

線形法における探索アルゴリズムの解析†

中村 良三** 大島 洋一†††

線形法を用いた見出し探索法の解析では、見出しの探索頻度のみならずバケットサイズをも考慮して、一般化したモデルにおける挿入・探索アルゴリズムの厳密な解析は明らかでない。すなわち、従来の解析では、バケットサイズを1と固定したもとの厳密な解析は提示されているが、バケットサイズをパラメータとしたときの厳密な解析は明らかでない。本稿では、見出しの探索頻度とバケットサイズとを考慮して一般化したモデルにおける挿入・探索アルゴリズムの厳密な解析を提示する。特に、提示する解析では、表の大きさが M でバケットサイズが b である分散表に、 n 個の見出しを一様に分散したとき、ある番地のバケットに空きがあるような見出し番地列の数を評価する関数と、ある番地の空きがあるバケットに続く連続した幾つかのバケットがすべて見出しで占有されるような見出し番地列の数を評価する関数とを定式化し考察している。

1. ま え が き

いろいろなアルゴリズムすなわち算法の中で、どれが最良であるかを定めるためには、アルゴリズムの平均的な振る舞いを調べたり、また、ときには何らかの意味で最良であるか否かを考察する必要がある。このような算法解析は、計算機科学分野において重要であり、また興味のある課題である。

これまでに、種々のアルゴリズムに対して、できるだけモデルを一般化したもとの解析が提示されてきた^{2),3)}。しかし、まだいろいろなアルゴリズムにおいて、一般化されたモデルでの厳密な解析が未解決である。

本稿での解析の対象となる算法は、見出し探索法でよく用いられる分散記憶法のひとつである線形法である。この技法に対するアルゴリズムの解析の過程を振り返って見ると次のようになる。1962年、はじめて Schay と Spruth によって、バケットに格納できるレコードすなわち見出しの個数をひとつと固定し、かつ見出しの探索頻度を一様と仮定したもとの近似的な解析が発表されている¹⁾。その後1973年、Knuth によって、上記と同じモデルに対する厳密な解析とバケットサイズを考慮した近似的な解析とが提示されている²⁾。近年、中村らによって、上記のモデルを拡張して、見出しの探索頻度を考慮できるようにしたモデルでの厳密な解析⁴⁾とバケットサイズを考慮した近似的

な解析⁵⁾とが提示されている。

しかし、前述のように、バケットサイズを任意の大きさとした、すなわちバケットサイズをパラメータとしたモデルでの厳密な解析は未解決な課題として残されている。

本論文では、線形法においてバケットサイズと見出しの探索頻度とを考慮して一般化したモデルのもとの挿入・探索アルゴリズムの厳密な解析を提示する。

以下、2章では、線形法における挿入・探索アルゴリズムを概観する。次に、3章においては、従来の解析を検討したあと、4章において、一般化したモデルにおける厳密な解析を提示する。特に、提示する解析では、表の大きさが M でバケットサイズが b である分散表に、 n 個の見出しを一様に分散したとき、ある番地のバケットに空きがある見出し番地列の数を評価する関数 $f_b(M, n)$ と、ある番地の空きがあるバケットに続く連続した $h-1$ 個のバケットがすべて見出しで占有されるような見出し番地列の数を評価する関数 $t_b(h, n)$ とを定式化して考察する。

2. 線形法におけるアルゴリズム

分散記憶法は、レコードを登録するときには、レコードの固有の見出しに対して演算を行い、その演算結果をそのレコードの格納場所の位置づけに使用し、探索するときには、登録時と同様の手続きによって、目的のレコードを探す方法である。このとき、見出しに対する演算結果は、計算機の記憶領域に実現された表すなわち分散表の指標を示す番地となる。このように見出しから番地を生成する演算を行う関数を分散関数といい、この番地のことを分散番地と言う。また生成されたひとつの番地からアクセスされるレコード格

† An Analysis of Search Algorithm of Linear Probing Method by RYOZO NAKAMURA (Department of Electrical Engineering and Computer Science, Faculty of Engineering, Kumamoto University) and YOICHI OSHIMA (Basic Engineering, Faculty of Engineering, Kumamoto University).

** 熊本大学工学部電気情報工学科

††† 熊本大学工学部共通講座数学教室

納領域の一分割単位をバケット (bucket) と呼ぶ。バケットサイズとは、ひとつのバケットに格納可能なレコードの総数を言う。

ところで、見出しを分散番地に変換する際、異なった見出しが分散表の上で同じ番地になる現象すなわち衝突が起こる。この衝突を解決する方法のひとつとして番地列生成法がある。線形法は衝突処理として番地列生成法を用いる。

