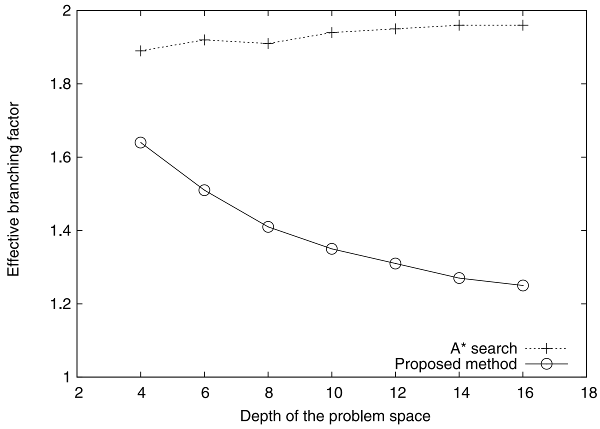

図 8　実験 2：有効分岐因子の比較（$T(D, 100, M_1)$）
Fig. 8　The comparison of the effective branching factor by the proposed method and A* search on $T(D, 100, M_1)$.

表 5　実験 3：生成節点数と有効分岐因子（$T(D, 100000, M_2)$）
Table 5　The number of generated nodes and the effective branching factor by the proposed method and A* search on $T(D, 100000, M_2)$.

Search tree		A* search		Proposed method	
d	N	G	B	G	B
4	31	15.24	1.58	13.50	1.51
6	127	30.36	1.48	24.16	1.40
8	511	50.60	1.41	36.74	1.33
10	2,047	75.14	1.35	51.16	1.28
12	8,191	102.30	1.31	66.58	1.25
14	32,767	132.56	1.27	83.40	1.22
16	131,071	164.28	1.24	99.50	1.19

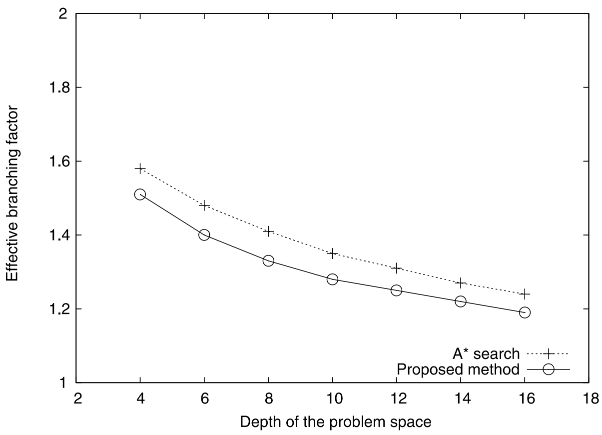

図 9　実験 3：有効分岐因子の比較（$T(D, 100000, M_2)$）
Fig. 9　The comparison of the effective branching factor by the proposed method and A* search on $T(D, 100000, M_2)$.

表 6 実験 4：生成節点数と有効分岐因子（$T(D, 100, M_2)$）
Table 6 The number of generated nodes and the effective branching factor by the proposed method and A* search on $T(D, 100, M_2)$.

Search tree		A* search		Proposed method	
d	N	G	B	G	B
4	31	14.36	1.55	13.18	1.50
6	127	30.22	1.48	24.34	1.41
8	511	50.76	1.41	36.74	1.33
10	2,047	80.66	1.36	54.18	1.29
12	8,191	157.06	1.37	88.72	1.29
14	32,767	399.42	1.40	182.50	1.31
16	131,071	1,284.48	1.45	510.00	1.36

