

トーナメントの並列比較回数とゲーム木の探索

野下 浩平 佐藤 信弘

電気通信大学 情報工学科

{noshita,sato-n}@igo.cs.uec.ac.jp

Parallel Rounds of Tournaments and Game-Tree Searching

Kohei Noshita and Nobuhiro Sato

Department of Computer Science

University of Electro-Communications

Chofu, Tokyo 182, Japan

Abstract

Let $U(n, t)$ be the minimum number of rounds required to select the t largest of n elements in the worst case on the parallel disjoint comparison model. We show some of the exact values of $U(n, t)$ for small $n \leq 16$, by means of the exhaustive searching of finite game-trees on computers. We prove several new upper and lower bounds of $U(n, t)$. In the proof, we perform the game-tree searching for finding fast algorithms which lead to the desired bounds. By combining these results, we determine the exact values of $U(n, t)$ for infinitely many n .

1 まえがき

昨年 (1998) の CSA コンピュータ将棋選手権において、16 人の競技者のうち上位 4 人を選ぶ組合せの方法が話題になった。その際に、著者の一人は推移律を仮定して 6 回戦で十分であることを示した。本稿の目的はその問題に関する一般的な結果を示すことである。すなわち、与えられた n と t に対して ($1 \leq t \leq \lfloor n/2 \rfloor$, $n \geq 2$)、相異なる n 個の要素からなる全順序集合の中で、最も大きい t 個を選択する並列比較の回数に関する結果を示す。このような選択問題の計算量は、いろいろな計算モデルで調べられてきている [3]。本稿では、2 項比較を並列に実行する並列比較モデル (parallel disjoint comparison) を取りあげる [2, 3]。ここで、1 回の並列比較では同一の要素が 2 つ以上の比較に関与できない。さらに、並列比較の結果を利用して次回の比較の相手を選ぶことができる。1 回の並列比較において、要素の対の集合 $\{a_1 : a_2, a_3 : a_4, \dots, a_{2k-1} : a_{2k}\}$ を選んで並列的に k 個の比較を行なう。ここで、 a_1, a_2, \dots, a_{2k} はすべて互いに異なる要素である ($2k \leq n$)。並列比較の結果、最大で 2^k 個の場合が生じる。

要素数が n の集合の中で大きいものから上位 t 個の要素を選択するのに（最悪の場合に）必要十分な並列比較の回数を $U(n, t)$ で表す。大小の順序を逆に考えると、 $U(n, t) = U(n, n - t)$ であるので、 $1 \leq t \leq \lfloor n/2 \rfloor$ を仮定してよい。

本稿ではまず、小さい n に対してコンピュータによるゲーム木の総当たりの探索を行ない、 $1 \leq n \leq 16$ である n に対して $U(n, t)$ の正確な値を定める。2 節において結果を表で示す。

一般の n に対する結果として、M. Aigner は次の上界を示した [2]。

$$U(n, t) \leq \lceil \log_2 n \rceil + (t - 1).$$

しかし、[2] の取り扱う問題はやや異なり、上位の t 個の要素を整列するものである。一方、本稿の問題では上位の t 個の要素の相対的な順序は問わない。そこで、M. Aigner の上界は、本稿の問題に対する上界でもあるが、本稿の問題に対してはさらに良い上界がありうる。本稿で示す新しい上界は次の通りである。

$$U(n, t) \leq \lceil \log_2 n \rceil + (t - 2), \quad \text{ここで } n \geq 16, t \geq 4 \text{ である.}$$

さらに $t=3$ に対して次の上界を示す。

$$U(n, 3) \leq \lceil \log_2 n \rceil + 1, \quad \text{ここで } 2^{k-1} < n \leq 2^{k-1} + 2^{k-3} \text{ である } (k \geq 4).$$

最後に、これらの結果をあわせて、無限個の n に対して $U(n, t)$ の正確な値を与える式を示す。

2 小さい n に対する正確な値

小さい n に対する $U(n, t)$ の正確な値を表に示す。各々の値を定めるために、コンピュータプログラムによるゲーム木の総当たりの探索を行なった。そこで使った主な技法は、アルファベータ法とハッシュ表による局面表である。計算結果は、著者それぞれが独立に作ったプログラム2つにより確認した。表の中のいくつかの値は、コンピュータによる結果と解析的な結果をあわせることにより求める。なお？印は値が未確定である。

次の2つの補題は基本的なものである。証明は容易であり、他の問題で使われた技法を調整すればよい [1, 3]。

補題 1 $U(n, t) \leq U(n + 1, t), U(n, t) \leq U(n + 1, t + 1)$.