線形法においては、見出しの値 K をもつレコードの登録は次のような手順で行う。はじめに、見出しの値 K は分散関数 $\phi(K)$ によって、大きさ M の分散表の 0 から $M-1$ までの分散番地のいずれかの番地に置き換えられる。次に、その番地のバケットに空きがあれば、そのバケットに見出しを含むレコードを登録する。しかし、そのバケットが満状態すなわち空きがなければ、次の登録候補番地として、 $\phi(K)-1$ 番地のバケットを調べ、そのバケットがまた満状態であれば、さらに $\phi(K)-2$ 番地と順次に空きがあるバケットを探し、そのバケットに見出しすなわちレコードを登録する。このとき、分散表の始端すなわち 0 番地のバケットの次は、終端の $M-1$ 番地のバケットへと巡回して登録場所を探すものとする。

一方、前述のようにして登録された見出しの探索では、登録されたときと同じ探索路をたどりながらバケット単位にアクセスする。

3. 従来のアルゴリズムの解析

まず、表の大きさが M で、バケットサイズが b の分散表に、 n 個の見出しを一様に分配するとする。したがって、 n 個の見出しからなる番地列の取りうる値 M^n 通りがすべて等確率となる。ここで、前述のアルゴリズムに従い、任意の番地列を分散表に格納したとき、 0 番地のバケットに空きがあるような番地列の数を表す関数を $f_b(M, n)$ で表す。また、分散表の 0 番地のバケットに空きがあり、 1 番地から h 番地までのバケットがすべて見出しで占有され、 $h+1$ 番地のバケットに空きがあるような番地列の数を表す関数を $g_b(M, n, h)$ で表す。このとき、任意の番地として 0 番地を指定しているが、前記の挿入アルゴリズムから、いずれの番地を指定しても関数 f_b, g_b の値は変わらない。

ところで、従来の解析では、バケットサイズを 1 と固定して、 $f_1(M, n)$ を次のように表している²⁾。

$$f_1(M, n) = M^n \left(1 - \frac{n}{M}\right), \quad (M \geq n > 0) \quad (1)$$

また、関数 f_1 と g_1 の間にも次のような簡潔な関係が成立する²⁾。

$$g_1(M, n, h) = \binom{n}{h} f_1(h+1, h) f_1(M-h-1, n-h) \quad (M \geq n \geq h \geq 0) \quad (2)$$

次に、前述のアルゴリズムに従い、 n 個の見出しが任意の順序で一様に登録されたあと、 $n+1$ 番地の見出しを登録する際、 $k+1$ 個のバケットを探して格納される確率を $P_{n,k}$ とすれば、 $P_{n,k}$ は上記の関数 g_1 を用いて次のように表される⁴⁾。

$$P_{n,k} = \frac{1}{M^n} \{g_1(M, n, k) + g_1(M, n, k+1) + \dots + g_1(M, n, n)\} \quad (M \geq n \geq k \geq 0) \quad (3)$$

たとえば、 $M=4, b=1, n=2$ の場合について、数値例を用いて考察する。このとき、すべての番地列は M^n すなわち 4^2 通りある。この中で、特定のバケットを 0 番地のバケットとすると、 0 番地のバケットに空きがあり、 1 番地のバケットが満状態で、 2 番地のバケットに空きがあるような番地列の数を表す関数は $g_1(4, 2, 1)$ となり、それは次のような値になる。

$$g_1(4, 2, 1) = \binom{2}{1} f_1(2, 1) f_1(2, 1) = 2$$

また、同様に $g_1(4, 2, 0), g_1(4, 2, 2)$ はそれぞれ次のような値になる。

$$g_1(4, 2, 0) = \binom{2}{0} f_1(1, 0) f_1(3, 2) = 3$$

$$g_1(4, 2, 2) = \binom{2}{2} f_1(3, 2) f_1(1, 0) = 3$$

次に、 3 番目の見出しを登録するとき、バケットを 1 回、 2 回および 3 回探して格納される各確率 $P_{20}, P_{21},$ および P_{22} はそれぞれ次のようになる。

$$P_{20} = \frac{1}{4^2} \{g_1(4, 2, 0) + g_1(4, 2, 1) + g_1(4, 2, 2)\} = \frac{8}{4^2}$$

$$P_{21} = \frac{1}{4^2} \{g_1(4, 2, 1) + g_1(4, 2, 2)\} = \frac{5}{4^2}$$

$$P_{22} = \frac{1}{4^2} \{g_1(4, 2, 2)\} \\ = \frac{3}{4^2}$$

一方、探索するときには、登録時と同じ探索路をたどるので、 k 回の探索すなわち k 個のバケットを探して見つかる見出しは、登録時の順序が k 番目以降の見出しのうち、 k 個のバケットを探して挿入された見出しである。したがって、 k 回の探索で見つかる見出しの確率は、各見出しの探索頻度を考慮に入れ、次のように表される⁴⁾。

$$\rho_k P_{k-1, k-1} + \rho_{k+1} P_{k, k-1} + \dots + \rho_n P_{n-1, k-1} \\ (k=1, 2, \dots, n) \quad (4)$$

