ある選択問題の並列比較回数について

野 下 浩 平[†] 佐 藤 信 弘[†,☆]

本稿は，与えられた全順序集合において大きい方から上位の要素を指定された個数だけ見つける選択問題をとりあげ，並列比較計算モデルにおける計算量を調べる．$U(n,t)$ は n 個の要素の中から上位の t 個の要素を見つけるための（最悪の場合）必要十分な並列比較（ラウンド）の回数とする．まず小さい n ($n \le 16$) に対する $U(n,t)$ の正確な比較回数を示す．この結果は，ゲーム探索のプログラムにより総当たり探索を行うことによる．次に，新しいアルゴリズムを具体的に示すことにより，一般の n，t に対する $U(n,t)$ の新しい上界式を証明する．最後に，本稿の上界式と下界を表す一般式をあわせることにより，無限個の n に対する $U(n,t)$ の正確な値を決定する．

On the Parallel Rounds of a Certain Selection Problem

Kohei Noshita[†] and Nobuhiro Sato[†,☆]

The complexity of a certain selection problem is investigated on the parallel disjoint comparison model. Let $U(n,t)$ be the minimum number of rounds (i.e., parallel comparisons) required to select the t largest of n elements in the worst case. We show some of the exact values of $U(n,t)$ for small $n \le 16$, by means of the exhaustive searching of finite game-trees on computers. We prove two new upper bounds of $U(n,t)$, by explicitly constructing new selection algorithms. By combining these results with the lower bounds, we determine the exact values of $U(n,t)$ for infinitely many n.

1. はじめに

本稿の目的は，比較演算を並列的に行う計算モデルの上で選択問題の計算量に関する結果を提示することである．ここでとりあげる選択問題は，与えられた n と t に対して，相異なる n 個の要素からなる全順序集合の中で，大きさに関して上位の t 個を選択するものである（t 個の要素の間の大小順序は問わない）．

選択問題の計算量は，逐次および並列の様々な計算モデルで調べられている[1),4)]．本稿の計算モデルは，2項比較の集合を並列に実行する最も基本的な並列比較計算モデルである[3),4)]．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．また，並列比較の結果を利用して（すなわち adaptive に）次回の比較の相手を選ぶことができる．

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）必要十分な並列比較の回数を $U(n,t)$ で表す．本稿では主として $U(n,t)$ を調べるが，これに直接関連する選択問題[2)~4)]として，上位の要素を t 個整列して求めるものがある．この並列比較回数を $W(n,t)$ とする．また，上位からちょうど 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．

逐次比較計算モデルの選択問題では，同様に定義できる U，V，W，特に V，およびそれらの間の関係は詳しく調べられており，その結果が Knuth の本[4)]にまとめられている．本稿ではおおむね完結した説明をするが，読者にはこの本（第2版）の 5.3 節を予備知識として期待する．

並列比較計算モデルでは，$W(n,t)$ が主に Aigner によって調べられている[2),3)]．本稿では $U(n,t)$ に関する新しい結果を示す．定義により，$U(n,t) \le V(n,t) \le W(n,t)$ が成り立つ．本稿も含めてこれまでの研究では，いずれの比較回数も完全に決定できていない．特に，U と W の間の V を調べることは今後の課題である．

本稿では最初に準備的な結果として，小さい n に対する $U(n,t)$ の正確な値を示す．この結果は後に下界の一般式を求めることに利用する．次に，一般の n

† 電気通信大学情報工学科
Department of Computer Science, The University of Electro-Communications
☆ 現在，株式会社アイザック
Presently with ISAC, Inc.