補題 2 $U(\lfloor n/2 \rfloor, t) + 1 \leq U(n, t)$.

表1の各値について説明する。

場合 $t = 1$ の各値は明らかである。4 節の定理 6 をみよ。

場合 $t = 2$ の各値はプログラムの実行結果である（人手でも容易にチェックできる）。4 節の定理 5 をみよ。

場合 $t = 3$ に対しては、 $n \leq 14$ まではコンピュータによる結果である。 $15 \leq n \leq 16$ の結果は、補題 1 による下界と M. Aigner による上記の上界 [2] が一致することより確定する。実は、コンピュータによる結果は $n = 13$ までの値がわかれば十分であり、 $n = 14$ の結果も同様に求めることができる。

場合 $t = 4$ に対しては、 $n \leq 12$ まではコンピュータによる結果である（実は $n = 11$ までの値で十分である）。 $13 \leq n \leq 16$ の結果は、補題 1 による下界と本稿の定理 1

表 1: $U(n, t)$

	1	2	3	4	5	6	7	8
2	1							
3	2							
4	2	2						
5	3	3						
6	3	4	3					
7	3	4	4					
8	3	4	4	4				
9	4	5	5	5				
10	4	5	5	5	5			
11	4	5	5	6	6			
12	4	5	5	6	6	6		
13	4	5	6	6	?	?		
14	4	5	6	6	?	?	?	
15	4	5	6	6	?	?	?	
16	4	5	6	6	?	?	?	?

の上界が一致することより確定する。なお、 $U(16, 4) \leq 6$ とそれを実現するアルゴリズムはすでに CSA メールで示した (1998)。

場合 $t \geq 5$ について、正確な値はコンピュータによる計算結果である。

3 並列比較回数の上界

並列比較によってえられた情報は半順序集合であらわす。 $u(P, t)$ によって半順序集合 P を入力として上位 t 個の要素を選択するために (最悪の場合に) 必要十分な並列比較回数をあらわす。順序のない要素 n 個の集合を P とすると、 $u(P, t) = U(n, t)$ である。次の補題は、並列比較の結果の面倒な場合分けを最悪の場合だけに限定するために有用である。

補題 3 P, Q を 2 つの半順序集合とする。

- (1) P の要素の対に順序を追加する操作により、 Q がえられるとすると、 $u(Q, t) \leq u(P, t)$ が成り立つ。
- (2) P の極小要素 a を最小要素とみなし、 P から a を削除する操作により、 Q がえられるとすると、 $u(Q, t) \leq u(P, t)$ が成り立つ。
- (3) P の極大要素 a を最大要素とみなし、 P から a を削除する操作により、 Q がえられるとすると、 $u(Q, t - 1) \leq u(P, t)$ が成り立つ。

証明は、 P のための最適なアルゴリズムを変換して、 Q のためのアルゴリズムを作るという考え方で容易にできる。他の問題での同様の証明技法、さらにこれに関連する性質については [1, 3] などを参照されたい。

定理 1 $U(n, t) \leq \lceil \log_2 n \rceil + (t - 2)$, ここで $n \geq 16, t \geq 4$ である。

これを証明するために、特別な形をした半順序集合を定義する。半順序集合をサイクルのない有向グラフで表し、推移的關係から導かれる枝を描かないことにする。 $\deg(a)$ によって節点 a から出ていく（下向きの）枝の本数を表す。