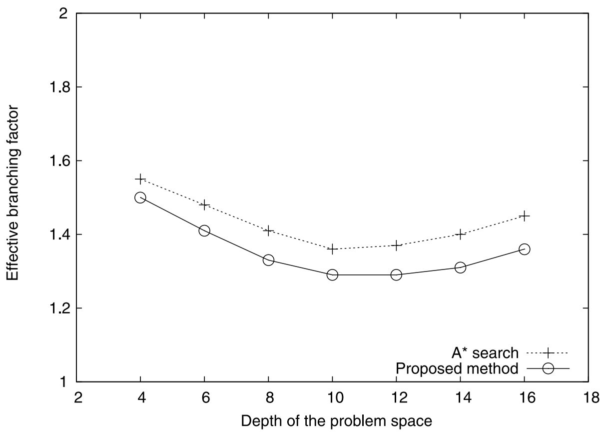

図 10 実験 4：有効分岐因子の比較（$T(D, 100, M_2)$）
Fig. 10 The comparison of the effective branching factor by the proposed method and A* search on $T(D, 100, M_2)$.

図 8）にも現れる．

実験 3 の結果（表 5・図 9）は，推定コストの初期値を経路に沿って非減少に設定した場合である．提案手法にはややおよばないものの，対照手法も問題空間が深くなるにつれてより狭い探索範囲で解が発見できることを示している．また，提案手法の有効分岐因子の値は，実験 1 の結果よりやや小さい．

推定コストの初期値を経路に沿って非減少となるよう設定すると，それとは限らない場合と比較して，評価関数の精度は高くなる．そのため，対照手法はより狭い探索範囲で解を発見する．提案手法は，推定値の更新により評価関数の精度がさらに高まる．そのため，推定コストの初期値が経路に沿って非減少であるとは限らない場合よりもさらに狭い探索範囲で解を発見する．

実験 4 は，コストの上限値が小さく，かつ推定コストの初期値を経路に沿って非減少に設定した場合である．表 6・図 10 から，対照手法，提案手法ともに有効分岐因子の値が，$d = 10$ を境に増加に転じていることが分かる．

節点 u とその子節点 v が最適経路上にあるとき，次に展開する節点として v が選ばれると，コストの真の値 $c^*(u, v)$ が観測される．これにより，この時点の最適経路のコストは $f(u)$ から $f(v)$ に $c^*(u, v) - c(u, v)$ だけ増加する．対照手法，提案手法ともに最適解を発見するから，v が展開されるときは，$f(u) < f(w) \leq f(v)$ となるいかなる節点 w も必ず展開されている．コストの上限値が小さいとき，深い問題空間ではコストの真の値として同じ値を割り当てられた辺が多くなり，経路のコストの差が小さくなる．そのため前式を満たす w が多く現れると，展開節点数とともに生成節点数が増加し，有効分岐因子の値が増加に転ずるものと考えられる．

5. おわりに

葉節点に向かう経路に沿った辺に順次大きなコストが付与されるという特徴を持った木構造における探索手法を提案した．提案手法は，A*アルゴリズムをベースとして，節点の選択時に観測される真のコストを利用し，推定コストを更新しながら探索を進めるものである．更新対象になるのは，初期節点を根とする場合を除き，展開対象となる節点を根とする部分木内のすべての辺の推定コストである．推定コストは評価関数の適格性を保つように更新するため，最適解の発見が保証される．また，推定コストの更新をともなう評価関数の値は，更新をともなわない場合の値に対しつねに支配的である．したがって，提案手法が探索中に展開する節点数は，同一の問題空間において推定コストを更新しない手法により展開される節点数以下であることが保証される．これらは A*アルゴリズムの特性であることから，提案手法が A*アルゴリズムを拡張しながらもその特性を継承することを示している．

これらの特性をシミュレーションにより検証した．推定コストを更新しない手法を対照手法として，提案手法と対照手法とを同一の問題空間に適用する．対照手法は推定コストの初期値を評価関数値とする A*アルゴリズムである．問題空間は完全二分木とし，上限値を定めた乱数を利用して，真のコストと推定コストを割り当てる．

まず，提案手法と対照手法において発見される解が一致することにより，各問題空間において最適解を発見していることが確認された．次に，最適解の発見までに展開される節点数を比較することにより，提案手法による展開節点数は対照手法のそれ以下であることが確認された．さらに，提案手法の有効分岐因子の値は，対照手法のそれを下回るという優位な結果が得られた．これは，提案手法が最適解を発見するまでの探索範囲が，対照手法のそれより狭いため，効率が良いことを意味する．