に対する $U(n,t)$ の新しい上界式を導く．Aigner は，一般の n に対する W の上界として次を示した[3)]．

$$W(n,t) \le \lceil \log_2 n \rceil + (t-1).$$

この上界は，上記の W と U の不等式を考慮すると，U に対する上界でもあり，従来の最も良い上界である．本稿で示す U の新しい上界は次のとおりである．

$$U(n,t) \le \lceil \log_2 n \rceil + (t-2),$$

ここで $n \ge 16$，$t \ge 4$ である．

本稿ではさらに $t=3$ に対する新しい次の上界を示す．

$$U(n,3) \le \lceil \log_2 n \rceil + 1,$$

ここで $2^{k-1} < n \le 2^{k-1} + 2^{k-3}$ である（$k \ge 4$）．

本稿で上界式を導くための主なアイデアは，最悪の場合の情報を表すために特別の形をした半順序集合（poset，ポセット）を導入すること，および，ゲーム探索のプログラムを利用して，ポセットから要素を選ぶ手順を求めることである．なお，プログラムによって比較する要素の最良の集合を発見したが，その計算結果をみれば，その正しさは人手によって検証できる．

最後に，一般の n に対する $U(n,t)$ の下界式を求め，上界と下界の一致する部分を調べて，無限個の n に対して $U(n,t)$ の正確な値を決定する．なお，並列選択問題に関するもう１つの代表的な計算モデルである（adaptive でない）比較ネットワーク[4)]との関係についてはあとがきでふれる．

2. 定義と準備的結果

選択の対象にする集合は n 個の要素を持つ全順序集合であり，2 つの要素の比較演算によって大小を知ることができるものとする．本稿の計算モデルは，2 項比較演算の集合を並列に実行する並列比較モデル（parallel disjoint comparison モデル）である[3),4)]．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．すなわち並列比較において要素の対の集合 $\{a_1 : a_2, a_3 : a_4, \cdots, a_{2k-1} : a_{2k}\}$ を選んで並列的に k 個の 2 項比較を行う．ここで，$a_1, a_2, \cdots, a_{2k}$ はすべて互いに異なる要素である（$2k \le n$）．並列比較の結果，最大で 2^k 個の場合が生じる．並列比較の結果を利用して（adaptive に）次回の並列比較における比較の相手を選ぶことができる．ここでの並列比較は，スポーツ競技の用語を用いてラウンド（round）ともよばれる．

さて，$U(n,t)$ によって，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（ラウンド）の回数を表す．同様に，$W(n,t)$ と $V(n,t)$ によって，それぞれ，整列した上位の要素を t 個，上位からちょうど t 番目の要素を求めるラウンド数とする．大小の順序を逆に考えると，$U(n,t) = U(n, n-t)$ である．それで断らないかぎり $1 \le t \le \lfloor n/2 \rfloor$ を仮定してよい．

表 1　小さい n に対する $U(n,t)$
Table 1　$U(n,t)$ for small n.

	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	6	6		
14	4	5	6	6	$\underline{6}$	$\underline{6}$	$\underline{6}$	
15	4	5	6	6	$\underline{6}$	?	?	
16	4	5	6	6	?	?	?	?

例 1　$U(6,3) = 3$，$V(6,3) = 5$，$W(6,3) = 5$．
$U(10,4) = 5$，$V(10,4) = 6$，$W(10,4) = 7$．

次の 2 つの補題は基本的なものである．

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

補題 2　$U(\lceil n/2 \rceil, t) + 1 \le U(n,t)$．

証明　n は偶数であるとする．集合 $\{a_1, a_2, \cdots, a_n\}$ に対して $U(n,t)$ を実現する最良のアルゴリズムは，第 1 ラウンドの比較の集合が $\{a_1 : a_2,\ a_3 : a_4, \cdots, a_{n-1} : a_n\}$ であると仮定できる．いま入力の a_i（i は偶数）に十分小さい値を与えるものとすると，第 2 ラウンド以後は，全要素数 $n/2$ と上位の個数 t に対するアルゴリズムと見なすことができる．このラウンド数は少なくとも $U(n/2,t)$ である．n が奇数の場合も同様に証明できる．

