ある種の最適2-3木

著者	清水 道夫
雑誌名	長野県短期大学紀要
巻	46
ページ	73-76
発行年	1991-12
URL	http://id.nii.ac.jp/1118/00000414/

ある種の最適2—3木

清 水 道 夫

1. まえがき

有効なデータ構造を表現するものに，ＡＶＬ木や２—３木など平衡木とよばれるものがある（文献１)。本論文では，２—３木の探索コストに関する最適構成問題を扱う。２—３木は，２個の子を持つ２分節と３個の子を持つ３分節からなり，葉のレベルが一様な木である。

２—３木の探索コストは，根から葉に向かって節および見出しをたどるとき，節の訪問回数と見出しの比較回数の和として表される。ここでは，節の訪問回数を無視し，比較回数のみのコストに着目する。A. L. Rosenberg と L. Snyder は，見出しの重みが一様なとき，見出しの比較コストが最小の２—３木を特徴づけ，その構成アルゴリズムを示した（文献２)。このとき，３分節内の２つの見出しの比較順序については，常に小さい方から比較すると仮定したものであるが，今後の研究課題として，大きい方の見出しからの比較を許す場合の考察を与えている。本論文はこの問題に関連したもので，３分節内で最初に比較される見出しが等確率つまり１／２で選択されるとした場合を考える。この仮定のもとでのコストを定義し，最適な２—３木を特徴づける。

最適木はデータ構造の初期構造を与える。２—３木はデータ構造として用いられるとき，見出しの挿入や削除によって動的に形を変えていくが，アクセスの回数に比べて更新の頻度が少ない場合は，初期構造が比較的長く保存されると考えられるから，最適木を考えることは意味があろう。

2. コストの定義

２—３木の３分節には２個の見出しを持ち，２分節には１個の見出しを持つ。見出しは辞書式順序に基づく大小関係で並んでおり，能率のよい探索を可能にしている。２分節および３分節の見出しは，左子の見出しよりも大きく，右子の見出しよりも小さい。ただし，３分節の２個の見出しを a, b とし，中子の見出しを m とすると，$a<m<b$ が成り立つ。３分節内での大きい方の見出しからの比較を許す場合の探索は次のように行う。

今，見出し k を探索するとき，見出し a, b ($a<b$) を持つ３分節を訪問したとする。最初に k と a を比較する場合は，$k<a$ のとき左子をたどり，$k>a$ のとき b と比較する。そして，$k<b$ のとき中子をたどり，$k>b$ のとき右子をたどる。ただし，$k=a$ または $k=b$ のときは探索終了。逆に，k を b から比較する場合は，$k>b$ のとき右子をたどり，$k<b$ のとき a と比較する。そして，$k<a$ のとき左子をたどり，$k>a$ のとき中子をたどる。このように探索するとき，３分節内で最初に比較される見出しが等確率で選択される場合のコストを考える。

〔定義１〕　コストの定義を明確にするために，次のような記号を用いる。

(1)　２—３木のすべての節にたいして，次のように定義されるアルファベット $\{s,\ m\}$ 上の記号列を対応づける。

(a)　根を Λ とする。

(b)　記号列に x 対応する節の左子または右子を xs

とする。記号列xに対応する3分節の中子をxmとする。

(2)　2分節の集合をN_2，3分節の集合をN_3とする。

(3)　節xに含まれる見出しを〔x〕で表す。

〔定義2〕　各見出しの比較回数の期待値（EC と略記）を表す関数placeを次のように定義する。

(a)

$$\mathrm{place}([A]) = \begin{cases} 1 & : A \in N_2 \\ 1.5 & : A \in N_3 \end{cases}$$

(b)　$x \in N_2 + N_3$のとき

$$\mathrm{place}([xs]) = \mathrm{place}([x]) + \begin{cases} 1 & : xs \in N_2 \\ 1.5 & : xs \in N_3 \end{cases}$$