真のコストと同様に，推定コストの初期値が経路に沿って非減少である場合，提案手法も対照手法もともに，有効分岐因子の値は，問題空間が深くなるにつれて低下する．推定コストの初期値が経路に沿って非減少とは限らない場合においては，問題空間が深くなるにつれて，提案手法の

有効分岐因子の値は低下することに対し，対照手法ではその値は増加する．このことは，本稿が対象とする問題空間においては，提案手法は対照手法より推定コストの精度の低さによる影響を受けにくいことを示している．

問題空間を特徴づける 1 つの要素としてコストの分布を考えると，本稿で対象としたように偏りが見られる場合と，そうでない場合とに大別できる．本稿では，問題空間全体で先行する辺よりコストが大きいという偏りを前提としているが，この偏りが一部分だけにとどまる問題空間も考えうる．そのような問題空間に対する手法として，提案手法と対照手法を組み合わせ，両手法の長所を活かす手法を検討することが残された課題である．

参考文献

[1] Hart, P., Nilsson, N. and Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths, *IEEE Trans. Systems Science and Cybernetics*, Vol.4, No.2, pp.100–107 (1968).

[2] Russell, S. and Norvig, P.: *Artificial Intelligence: A Modern Approach*, 3rd edition, pp.103–104, Prentice Hall (2009).

[3] 吉岡伸也，橋本浩良，上坂克巳：車線数と交通量が所要時間に及ぼす影響に関する実証的研究，土木計画学研究・講演集（CD-ROM），Vol.39, p.315 (2009).

[4] 鳥海重喜，中村幸史，田口　東：通勤電車の遅延計算モデル，オペレーションズ・リサーチ，Vol.50, No.06, pp.409–416 (2005).

[5] 仮屋崎圭司，岩倉成志，森地　茂：第 93 回運輸政策コロキウム　都市鉄道の運行ダイヤ過密化に伴う列車遅延の波及に関する研究，運輸政策研究，Vol.11, No.4, pp.111–118 (2009).

[6] 萩原武司, 小澤友記子, 北澤俊彦：データオリエンティッドなイベント時渋滞予測モデル分析，土木学会年次学術講演会講演概要集（CD-ROM），Vol.64, No.Disk2, pp.IV–003 (2009).

[7] E., R. and Korf: Real-time heuristic search, *Artificial Intelligence*, Vol.42, No.2-3, pp.189–211 (1990).

[8] Hernández, C. and Meseguer, P.: LRTA*(k), *Proc. 19th International Joint Conference on Artificial Intelligence*, San Francisco, CA, USA, pp.1238–1243, Morgan Kaufmann Publishers Inc. (2005).

[9] Hernández, C. and Meseguer, P.: Improving LRTA*(k), *Proc. 20th International Joint Conference on Artifical Intelligence*, San Francisco, CA, USA, pp.2312–2317, Morgan Kaufmann Publishers Inc. (2007).

[10] Nilsson, N.J.: *Problem-Solving Methods in Artificial Intelligence*, pp.72–75, McGraw-Hill Pub. Co. (1971).

芦田 昌也　（正会員）

1966 年生．1990 年大阪大学工学部卒業．1996 年大阪大学大学院工学研究科博士後期課程単位取得退学．1996 年和歌山大学経済学部講師．1998 年同准教授．人工知能技術の応用に関する研究に従事．電子情報通信学会，電気学会各会員．

瀧 寛和　（正会員）

1956 年生．1978 年大阪大学基礎工学部卒業．1980 年大阪大学大学院基礎工学研究科博士前期課程修了．1980 年三菱電機入社．1986 年（財）新世代コンピュータ技術開発機構出向．1990 年三菱電機帰任．1991 年大阪大学工学博士．1998 年和歌山大学システム工学部教授．脳科学の人工知能への応用研究に従事．人工知能学会評議員，電子情報通信学会，電気学会，IEEE，IEEE-CS 各会員．