補題 1 の証明も同様である．なお，この証明の方針は他の問題や他の計算モデルで知られている[3),4)]．

次に，小さい n に対する $U(n,t)$ の正確な値を**表 1** に示す[5)]．

表 1 について説明する．場合 $t=1$ の各値は明らかである．4 章の定理 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 章の定理 5 をみよ．場合 $t=3$ に対しては，$n \le 14$ まではコンピュータプログラムによる結果

である．$15 \le n \le 16$ の結果は，補題 1 による下界と Aigner による上記の上界[3)]が一致することより確定する．実は，プログラムによる結果は $n = 13$ までの値が分かれば十分であり，$n = 14$ 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，$n \le 11$ まではプログラムによる結果を利用する．$12 \le n \le 16$ の結果は，補題 1 による下界と本稿の定理 1 の上界が一致することより確定する．なお，$U(16, 4) \le 6$ は筆者の 1 人が本稿以前に示した（あとがき参照）．場合 $t \ge 5$ について，正確な値はプログラムによる計算結果である．なお ? 印は値が未決定である．

ここで，コンピュータプログラムによる計算結果は，ゲーム木の総当たり的探索により求めたものである．そこで使った基本技法は，アルファベータ法とハッシュ表による局面表である．なお，下線のついた値は，非常に時間がかかるので，複数個のコンピュータのクラスタによる並列計算によって求めた[5)]．それ以外の値については，筆者の 2 人がそれぞれ独立に作った逐次プログラムにより確認した．

3. 並列比較回数の上界

並列比較によって得られた情報は半順序集合（poset，ポセット）で表す．ポセットをサイクルのない有向グラフで描く場合，推移的関係から導かれる枝を描かないことにする．枝は上下方向に描き，上の方が下より大きいとする．ポセットを表すグラフにおいて，$\deg(a)$ によって節点 a から出ていく（下向きの）枝の本数を表す．

さて，$u(P, t)$ によってポセット 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表す．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．

例 2 $u(\{a < b < c,\ d < e < f\}, 3) = 1$ である．並列比較として交差的に $\{a : f,\ b : e,\ c : d\}$ を選ぶ．比較 $b : e$ によって上位 2 つと下位 2 つの要素が決まることに注意されたい．

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用いる．この補題の意味は，2 つのポセットに対して，比較で得た大小関係の情報に関して，片方が他方より多いことが分かる場合には，最悪の場合として，（情報の少ない方は多い方よりラウンド数が多いか等しいので）情報の少ない方だけを調べればよいということである．

補題 3 P，Q を 2 つのポセットとする．
(1) P の要素の対に順序を追加する操作により，Q が得られるとすると，$u(Q, t) \le u(P, t)$ が成り立つ．
(2) P の極小要素 a を最小要素と見なし，P から a を削除する操作により，Q が得られるとすると，$u(Q, t) \le u(P, t)$ が成り立つ．
(3) P の極大要素 a を最大要素と見なし，P から a を削除する操作により，Q が得られるとすると，$u(Q, t-1) \le u(P, t)$ が成り立つ．

証明は，上記の意味を考えれば直感的に明らかである．厳密な証明は，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文献 2)，4) を参照されたい．

定理 1 $U(n, t) \le \lceil \log_2 n \rceil + (t - 2)$，
ここで $n \ge 16$，$t \ge 4$ である．

この定理を証明するために，特別な形をした新しいポセットを導入する．その意図は，並列比較のために各要素の比較の相手をみつけるのに，並列比較の結果として現れるいろいろなポセットの中で，（補題 3 によって）最悪の場合のポセットが陽に指摘できるものに限定することである．

定義 ポセット P が根を持つ有向木であり，かつ，次の性質を持つとき，P を F 木という．任意の節点 a (ϵP) に対して，$\deg(a) \le 2$ であり，かつ，共通の親を持つ任意の 2 つの節点 a，b (ϵP) に対して $(a \ne b)$，$\deg(a) \le 1$ または $\deg(b) \le 1$ の少なくとも一方が成り立つ．