定義 半順序集合 P が根をもった有向木であり、かつ次の性質をもつ時、 P を F 木という。任意の節点 $a (\in P)$ に対して、 $\deg(a) \leq 2$ であり、かつ共通の親をもつ任意の 2 つの節点 $a, b (\in P)$ に対して ($a \neq b$)、 $\deg(a) \leq 1$ または $\deg(b) \leq 1$ の少なくとも 1 つが成り立つ。

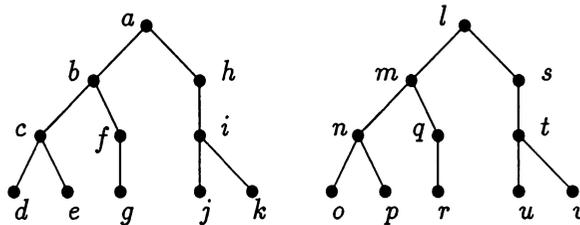


図 1: $\{P_4^m, P_4^m\}$

高さが h ($h \geq 1$) 以下であるような F 木のすべての集合を \mathcal{F}_h であらわす。ここで、木の高さは根から葉までの道の枝数の最大値である。 P_h^m は \mathcal{F}_h の中の最大の F 木であるとする。ここで、最大の F 木とは F 木 ($\in \mathcal{F}_h$) のうち節点数の最大のものである。いいかえれば、 P_h^m は F 木 ($\in \mathcal{F}_h$) であり、その根の一方の子が P_{h-1}^m であり、他方の子の唯一の子が P_{h-2}^m である。図 1 に P_4^m を 2 つ示す。なお、 P_h^m の節点数は $f(h+3) - 2$ である。ここで $f(i)$ は i 番目の Fibonacci 数である。

任意の $P (\in \mathcal{F}_h)$ に対して、補題 3 に示すように、 P_h^m の適当な要素を削除することにより P がえられる。これは h に関する帰納法で証明できる。よって $u(P, h) \leq u(P_h^m, h)$ が成り立つ。

2 つの F 木 $P_1, P_2 (\in \mathcal{F}_h)$ に対して、次のような 2 つの操作からなる並列比較を実行した時、その結果できる半順序集合を $\mu_h(P_1, P_2)$ であらわす。

M1 P_1 と P_2 の根を比較する。

M2 P_1 と P_2 のそれぞれにおいて、親を共通にもつ節点对 a, b のすべてについて比較 $a : b$ を行なう。

上記の並列比較のあと、 $\mu_h(P_1, P_2)$ の中の節点 (葉) のうち、その高さが h をこえるものを削除するものとする。任意の $P_1, P_2 (\in \mathcal{F}_h)$ に対して、 $\mu_h(P_1, P_2)$ はまた F 木 ($\in \mathcal{F}_h$) である。これは h に関する帰納法で証明できる。ここで M2 より、 P_1 と P_2 のそれぞれの根であった節点 r に対して $\deg(r) \leq 1$ になることに注意されたい。

定理 1 の証明 定理の上界を実現するアルゴリズムを構成する。ここで示すアルゴリズムは最悪の場合のみを取り扱うが、それ以外の場合は補題 3 によって最悪の場合に帰着できる。いま、 $n = 2^k$ ($k > 4$) かつ $t \geq 4$ と仮定する。入力は n 個の順序のない要素の集合であり、各要素は F 木 ($\in \mathcal{F}_t$) である。

ステップ 1 (並列比較 $k-1$ 回によって 2 つの F 木を作る。)

2 つの F 木が残るまで、 $j = 1, 2, \dots, k-1$ に関して次を繰り返す。

前回の $(j-1)$ 回目の並列比較でできた F 木の集合を P_1, P_2, \dots, P_m とする。ここで $m = 2^{k-j+1}$ である。 $i = 1, 3, \dots, m-1$ に対して $\mu_i(P_i, P_{i+1})$ を作る。これで 2^{k-j} 個の F 木ができる。

ステップ 2 (並列比較 $t-4$ 回によって上位 $t-4$ 個の要素を選択する)