ただし、見出しの探索頻度は登録順に ρ_i ($i=1, 2, \dots, n$) とする。

したがって、成功時の探索路長すなわち探索コストの平均 S_n および分散 V_n は次のように導出できる⁴⁾。

$$S_n = \sum_{k=0}^{n-1} (k+1) \sum_{j=k}^{n-1} \rho_{j+1} P_{j,k} \\ = \sum_{k=0}^{n-1} \rho_{k+1} \sum_{j=0}^k (j+1) P_{k,j} \quad (5)$$

$$V_n = \sum_{k=0}^{n-1} (k+1)^2 \sum_{j=k}^{n-1} \rho_{j+1} P_{j,k} - S_n^2 \\ = \sum_{k=0}^{n-1} \rho_{k+1} \sum_{j=0}^k (j+1)^2 P_{k,j} - S_n^2 \quad (6)$$

また、(5)、(6)の具体的な考察は文献 4) を参照されたい。

4. 提示するアルゴリズムの解析

前述したように、バケットサイズを1と固定した従来の解析では、関数 f_1 と関数 g_1 とともに簡潔に導出され、挿入・探索アルゴリズムの厳密な解析がなされている。しかし、バケットサイズをパラメータとして一般化したもとは、関数 f_0 と g_0 とともに定かでない、よって、そのアルゴリズムの解析も明らかでない。

ここで提示する解析では、まず、バケットサイズを考慮したモデルのもとで、関数 f_0 と関数 g_0 とを導き出し、挿入・探索アルゴリズムの厳密な解析を提示する。次に、関数 f_0 について考察する。

4.1 解析

はじめに、ある番地のバケットに空きがあるような番地列の数を表す関数 $f_0(M, n)$ を導出する。このとき、ある番地のバケットを0番地のバケットにとっても、前述のアルゴリズムから、一般性は失われない。よって、この解析では、0番地のバケットに少なくと

もひとつ以上の空きがあるように、0番地から i 番地までのバケットに格納できる見出しの数を k_i とする。すなわち、0番地のバケットに格納される見出しの数を k_0 、0番地から1番地までのバケットに格納される見出しの数を k_1 とする。したがって、 i 番地のバケットに格納される見出しの数は $k_i - k_{i-1}$ 個となる。

ここで、バケットサイズが b の M 個のバケットに、 n 個の見出しを分配する方法の数を考察する。

はじめに、0番地のバケットには、 k_0 個の見出しが $\binom{n}{k_0}$ 通りで分配される。このとき、前提条件すなわち0番地のバケットにはひとつ以上の空きがあるという条件から、 k_0 の値は0から $b-1$ 個の範囲となる。したがって、0番地のバケットに分配可能な数は $\sum_{k_0=0}^{b-1} \binom{n}{k_0}$ となる。次に、1番地のバケットには、0番地のバケットに分配された残りの $n - k_0$ 個の見出しから、 $k_1 - k_0$ 個が選ばれ分配される。このとき、上記の条件から k_1 の値は k_0 から $2b-1$ 個となり、1番地のバケットに分配可能な数は $\sum_{k_1=k_0}^{2b-1} \binom{n-k_0}{k_1-k_0}$ となる。以下同様なことを $\lambda-1$ 番地まで続ける。

ここで、0番地にひとつの空きを作るように格納するとき、0番地から $\lambda-1$ 番地までのバケットに分配される見出しの数は $k_{\lambda-1}$ だから、0番地から $\lambda-1$ 番地までには $b\lambda-1-k_{\lambda-1}$ 個の空きがある。したがって、 λ 番地以後のバケットに残りの $n-k_{\lambda-1}$ 個の見出しを勝手に分配したとしても、あふれるのは高々 $n-k_{\lambda-1}-b$ 個で、これは $b\lambda-1-k_{\lambda-1}$ 個以下であるから、0番地にはひとつの空きが保障される。よって、 λ 番地以後のバケットには $n-k_{\lambda-1}$ 個を勝手に分配してもよく、その分配方法の数は $(M-\lambda)^{n-k_{\lambda-1}}$ である。

上記から、0番地のバケットにひとつ以上の空きがあるような番地列の数を表す関数 $f_0(M, n)$ は次のように定式化できる。

$$f_0(M, n) \\ = \sum_{k_0=0}^{b-1} \sum_{k_1=k_0}^{2b-1} \dots \sum_{k_{\lambda-1}=k_{\lambda-2}}^{\lambda b-1} \binom{n}{k_0} \binom{n-k_0}{k_1-k_0} \dots * \\ * \binom{n-k_{\lambda-2}}{k_{\lambda-1}-k_{\lambda-2}} (M-\lambda)^{n-k_{\lambda-1}} \\ = \sum \dots \sum \frac{n! (M-\lambda)^{n-k_{\lambda-1}}}{k_0! (k_1-k_0)! \dots (k_{\lambda-1}-k_{\lambda-2})! (n-k_{\lambda-1})!} \quad (7)$$