例 3 3 つのポセット
$P = \{a\}$，$Q = \{a < b\}$，$R = \{a < b < c,\ d < c\}$ はそれぞれ F 木である．後でもっと大きい典型的な F 木の例を示す．

高さが h 以下であるような F 木のすべての集合を $\mathcal{F}_h$ で表す $(h \ge 1)$．ここで，根からの道上の節点数により節点の高さを定義し，葉の高さの最大値により木の高さを定義する．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 $(\epsilon\ \mathcal{F}_h)$ 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 $(\epsilon\ \mathcal{F}_h)$ 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．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．

例 4 **図 1** に 2 つの F 木 P_4^m を示す $(h = 4)$．

任意の P $(\epsilon\ \mathcal{F}_h)$ は，補題 3 に示すように P_h^m の

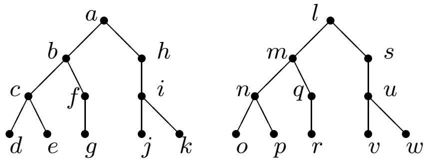


図 1　F 木の対 $\left\{P_4^m, P_4^m\right\}$
Fig. 1　A pair of F-trees $\left\{P_4^m, P_4^m\right\}$.

適当な要素を削除することにより作ることができる（詳しくは h に関する帰納法で証明できる）．よって，$u(P,h) \le u(P_h^m, h)$ が成り立つ．

2 つの F 木 P_1，P_2 ($\epsilon\, \mathcal{F}_h$) 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とき，その結果できるポセットを $\mu_h(P_1, P_2)$ で表す．

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，親を共通に持つ節点対 a，b のすべてについて比較 $a:b$ を行う．

例 5　図 1 において，比較 $a:l$ が **M1** である．比較 $b:h$，$c:f$，$d:e$，$j:k$ は **M2** のうち左の F 木に対するものである（右の F 木も同様）．

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 を超えるものを削除するものとする（後のアルゴリズムでは下位の要素と判定して考慮の対象から除外する）．任意の P_1，P_2 ($\epsilon\, \mathcal{F}_h$) に対して，$\mu_h(P_1, P_2)$ はまた F 木 ($\epsilon\, \mathcal{F}_h$) である．これは明らかであるが，詳しくは 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_1，r_2 に対して $\deg(r_1) \le 1$，$\deg(r_2) \le 1$ になることに注意されたい（**M1** により片方の deg が 2 になる）．

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．ただし，ステップ 3 は別に扱う．いま，$n = 2^k$ ($k > 4$) かつ $t \ge 4$ と仮定する．入力は n 個の順序のない要素の集合であり，それぞれの要素は要素 1 個の F 木 ($\epsilon\, \mathcal{F}_t$) 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，$j = 1, 2, \cdots, k-1$ に関して次を繰返す．
前回の $(j-1)$ 回目の並列比較でできた F 木の集合を $P_1, P_2, \cdots, P_m$ とする．ここで $m = 2^{k-j+1}$ である．$i = 1, 3, \cdots, m-1$ に対して $\mu_t(P_i, P_{i+1})$ 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_t^m であると仮定できる（つまり最悪の場合最も手間のかかるものであると仮定する）．このステップでは，繰返しの結果 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，$j = 1, 2, \cdots, t-4$ に関して次を繰り返す．
前回の繰返しでできる 2 つの F 木で，$\mathcal{F}_{t-j+1}$ に属するものを 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 ($j \le t$)）．これによって，$\mathcal{F}_{t-j}$ の F 木が 2 つできる．

もし $t = 4$ ならばステップ 2 は省略される．
ステップ 2 でできる 2 つの F 木 P_1 と P_2 は,（最悪の場合を考えて）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，ステップ 2 で，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 個の比較を選び，1 回目の並列比較を実行する．