$x \in N_3$のとき

$$\mathrm{place}([xm]) = \mathrm{place}([x]) + \begin{cases} 1.5 & : xm \in N_2 \\ 2 & : xm \in N_3 \end{cases}$$

高さ3の2—3木の例および各見出しのECを図1に示す。コストは各見出しのECの総和で表される。

〔定義3〕　2—3木Tのコスト$C(T)$を次式で表す。

$$C(T) = \sum_{x \in N_2} \mathrm{place}([x]) + 2 \sum_{x \in N_3} \mathrm{place}([x])$$

図1の木のコストは67.5になる。

3.　最適木の構成

n個の葉を持つコスト最小の2—3木は，ECが最小のn—1個の見出しを逐次挿入することによって得られる。2—3木の根のレベルを0とし，葉のレベルをLとする。見出しの挿入は，レベルL—1の2分節を3分節に変換することによって行う。なお，見出しの辞書式順序はコストに関係しないので考えない。

〔定理1〕　次の構成法によって構成した2—3木はコスト最小である。ここに，n個の葉を持つコスト最小木をTとし，Tの根から葉までの路上の3分節数をtとする。

（構成法）

〔I〕　n＝1～4の場合は図2に示す。

〔II〕　$n \geqq 5$のときを示す。$2^i \leqq n \leqq 3\cdot 2^{i-1}$のとき，$T$の$t=0$の路上に挿入し，$3\cdot 2^{i-1} < n < 2^{i+1}$のとき，$T$の$t=1$の路上に挿入する。このとき，次の変換1と2を随時行う。なお，$t=1$の路上に挿入するときは，3分節を根とする部分木の左部分木か右部分木に挿入する。