ただし、 $\lambda = \lceil n/b \rceil$

* $\lambda = \lceil n/b \rceil, \lceil x \rceil$ は x 以上の最小の整数を表す。

ここで導入した k_i は、仮にバケットサイズ b が無限大であれば、あるバケットに分配される見出しの個数となる。

たとえば、 $M=3$, $b=2$, $n=3$ の場合について、数値例を用いて考察する。

このとき、すべての番地列の数は、 $M^n=3^3=27$ 個あるが、そのうち 0 番地のバケットにひとつ以上の空きがある番地列の数は、下に示す下線印をつけた 20 個の番地列となる。

000, 001, 002, 010, 011, 012, 020, 021, 022,
100, 101, 102, 110, 111, 112, 120, 121, 122,
200, 201, 202, 210, 211, 212, 220, 221, 222

上記の番地列の個数は (7) から次のようになる。

$$f_2(3, 3) = \sum_{k_0=0}^1 \sum_{k_1=k_0}^3 \binom{3}{k_0} \binom{3-k_0}{k_1-k_0} (3-2)^{3-k_1} \\ = 20$$

次に、0 番地のバケットに空きがあり、1 番地から h 番地までのバケットがすべて見出しで占有され、 $h+1$ 番地のバケットに空きがあるような番地列の数を表す関数 $g_b(M, n, h)$ を考察する。

従来の解析のように、バケットサイズを 1 と固定すれば、関数 g_1 は簡潔に (2) のように表すことができる。しかし、バケットサイズが b であるような一般的な場合には、あるバケットに空きがある状態は、ひとつの空きがある場合から b 個の空きがある場合までを考慮する必要がある。

したがって、はじめに、0 番地のバケットに空きがあり、1 番地から $h-1$ 番地までのバケットがすべて見出しで占有され、 h 番地のバケットに空きがある場合を考察する。このとき、0 番地から $h-1$ 番地までのバケットに格納可能な見出しの総数を w とすると、 w は、 $(h-1)b$ 個から $hb-1$ 個までの数となる。また、 k 番地以後のバケットすなわち $k, k+1, \dots, h-1$ 番地のバケットに分配される見出しの個数を i_k で表す。

前記から、1 番地から $h-1$ 番地までのすべてのバケットが、見出しで一杯になるためには、まず、 $h-1$ 番地のバケットには h 番地のバケットから見出しがあふれて来ないという仮定から、このバケットに分配される見出しの個数 i_{h-1} の値は b から w 個までの範囲となる。よって、 $h-1$ 番地のバケットに分配可能な方法の数は $\sum_{i_{h-1}=b}^w \binom{w}{i_{h-1}}$ となる。次に、 $h-2$ 番地のバケットには、 $h-1$ 番地のバケットに分配された残りの $(w-i_{h-1})$ 個の見出しから、 $(i_{h-2}-i_{h-1})$ 個が選

ばれ分配される。このとき、 $i_{h-1}-b$ 個の見出しが $h-1$ 番地のバケットよりあふれて来るため、 $h-2$ 番地に分配される数 $(i_{h-2}-i_{h-1})$ は $b-(i_{h-1}-b)$ から $w-i_{h-1}$ までの範囲すなわち i_{h-2} の値は $2b$ から w 個までの範囲となる。よって、 $h-2$ 番地以後のバケットすなわち $h-2$ 番地と $h-1$ 番地とのバケットに分配可能な方法の数は $\sum_{i_{h-2}=2b}^w \sum_{i_{h-1}=b}^w \binom{w-i_{h-1}}{i_{h-2}-i_{h-1}} \binom{w}{i_{h-1}}$ 通りとなる。以下同様なことを 1 番地のバケットまで続けられよい。

その結果、0 番地のバケットに空きがあり、1 番地から $h-1$ 番地までのバケットがすべて見出しで占有される番地列の数を表す関数を $t_b(h, w)$ とすると、関数 t_b は次のように表すことができる。

$$t_b(h, w) \\ = \sum_{i_1=(h-1)b}^w \sum_{i_2=(h-2)b}^w \dots \sum_{i_{h-1}=b}^w \binom{w-i_2}{i_1-i_2} \binom{w-i_3}{i_2-i_3} \dots \\ * \dots \binom{w-i_{h-1}}{i_{h-2}-i_{h-1}} \binom{w}{i_{h-1}} \\ = \sum \dots \sum \frac{w!}{(w-i_1)!(i_1-i_2)! \dots (i_{h-2}-i_{h-1})! i_{h-1}!} \\ (h > 1, hb-1 \geq w \geq (h-1)b) \quad (8)$$