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

この並列比較から始めて，あとたかだか 2 ラウンドで上位 4 個を選択する．すなわち，$u(\{P_4^m, P_4^m\}, 4) \le 3$ 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 $u(\{P_4^m, P_4^m\}, 4) \le 3$ 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る．なお，このプログラムの探索結果は，人手で解析し，正しさを検証することができる．参考のために人手による詳しい解析を付録 A にのせる．

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．これが求める上界式である．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

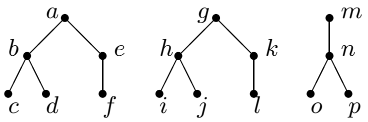


図 2 F 木の集合 $\left\{P_3^m, P_3^m, Q\right\}$
Fig. 2 A set of F-trees $\left\{P_3^m, P_3^m, Q\right\}$.

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

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 からなるポセット $\{P_3^m, P_3^m, P_3^m, P_3^m, P_3^m\}$ が得られる．ここで，各々の P_3^m は入力の $n/5 = 2^{k-3}$ 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，$\mu_3(P_3^m, P_3^m)$ により 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 の一番左の P_3^m のように節点に名前がついていると仮定すると，$b : e$ と $c : d$ である．

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．この選択の最初の並列比較，すなわち全体の $(k - 1)$ 回目の並列比較では，集合

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

を選ぶ．この並列比較からはじめて，あとたかだか 2 ラウンドで上位 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検証することができる．参考のために付録 B に人手による解析をのせる．

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

この章では，$U(n, t)$ に対する下界の式を求め，前章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 章の表 1 より，$10 < n \leq 16$ に対して $U(n, 4) = 6$ である．この範囲の n から始めて，補題 2 を繰り返し適用すれば，定理の範囲の n に対して，下界 $U(n, 4) \geq \lceil \log_2 n \rceil + 2$ が得られる（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，定理の等式が得られる．

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 章の表 1 より，$8 < n \leq 12$ に対して $U(n, 3) = 5$ である．定理 3 の証明と同様に，補題 2 を繰り返し適用する．$8 \cdot 2^k < n \leq 12 \cdot 2^k$，$k \geq 0$ に対して，$U(n, 3) \geq \lceil \log_2 n \rceil + 1$ が得らえる．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得られる．後半も同様にできる．2 章の表 1 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，$U(n, 3) \geq \lceil \log_2 n \rceil + 2$ が得らえる．一方，Aigner の上界[3)]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 章の表 1 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. あとがき

1998 年の 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必要になった．その際に，推移律と引き分けなしを仮定して，7 回戦で十分であることが分かったが，6 回戦でできるかどうかが話題になった．筆者の 1 人が 6 回戦で十分であることを示し（$U(16, 4) \leq 6$），CSA メーリングリストなどで発表した．定理 1 はこのアルゴリズムを一般化して導いたものである．

比較ネットワークは，並列選択問題の（adaptive でない）計算モデルとして詳しく研究されている[4)]．本稿の（adaptive である）並列比較計算モデルは，選択問題に対して比較ネットワークより真に強力であることが知られている．たとえば，比較ネットワークに対する Yao による $V(n, 3)$ の下界式があるが[4)]，（$V(n, 3)$ よりつねに大きいか等しい）$W(n, 3)$ に対して，この

下界式よりも真に小さい上界式が知られている[3]．なお，n 個の要素全部を整列する問題については真に強力であるかどうかは分かっていない．

参 考 文 献

1) JáJá, J.: *An Introduction to Parallel Algorithms*, Addison-Wesley, Reading (1992).
2) Aigner, M.: Producing Posets, *Discrete Mathematics*, Vol.35, pp.1–15 (1981).
3) Aigner, M.: Parallel Complexity of Sorting Problems, *Journal of Algorithms*, Vol.3, pp.79–88 (1982).
4) Knuth, D.E.: *The Art of Computer Programming*, Vol.3 (Sorting and Searching), 2nd Edition, Addison-Wesley, Reading (1998).
5) 佐藤信弘，新藤雅也，野下浩平，中山泰一：並列ゲーム木探索のための分散共有ハッシュ法の評価，情報処理学会論文誌，Vol.42, No.5, pp.1198–1206 (2001).