2 つの F 木はそれぞれ P_j^m であると仮定できる。繰り返しの結果、2 つの F 木のそれぞれが \mathcal{F}_4 に属するようになるまで、 $j = 1, 2, \dots, t-4$ に関して次を繰り返す。

前の回にできる 2 つの F 木で、 \mathcal{F}_{t-j+1} に属すものを P_1 と P_2 とする。これに対して、 $\mu_{t-j+1}(P_1, P_2)$ を作り、その根を削除する (すなわちこの根はすべての要素の中の j 番目に大きいものである)。これによって、 \mathcal{F}_{t-j} の F 木が 2 つできる。

もし $t=4$ ならばこのステップ 2 は省略される。

この F 木 P_1 と P_2 はどれも P_4^m であると仮定できる。残った仕事は、 $\{P_4^m, P_4^m\}$ から上位 4 個の要素を選択することである (図 1)。なぜなら、これまでにすでに全体の上位 $t-4$ 個の要素が選択されているからである。

ステップ 3 (最後の並列比較 3 回によって次の上位 4 個を選択する。)

図 1 で次の 10 個の対を選び、1 回目の並列比較を実行する。

$$\{d : e, j : k, o : p, u : v, c : f, n : q, b : h, m : s, a : t, i : l\}.$$

この並列比較からはじめて、あと高々 2 回で上位 4 個を選択する。すなわち、 $u(\{P_4^m, P_4^m\}, 4) \leq 3$ である。

上のアルゴリズムの正しさは、容易に確かめることができる。ただし、ステップ 3 の $u(\{P_4^m, P_4^m\}, 4) \leq 3$ は自明ではない。この証明は、コンピュータプログラムによるゲーム木の探索によって証明した。なお、このプログラムの探索結果は、面倒であるが人手で解析し、正しさを確認することができる [4]。

並列比較は、ステップ 1 で $k-1$ 回、ステップ 2 で $t-4$ 回、ステップ 3 で 3 回実行するので、全体で $(k-1) + (t-4) + 3 = \log_2 n + (t-2)$ 回となる。なお、 n が 2^k の形でない場合には、 2^k になるまでダミーの要素 (非常に小さい値) をいくつか追加することにより、上記アルゴリズムを適用することができる。 $n = 16$ ($k = 4$) の場合も同様である。

定理 2 $2^{k-1} < n \leq 2^{k-1} + 2^{k-3}$ ($k \geq 4$) に対して, $U(n, 3) \leq \lceil \log_2 n \rceil + 1$ が成り立つ.

証明 証明には $n = 2^{k-1} + 2^{k-3}$ の場合を示せば十分である. 定理 1 のアルゴリズムで, $t = 3$ としてステップ 1 の並列比較を $k - 3$ 回実行する. これで 5 つの P_3^m からなる半順序集合がえられる. ここで, 各々の P_3^m は入力 $n/5 = 2^{k-3}$ 個の順序のない要素から作られる. 前と同様に, 最悪の場合の F 木は P_3^m と仮定してよい. 並列比較の $(k - 2)$ 回目において, これらの 5 つのうち, 4 つの P_3^m を使って, $\mu_3(P_3^m, P_3^m)$ によって 2 つの P_3^m を作る. 残りの 1 つの P_3^m に対して, 2 つの比較を並列実行して, 図 2 の Q を作る. ここで, 2 つの比較とは, 図 2 の P_3^m のように節点に名前がついているとして, $b : e$ と $c : d$ の比較である.

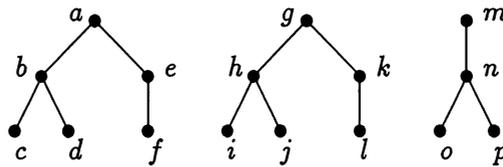


図 2: $\{P_3^m, P_3^m, Q\}$

これで残った仕事は, あと 3 回の並列比較によって, 図 2 の中から上位 3 個の要素を選択することである. 最初の並列比較, すなわち全体の $(k - 1)$ 回目の並列比較では,

$$\{c : d, i : j, o : p, b : e, k : n, a : h, g : m\}.$$

を比較する. この並列比較からはじめて, あと高々 2 回で上位 3 個を選択する. すなわち $u(\{P_3^m, P_3^m, Q\}, 3) \leq 3$ である. この証明は, プログラムによるゲーム木の探索による. なお, その探索の結果から人手によっても正しさを確認することができる [4]. 以上をあわせて, 全並列比較回数は $(k - 2) + 3 = k + 1 = \lceil \log_2 n \rceil + 1$ になる.