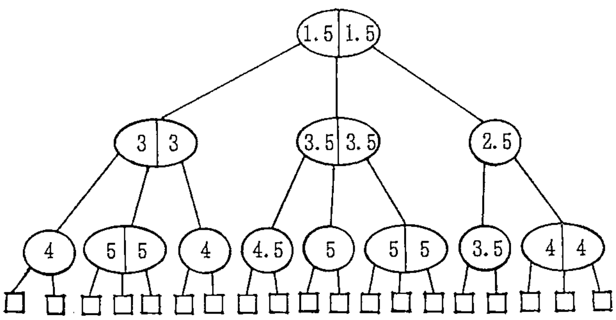

図1　2—3木の例

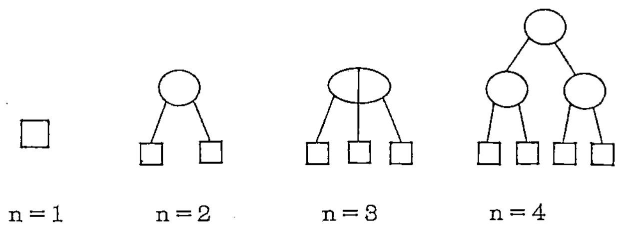

図2　2—3木（n＝1～4）

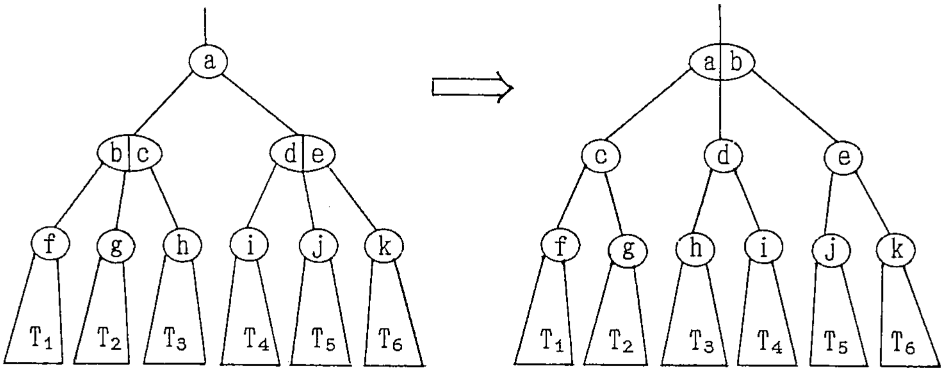

図3　変換1

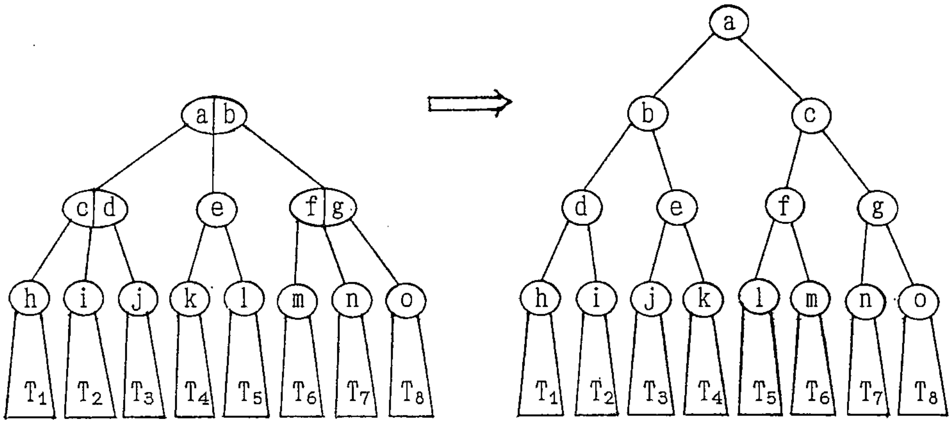

図4　変換2

変換1：2分節の左子と右子が3分節のとき，図3のような変換を行う。

変換2：根およびその左子と右子が3分節のときは，図4のように分割して，高さを1増やす。□

この定理を証明するために，つぎの補題1と補題2を与えておく。

〔補題1〕変換1によってコストが変化しない。

（証明）図3の左図の見出し a のＥＣを α とすると，見出し b～k のＥＣはそれぞれ α+1.5, α+1.5, α+1.5, α+1.5, α+2.5, α+3, α+2.5, α+2.5, α+3, α+2.5となり，総和は11α+22である。一方右図では，見出し a～k のＥＣがそれぞれα+0.5, α+0.5, α+1.5, α+2, α+1.5, α+2.5, α+2.5, α+3, α+3, α+2.5, α+2.5となるから，総和は11α+22である。ただし，見出し f～k の2分節が葉の場合は，ＥＣの総和が5α+6となる。また，部分木 T_1～T_6 は同じ大きさの完全2分木であるから，全体のコストは変化しない。□

〔補題2〕変換2によってコストが減少する。

（証明）図4の左図の見出し a～o のＥＣはそれぞれ，1.5, 1.5, 3, 3, 3, 3, 3, 4, 4.5, 4, 4, 4, 4, 4.5, 4だから，総和は51となる。一方右図は完全2分木であるから，ＥＣの総和は1＋2×2＋3×4＋4×8＝49となる。ただし，レベル2つまり h～o は葉の場合もあり，このとき左図は18で右図は17になる。また，部分木 T_1～T_8 は完全2分木であるから，レベル2の見出しのＥＣによってコストが決定する。したがって，全体のコストは減少する。□

（定理1の証明）〔I〕は明らかだから〔II〕を示す。変換1はコストが変わらないが，レベル i

-1に常に2分節が存在することを保証するから，見出しの挿入を可能にする。変換2はコストが減少するから，なるべく高さの高い木のコストが最小になることを示している。したがって，最適木の高さは，$2^i \leqq n \leqq 2^{i+1}$のとき$i$になる。$T$の$t=0$の路上に見出しを挿入するのは，定義2よりＥＣが最小だからである。しかし，$3 \cdot 2^{i-1} < n < 2^{i+1}$のとき，$t=0$の路がないから$t=1$の路上に挿入する。そのとき，3分節を根とする部分木の左部分木か右部分木に挿入するのは，定義2より中部分木に挿入するよりもＥＣが小さいからである。なお，$n=3 \cdot 2^{i-1}$のときは，Tの根が3分節で，左，中，右部分木が高さ$i-1$完全2分木である。 □