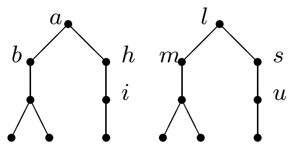
付録 A．$U(n,t)$ のための最終 3 ラウンド

3 章図 1 のポセット $\{P_4^m, P_4^m\}$ から始める．

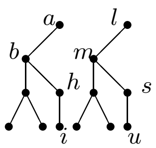
ラウンド 1． 比較の集合は次のとおりである：

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

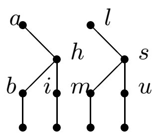
この中の 4 つの比較 $d:e$, $j:k$, $o:p$, $v:w$ によって小さい方の 4 つの要素を削除する．2 つの比較 $c:f$ と $n:q$ によって（最悪の場合）次のポセットができる．



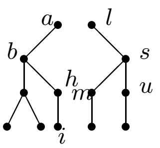
比較 $b:h$ と $m:s$ によって，(対称性を考慮すると）図の 3 つの場合に分かれる（場合 1，2，3）．



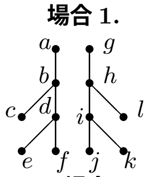
場合 1.



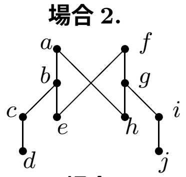
場合 2.



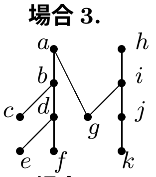
場合 3.



場合 1a.



場合 2a.



場合 3a.

これらの 3 つの場合 1，2，3 について，比較 $a:u$ と $i:l$ によって $a>u$ かつ $i<l$ が成り立つ場合に，図の 3 つの自明でない場合が生じる（場合 1a, 2a, 3a）．ここで要素の名前を新しく付けかえた．それ以外の大小が成り立つ場合は容易に検証できる．たとえば，場合 2 において，もし $a<u$ であれば，$t=2$ に対するポセット $\{m>x, u>a, u>y\}$ が残る．ここで，x と y は適当な要素である．このポセットは後 2 ラウンドで容易に終了できる．

ラウンド 2 と 3（場合 1a）． 比較の集合は次のとおりである：

$$\{e:f,\ j:k,\ b:i,\ d:h,\ a:l,\ c:g\}.$$

比較 $e:f$ と $j:k$ によって 2 つの小さい要素を削除できる．比較 $b:i$ と $d:h$ によって，$b>i$ かつ $d<h$ が成り立つ場合と，それ以外（$b<i$ または $d>h$）の場合を考える．前者の場合は，大きい方の a, g と 4 つの小さい方の要素を削除できる．これで残るポセットは，$t=2$ に対する $\{b>c, h>l\}$ である．これは後 1 ラウンドで終了できる．その以外の場合，残りの比較 $a:l$ と $c:g$ を考慮すると，同様にして後 1 ラウンドで終了することが分かる．

ラウンド 2 と 3（場合 2a）． 比較の集合は次のとおりである：

$$\{b:g,\ c:e,\ h:i\}.$$

比較 $b:g$ によって，(対称性を考慮して）2 つの大きい方の要素と 3 つの小さい方の要素を削除できる．比較 $c:e$ または $h:i$ のいずれかによって，$t=2$ に対する $\{x>y, u>v\}$ の形のポセットが残る．これは後 1 ラウンドで終了できる．