ただし、 $h=1$ のときには、 $t_b(1, w)=1$, ($0 \leq w \leq b-1$) となる。

たとえば、バケットサイズが 2 の表に、4 個の見出しを挿入するとき、0 番地のバケットに空きがあり、1 番地から 2 番地までのバケットが見出しでいっぱいになる番地列の数は $t_2(3, 4)$ で表され、その関数の値は次のようになる。

$$t_2(3, 4) = \sum_{i_1=4}^4 \sum_{i_2=2}^4 \binom{4-i_2}{i_1-i_2} \binom{4}{i_2} \\ = 11$$

すなわち、それらの番地列は次のようになる。

1122, 1212, 1221, 1222, 2112, 2121,
2122, 2211, 2212, 2221, 2222

前述の関数 f_b では、1 番地と 2 番地のバケットがすべて見出しで占有される必要はなく、ただ、0 番地のバケットに空きがあればよいので、 $f_2(3, 4)=43$ のように番地列の数は多くなる。

ところで、前記の関数 g_b すなわち分散表の 0 番地のバケットに空きがあり、1 番地から h 番地までのバケットがすべて見出しで占有され、 $h+1$ 番地のバケットに空きがあるような番地列の数を表す関数は、上述した関数 f_b と t_b によって、次のように表すことができる。

$$\begin{aligned}
 q_b(M, n, h) &= \sum_{i=0}^{b-1} \binom{n}{hb+i} t_b(h+1, hb+i) \\
 &\quad \times f_b(M-h-1, n-hb-i) \\
 &\quad (M \geq n \geq h \geq 0) \quad (9)
 \end{aligned}$$

次に、 $n+1$ 番目の見出しを登録するとき、バケツトサイズ b のバケツトを $k+1$ 回探して挿入する場合の確率 P_{nk} は、(3) に準じて次のように表現することができる。

$$\begin{aligned}
 P_{nk} &= \frac{1}{M^n} (q_b(M, n, k) \\
 &\quad + q_b(M, n, k+1) + \dots + q_b(M, n, n)) \quad (10)
 \end{aligned}$$

たとえば、分散表の大きさ M が 3、バケツトサイズ b が 2 の表に、 a_1, a_2, a_3 からなる 3 個の見出しを $a_1 a_2 a_3$ の順序で登録するとき、すべての配置可能な場合を表示すると表 1 のようになる。表 1 で左端の列は場合の数の通し番号を示している。次の列は、 a_1, a_2, a_3 に対する分散関数の値すなわち見出し番地列を表す。そして、次の 3 列は各バケツトに配置された見出しを示している。また、右端の列は、配置された場合に対応する関数 q_2 を示している。

表 1 から明らかなように、0 番地のバケツトに空きがあり、また 1 番地のバケツトにも空きがあるような番地列の数 $q_2(3, 3, 0)$ は次のようになる。

$$\begin{aligned}
 q_2(3, 3, 0) &= \sum_{i=0}^1 \binom{3}{i} t_2(1, i) f_2(2, 3-i) \\
 &= \binom{3}{0} t_2(1, 0) f_2(2, 3) + \binom{3}{1} t_2(1, 1) f_2(2, 2) \\
 &= 13
 \end{aligned}$$

次に、0 番地のバケツトに空きがあり、1 番地のバケツトが見出しでいっぱいになり、2 番地のバケツトに空きがあるような番地列の数 $q_2(3, 3, 1)$ は次のようになる。

$$\begin{aligned}
 q_2(3, 3, 1) &= \sum_{i=0}^1 \binom{3}{2+i} t_2(2, 2+i) f_2(1, 1-i) \\
 &= \binom{3}{2} t_2(2, 2) f_2(1, 1) + \binom{3}{3} t_2(2, 3) f_2(1, 0) \\
 &= 7
 \end{aligned}$$

また、0 番地のバケツトに空きがあり、1 番地と 2 番地とのバケツトが見出しでいっぱいになり、3 番地のバケツトに空きがある番地列の数 $q_2(3, 3, 2)$ は 0 となる。また、同様にして $q_2(3, 3, 3)$ も 0 になる。

したがって、4 番目の見出しを登録するとき、バケツトを 1 回だけ探して登録される確率 P_{30} 、バケツトを 2 回探して登録される確率 P_{31} 、バケツトを 3 回探して登録される確率 P_{32} はそれぞれ次のようになる。

表 1 登録順序列 $a_1 a_2 a_3$ の 3 個の見出しを分散表 ($M=3, b=2$) へ格納した状態

Table 1 Feature that 3 keys of insertion order $a_1 a_2 a_3$ are mapped into the hash table ($M=3, b=2$).