4. 最適木のコスト

3.では見出しの逐次挿入によって最適木を構成したが，そのコストを考えやすくするために，もっと直接的な構成法を与える。次の構成法はあらかじめ最適木の高さを決定し，レベル$i-1$の2分節だけを3分節に置き換えるというものである。

（構成法2）

（Ⅰ）$2^i \leqq n \leqq 3 \cdot 2^{i-1}$のとき

高さiの完全2分木のレベル$i-1$の2分節のうち，$(n-2^i)$個を3分節に置き換える。

（Ⅱ）$3 \cdot 2^{i-1} < n < 2^{i+1}$のとき

3分節を根とする左，中，右部分木をそれぞれ高さ$i-1$の完全2分木とする。左，右部分木のレりベル$i-1$の2分木のうち$(n-3 \cdot 2^{i-1})$個を3分節に置き換える。

3分節に置き換えるだけの十分な数の2分節がレベル$i-1$に用意されていることに注意すれば，構成法2によって作られた木が，n個の葉を持つコスト最小木であることは明らかである。ここで，そのコストを示す。

〔定理2〕n個の葉を持つ最適木Tのコスト$C(T)$は次式で与えられる。

（Ⅰ）$2^i \leqq n \leqq 3 \cdot 2^{i-1}$のとき

$$C(T) = (i+1)n-2^{i+1}+1$$

（Ⅱ）$3 \cdot 2^{i-1} < n < 2^{i+1}$のとき

$$C(T) = (i+1.5)\, n-2^{i+1}-3 \cdot 2^{i-2}+1$$

（証明）（Ⅰ）のみ導く。（Ⅱ）も同様に導くことができる。まず，

$$f(x) = 1+x+x^2+\cdots+x^i = \frac{x^{i+1}-1}{x-1}$$

とおく。これを微分すると

$$f'(x) = 1+2x+3x^2+\cdots+i\,x^{i-1} = \frac{i\,x^{i+1}-(i+1)\,x^i+1}{(x-1)^2}$$

となる。ところで，高さiの完全2分木のコストは$f'(2)$で表される。

$$f'(2) = i\,2^{i+1}-(i+1)2^i+1 = i\,2^i-2^i+1$$

つぎに，置き換えによる増加分を考える。レベル$i-1$の2分節の見出しのＥＣはiだが，これを3分節に置き換えて見出しを挿入すると，$2(i+0.5)$になる。よって，節1個当たりの増加分は$i+1$だから，これを$(n-2^i)$倍して$f'(2)$に加えると$C(T)$が求まる。

$$C(T) = f'(2) + (i+1)(n-2^i) = (i+1)n-2^{i+1}+1$$ □

文献

(1) D. E. Knuth:" The Art of Computer Programming 3 : Sorting and Searching", Addison-Wesley(1973).

(2) A. L. Rosenberg and L. Snyder:" Minimal-Comparison 2, 3-Trees", SIAM J. Comput., Vol. 7, No. 4, pp. 465—480(1978).

(3) R. E. Miller, N. Pippenger, A. L. Rosenberg and L. Snyder:" Optimal 2, 3-Trees", SIAM J. Comput., Vol. 8, No. 1, pp. 42—59(1979).

(4) A. V. Aho and J. E. Hopcroft and J. D. Ullman:" Data Structures and Algorithms", Addison-Wesley(1983).

(5) 清水道夫:" 高さ平衡木にたいする総数の評価と最適構成に関する研究", 大阪大学博士論文(1986)。