ラウンド 2 と 3（場合 3a）． 比較の集合は次のとおりである：

$$\{e:f,\ d:i,\ b:g,\ c:h\}.$$

比較 $e:f$ によって，1 つの小さい方の要素を削除できる．比較 $d:i$ によって，2 つのポセットが得られる．いま $d>i$ であれば，2 つの大きい方の要素 a と b，および 4 つの小さい方の要素を削除できる．比較 $c:h$ によって，$t=2$ に対する $\{x>y, u>v\}$ の形のポセットが残る．これは後 1 ラウンドで終了できる．一方 $d<i$ であれば，要素 h と 2 つの小さい方の要素（d を含む）を削除できる．これによって，$t=3$ に対するポセット $\{a>b>c, a>g, i>j>k, i>g\}$ が残る．比較 $b:g$ において，$b>g$ であれば，要素 g が削除でき，これより，$t=3$ に対する $\{x>y>z, u>v>w\}$ の形のポセットが残る．これは後 1 ラウンドで終了する（3 章例 2 参照）．一方 $b<g$ であれば，要素 i, b, c を削除でき，これにより，$t=2$ に対する $\{x>y, u>v\}$

の形のポセットが残る．これは後 1 ラウンドで終了できる．

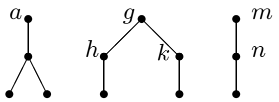
付録 B. $U(n,3)$ のための最終 3 ラウンド

3 章図 2 のポセット $\{P_3^m, P_3^m, Q\}$ から始める．

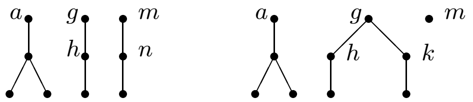
ラウンド 1. 比較の集合は次のとおりである：

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

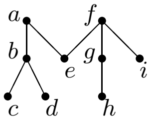
比較 $c:d$，$i:j$，$o:p$ により 3 つの小さい方の要素を削除する．比較 $b:e$ により（最悪の場合）図のポセットになる．



比較 $k:n$ により，$k<n$ または $k>n$ のいずれの場合にも小さい方の 2 つの要素を削除する．これによって図の 2 つのポセットのいずれかになる．



これらの 2 つのポセットにおいて，比較 $a:h$ と $g:m$ により 8 つの場合ができる．



この中で，図に示すポセットが最も難しいことが分かる．ここで，要素の名前を新しく付けかえた．それ以外の場合にできるポセットは，補題 3 に示す変換によって，この難しいポセットから変換できる．

ラウンド 2 と 3. 比較の集合は次のとおりである：

$$\{c:d,\ b:e,\ g:i,\ a:h\}.$$

比較 $c:d$ によって，1 つの小さい方の要素を削除できる．比較 $b:e$ と $g:i$ によって，次の 4 つのポセットができる．

場合 1 $[b>e, g>i]$. $\{a>b>x, f>g>h, g>i\}$.

場合 2 $[b>e, g<i]$. $\{a>b>x, f>i>g\}$.

場合 3 $[b<e, g>i]$. $\{a>e, g>h, g>i\}$，ここで $t=2$ である．

場合 4 $[b<e, g<i]$. $\{a>e, i>g\}$，ここで $t=2$ である．

ここで x は適当な要素である（c または d）．場合 2 と 4 は，後 1 ラウンドで十分である．残り 2 つの場合には，比較 $a:h$ によって次の 4 つのポセットができる．

場合 1a. $\{a>b>x, f>g>i\}$ $(t=3)$ または $\{h,i\}$ $(t=1)$.

場合 3a. $\{a>e, g>i\}$ $(t=2)$ または $\{h,i\}$ $(t=1)$.

いずれの場合も後 1 ラウンドで十分である．

(平成 13 年 10 月 22 日受付)
(平成 14 年 3 月 14 日採録)

野下 浩平（正会員）

1943 年生．1966 年東京大学工学部計数工学科卒業．電電公社，東京大学，中央大学等を経て，現在電気通信大学情報工学科教授，工学博士．アルゴリズムの計算量解析，組合せゲームの理論と実験，卓球に興味を持つ．

佐藤 信弘（正会員）

1996 年群馬工業高等専門学校電子情報工学科卒業．1998 年同専攻科生産システム工学専攻修了．2000 年電気通信大学大学院情報工学専攻博士前期課程修了．現在,（株）アイザック勤務．スケジューリング関連アプリケーション開発に従事．