No.	見出し番地列	0 番地	1 番地	2 番地	$q_2(3, 3, h)$
1	000	$a_1 a_2$		a_3	
2	111	a_2	$a_1 a_2$		$q_2(3, 3, 1)$
3	222		a_2	$a_1 a_2$	$q_2(3, 3, 0)$
4	001	$a_1 a_2$	a_2		
5	002	$a_1 a_2$		a_2	
6	010	$a_1 a_2$	a_2		
7	020	$a_1 a_2$		a_2	
8	100	$a_2 a_2$	a_1		
9	200	$a_2 a_2$		a_1	
10	110	a_2	$a_1 a_2$		$q_2(3, 3, 1)$
11	112		$a_1 a_2$	a_2	$q_2(3, 3, 1)$
12	101	a_2	$a_1 a_2$		$q_2(3, 3, 1)$
13	121		$a_1 a_2$	a_2	$q_2(3, 3, 1)$
14	011	a_1	$a_2 a_2$		$q_2(3, 3, 1)$
15	211		$a_2 a_2$	a_1	$q_2(3, 3, 1)$
16	220	a_2		$a_1 a_2$	$q_2(3, 3, 0)$
17	221		a_2	$a_1 a_2$	$q_2(3, 3, 0)$
18	202	a_2		$a_1 a_2$	$q_2(3, 3, 0)$
19	212		a_2	$a_1 a_2$	$q_2(3, 3, 0)$
20	022	a_1		$a_2 a_2$	$q_2(3, 3, 0)$
21	122		a_1	$a_2 a_2$	$q_2(3, 3, 0)$
22	012	a_1	a_2	a_2	$q_2(3, 3, 0)$
23	021	a_1	a_2	a_2	$q_2(3, 3, 0)$
24	102	a_2	a_1	a_2	$q_2(3, 3, 0)$
25	201	a_2	a_2	a_1	$q_2(3, 3, 0)$
26	120	a_2	a_1	a_2	$q_2(3, 3, 0)$
27	210	a_2	a_2	a_1	$q_2(3, 3, 0)$

$$\begin{aligned}
 P_{30} &= \frac{1}{3^3} (q_2(3, 3, 0) + q_2(3, 3, 1) \\
 &\quad + q_2(3, 3, 2) + q_2(3, 3, 3)) \\
 &= \frac{20}{3^3}
 \end{aligned}$$

$$\begin{aligned}
 P_{31} &= \frac{1}{3^3} (q_2(3, 3, 1) + q_2(3, 3, 2) + q_2(3, 3, 3)) \\
 &= \frac{7}{3^3}
 \end{aligned}$$

$$P_{32} = 0$$

一方、見出しを探索する場合には、前述の論議から、登録時と同じ探索路をたどるので、 k 回の探索で見つかる見出しは、登録時の順序が $(k-1)b+1$ 番目以降の見出しで、かつ、 k 個のバケツトを探して挿入された見出しである。よって、 K 回の探索すなわちアクセスで見つかる確率は、各見出しの探索頻度を考慮に入れ、次のようになる。

$$\begin{aligned} & \rho_{(k-1)b+1}P_{(k-1)b, k-1} + \rho_{(k-1)b+2}P_{(k-1)b+1, k-1} \\ & + \dots + \rho_n P_{n-1, k-1} \\ & (k=1, 2, \dots, \lceil n/b \rceil) \end{aligned} \quad (11)$$

したがって、バケットサイズと見出しの探索頻度とを考慮に入れた探索コストすなわちアクセス回数の平均と分散は次のようになる。

$$S_n = \sum_{k=0}^{\lambda-1} (k+1) \sum_{j=kb}^{n-1} \rho_{j+1} P_{j,k} \quad (12)$$

$$V_n = \sum_{k=0}^{\lambda-1} (k+1)^2 \sum_{j=kb}^{n-1} \rho_{j+1} P_{j,k} - S_n^2 \quad (13)$$

ただし、 $\lambda = \lceil n/b \rceil$

上記から、見出しの探索頻度とバケットサイズとを考慮して一般化したモデルにおける探索コストの厳密な評価式が導出できた。

4.2 関数 $f_b(M, n)$ に対する考察

分散表の大きさが M で、バケットサイズが b の表に、 n 個の見出しを登録するとき、0番地のバケットに空きがあるような番地列の数を評価する関数 $f_b(M, n)$ を(7)で導出した。ここでは、関数 $f_b(M, n)$ の性質を考察し議論する。

はじめに、見出しの数 n をバケットサイズ b の整数倍、すなわち $n = pb$, ($p=1, 2, 3, \dots$) とする。そして、分散表の大きさを変数 x とする。このとき、(7)は次のように表される。

$$\begin{aligned} f_b(x, pb) &= \sum_{k_0=0}^{b-1} \dots \sum_{k_{p-1}=k_{p-2}}^{pb-1} \\ & \frac{(pb)!(x-p)^{pb-k_{p-1}}}{k_0! \dots (k_{p-1}-k_{p-2})!(pb-k_{p-1})!} \end{aligned} \quad (14)$$

ここで、(14)を変数 x について b 回微分した b 階導関数を $f_b^{(b)}(x, pb)$ と表すと、それは次のようになる。

$$\begin{aligned} f_b^{(b)}(x, pb) &= pb(pb-1) \dots ((p-1)b+1) \sum \dots * \\ & * \sum \frac{((p-1)b)!(x-p)^{(p-1)b-k_{p-1}}}{k_0!(k_1-k_0)! \dots ((p-1)b-k_{p-1})!} \end{aligned}$$