4 正確な並列比較回数

この節では, $U(n, t)$ に対する下界の式を求め, 前節の上界の式とあわせて, いくつかの最適の並列比較回数を導く.

定理 3 $10 \cdot 2^k < n \leq 16 \cdot 2^k$ ($k \geq 0$) に対して, $U(n, 4) = \lceil \log_2 n \rceil + 2$ が成り立つ.

証明 2 節の表より, $10 < n \leq 16$ に対して $U(n, 4) = 6$ である. この範囲の n からはじめて, 補題 2 を繰返し適用すれば, 定理の範囲の n に対して, 下界 $U(n, t) \geq \lceil \log_2 n \rceil + 2$ がえられる (詳しくは帰納法). 定理 1 よりこの下界は上界に一致するので, 定理の等式がえられる.

定理 4 $2^{k-1} < n \leq 2^{k-1} + 2^{k-3}$ ($k \geq 4$) に対して, $U(n, 3) = \lceil \log_2 n \rceil + 1$ が成り立つ.
また, $2^{k-1} + 2^{k-2} < n \leq 2^k$ ($k \geq 4$) に対して, $U(n, 3) = \lceil \log_2 n \rceil + 2$ が成り立つ.

証明 2 節の表より, $8 < n \leq 12$ に対して $U(n, 3) = 5$ である. 定理 5 の証明と同様に, 補題 2 を繰返し適用する. $8 \cdot 2^k < n \leq 12 \cdot 2^k$, $k \geq 0$ に対して, $U(n, 3) \geq \lceil \log_2 n \rceil + 1$ がえられる. この下界の n の範囲と定理 2 の上界の n の範囲の共通部分に対して定理の等式がえられる.

後半も同様にできる. 2 節の表より $12 < n \leq 16$ に対して $U(n, 3) = 6$ である. $12 \cdot 2^k < n \leq 16 \cdot 2^k$, $k \geq 0$ に対して, $U(n, 3) \geq \lceil \log_2 n \rceil + 2$ がえられる. 一方, M. Aigner の上界 [2] は, この範囲の n に対してこの下界に一致する.

定理 5 $U(n, 2) = \lceil \log_2 n \rceil + 1$, ここで $n \geq 6$ である.

証明 上界は, 定理 1 のアルゴリズムと同様にして, 容易に示すことができる. 下界は 2 節の表と補題 2 から導くことができる.

次の定理は明らかである.

定理 6 $U(n, 1) = \lceil \log_2 n \rceil$.

5 おわりに

本稿で取り扱った問題は, 上位の t 個の要素を求めるものである. これに関連する選択問題として, 整列した上位の要素を t 個求めるものがある [1, 2]. この並列比較回数を $W(n, t)$ とする. また, 上位から丁度 t 番目の要素を求める問題もある. この並列比較回数を $V(n, t)$ とする. 容易にわかるが,

$$U(n, t) \leq V(n, t) \leq W(n, t)$$

が成り立つ. 今後の課題として, 逐次比較モデルの選択問題 [3] にならって, これら 3 つの値の関係を詳しく調べることもある.

参考文献

- [1] M. Aigner, Producing Posets, *Discrete Mathematics*, 35 (1981), 1–15.
- [2] M. Aigner, Parallel Complexity of Sorting Problems, *Journal of Algorithms*, 3 (1982), 79–88.
- [3] D.E. Knuth, *The Art of Computer Programming*, Vol. 3 (Sorting and Searching), Second Edition, Addison-Wesley, Reading (1998).
- [4] K. Noshita, On the Number of Rounds to Select Some Largest Elements, Typescript (1998), 8 pp.