このとき、 k_{p-1} の上限は $(p-1)b$ となるので、 k_{p-1} についての和を考えると次のようになる。

$$\begin{aligned} & \sum_{k_{p-1}=k_{p-2}}^{(p-1)b} \frac{(x-p)^{(p-1)b-k_{p-1}}}{(k_{p-1}-k_{p-2})!((p-1)b-k_{p-1})!} \\ & = \sum_{i=0}^{(p-1)b-k_{p-2}} \frac{(x-p)^{(p-1)b-k_{p-2}-i}}{i!((p-1)b-k_{p-2}-i)!} \\ & = \frac{1}{((p-1)b-k_{p-2})!} \sum_{i=0}^{(p-1)b-k_{p-2}} \binom{(p-1)-k_{p-2}}{i} * \\ & * (x-p)^{(p-1)b-k_{p-2}-i} \end{aligned}$$

$$= \frac{1}{((p-1)b-k_{p-2})!} (x-p+1)^{(p-1)b-k_{p-2}}$$

よって、

$$\begin{aligned} f_b^{(b)}(x, pb) &= pb(pb-1) \dots ((p-1)b+1) \sum_{k_0=0}^{b-1} \dots * \\ & * \sum_{k_{p-2}=k_{p-3}}^{(p-1)b-1} \frac{((p-1)b)!(x-(p-1))^{(p-1)b-k_{p-2}}}{k_0!(k_1-k_0)! \dots ((p-1)b-k_{p-2})!} \end{aligned}$$

すなわち、 $f_b^{(b)}(x, pb)$ は次のようになる。

$$\begin{aligned} f_b^{(b)}(x, pb) &= pb(pb-1) \dots ((p-1)b \\ & + 1) f_b(x, (p-1)b) \end{aligned} \quad (15)$$

すなわち、関数 f_b の b 階導関数 $f_b^{(b)}(x, pb)$ は、帰納的に $f_b(x, (p-1)b)$ で表すことができる。

ここで、(15)を b 重積分すると、次のように表現される。

$$\begin{aligned} f_b(M, pb) &= pb(pb-1) \dots ((p-1)b+1) * \\ & * \int_{c_0}^M \int_{c_0-1}^{x_{b-1}} \dots \int_{c_1}^{x_1} f_b(x, (p-1)b) dx \dots dx_{b-1} \end{aligned} \quad (16)$$

このとき、 c_0, c_0-1, \dots, c_1 は、それぞれ次の条件を満たす。

$$\left. \begin{aligned} f_b(c_0, pb) &= 0 \\ f_b'(c_0-1, pb) &= 0 \\ & \vdots \\ f_b^{(b-1)}(c_1, pb) &= 0 \end{aligned} \right\} \quad (17)$$

すなわち、 $f_b(c_0, pb) = 0$ になるためには、(14)から明らかに $c_0 = p$ となる。次に、 $f_b'(c_0-1, pb) = 0$ となるためには、まず、(14)を微分すると、次のようになる。

$$f_b'(x, pb) = \sum \sum \frac{(pb)!(x-(p-1))^{pb-1-k_{p-2}}}{k_0! \dots (k_{p-2}-k_{p-3})!(pb-1-k_{p-2})!} \quad (18)$$

したがって、 $f_b'(c_0-1, pb) = 0$ になるためには、 $c_0-1 = p-1$ となる。

次に、 $f_b''(x, pb)$ は(18)を微分して次のようになる。

$$\begin{aligned} f_b''(x, pb) &= \sum \sum \frac{(pb)!(x-(p-1))^{pb-2-k_{p-3}}}{k_0! \dots (k_{p-2}-k_{p-3})!(pb-2-k_{p-2})!} \end{aligned}$$

このとき、 $f_b''(c_0-2, pb) = 0$ になるためには、 $c_0-2 = p-1$ となる。

同様に、(17)を満たす c_0-3, \dots, c_1 の値は、すべて $p-1$ となる。

次に、(16)の初期値となる関数 $f_b(x, b)$ は、(14)か

ら次のようになる.

$$\begin{aligned} f_b(x, b) &= \sum_{k_0=0}^{b-1} \binom{b}{k_0} (x-1)^{b-k_0} \\ &= x^b - 1 \end{aligned} \quad (19)$$

上記の考察から, 分散表の大きさが M , バケットサイズが b の表に, pb 個の見出しを挿入するとき, あるバケットにひとつ以上の空きがある番地列の数は, 次のような順序で導出することができる.

$$\begin{aligned} f_b(M, 2b) &= 2b(2b-1)\cdots(b+1)* \\ &\quad * \int_2^M \int_1^{x_{b-1}} \cdots \int_1^{x_1} (x^b-1) dx \cdots dx_{b-1} \end{aligned}$$

$$\begin{aligned} f_b(M, 3b) &= 3b(3b-1)\cdots(2b+1)* \\ &\quad * \int_3^M \int_2^{x_{b-1}} \cdots \int_2^{x_1} f_b(x, 2b) dx \cdots dx_{b-1} \end{aligned}$$

すなわち, 次のような一般的な関係に従って順次導出することができる.

$$\begin{aligned} f_b(M, pb) &= pb(pb-1)\cdots((p-1)b+1)* \\ &\quad * \int_p^M \int_{p-1}^{x_{b-1}} \cdots \int_{p-1}^{x_1} f_b(x, (p-1)b) dx \cdots dx_{b-1} \end{aligned} \quad (20)$$

たとえば, 従来の解析のように, バケットサイズ b を 1 と仮定すれば, 上記の導出過程は次のようになる.

$$\begin{aligned} f_1(M, 2) &= 2 \int_2^M (x-1) dx \\ &= 2[x^2/2 - x]_2^M \\ &= M^2 - 2M \end{aligned}$$

$$\begin{aligned} f_1(M, 3) &= 3 \int_3^M (x^2 - 2x) dx \\ &= 3[x^3/3 - x^2]_3^M \\ &= M^3 - 3M^2 \end{aligned}$$

$$\begin{aligned} f_1(M, 4) &= 4 \int_4^M (x^3 - 3x^2) dx \\ &= 4[x^4/4 - x^3]_4^M \\ &= M^4 - 4M^3 \end{aligned}$$

したがって, 帰納的に次の関係が成立する.

$$f_1(M, n) = M^n - nM^{n-1}$$

すなわち, 上述のように, バケットサイズ b を 1 と仮定すれば, (19)の初期値と(20)の導出関係とを用いて前記の(1)が簡単に導出される.

ところで, バケットサイズを任意の大きさにしたとき, (20)を用いた導出手続きによって, 関数 f_b は理論的に計算可能であるが, この関数をより簡潔に表現するため今後の検討を要する.

5. む す び

線形法を用いた見出し探索法において, 見出しの探索頻度のみならずバケットサイズをも考慮に入れて, 一般化したモデルにおける挿入・探索アルゴリズムの厳密な解析を提示した. 特に, 提示した解析では, あるバケットに空きがあるような番地列の数を評価する関数と, ある番地の空きがあるバケットに続く連続した幾つかのバケットがすべて見出しで占有されるような番地列の数を評価する関数とを定式化して考察した.

提示した解析から導出された評価式によって, はじめてバケットサイズと見出しの探索頻度とを考慮して一般化したモデルでの厳密な探索コストの評価を行うことができる.

謝辞 日頃から, 御助言, 御指導を賜っている松山公一熊本大学名誉教授ならびに九州大学工学部 牛島和夫教授に感謝いたします. また, 懇切丁寧な御指摘をいただいた査読者の方に感謝いたします.

参 考 文 献

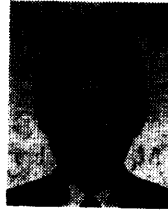
- 1) Schay, G. and Spruth, W.: Analysis of File Addressing Method, *Comm. ACM*, Vol. 5, No. 8, pp. 459-462 (1962).
- 2) Knuth, D.E.: *The Art of Computer Programming, Vol. 3, Sorting and Searching*, pp. 534-538, Addison-Wesley, Reading, Mass. (1973).
- 3) Horowitz, E. and Sahni, S.: *Fundamental of Computer Algorithms*, pp. 82-92, Computer Science Press Inc., Md. (1978).
- 4) 中村, 松山: 見出しの探索頻度を考慮した探索路長の考察, *情報処理学会論文誌*, Vol. 24, No. 4, pp. 505-512 (1983).
- 5) 中村, 松山: 線形法におけるバケット方式を用いたときのアクセス回数について, *情報処理学会論文誌*, Vol. 26, No. 1, pp. 46-52 (1985).

(昭和 62 年 8 月 6 日受付)

(昭和 63 年 4 月 14 日採録)

**中村 良三 (正会員)**

昭和 15 年生。昭和 39 年防衛大学
応用物理専攻卒業。昭和 43 年熊本
大学大学院電気工学専攻修士課程修
了。中部電力(株)を経て、昭和 50
年から熊本大学工学部勤務。現在、
同工学部電気情報工学科助教授。工学博士。算法解
析、計算機言語、推論処理等に興味を持っている。電
気学会、電子情報通信学会各会員。

**大島 洋一**

昭和 40 年熊本大学理学部数学科
卒業。昭和 42 年九州大学大学院理
学研究科修士課程修了。昭和 42 年
6 月同大学院博士課程退学。同年 7
月大阪市立大学理学部助手。昭和
45 年熊本大学工学部講師。現在同工学部共通講座数学
教室教授。理学博士。研究分野は、確率過程、特にマ
ルコフ過程のポテンシャル論。