.

On a Search Problem in Multidimensional Grids

Miklós Ruszinkó†
Computer and Automation Reseach Institute of the
Hungarian Academy of Sciences
H-1518, Budapest POB 63, Hungary; e-mail: h2942rus@ella.hu

Gábor Tardos*
Mathematical Institute of the Hungarian Academy of Sciences
H-1364, Budepest, Pf. 127, Hungary; e-mail: tardos@cs.elte.hu

ABSTRACT

A search problem of G. O. H. Katona [2] was solved in [1] where the unknown element $\mathbf{x}$ *is a point of a 2-dimensional lattice, and the queries are of type "is* $\mathbf{x} = (x_1, x_2) \leq \mathbf{a} = (a_1, a_2)$*?", where* $\mathbf{a}$ *is an arbitrary lattice point and* $\mathbf{x} \leq \mathbf{a}$ *means that* $a_i \leq b_i$ $(i = 1, 2)$*. In the recent paper we consider the generalization of this problem for arbitrary dimension* d*.*

1. Introduction

The following model of the combinatorial search theory was initiated by A. Rényi. A finite set X and a family $\mathcal{A}$ of its subsets are given. We need to find an unknown element x in X by asking questions of type "is x in A?", for any member A of $\mathcal{A}$. Our goal is to minimize the number of questions to identify x in the worst or in the average case.

There are two different approaches of the described model. In the first we have to ask all queries in advence and from the answers we have to identify x. This is called *parallel search*. In the second case the queries are asked sequentially, all questions may depend on the previous answers. This is called an *adaptive search*. In the recent paper we shall give a worst case analysis in the adaptive model for a special search problem.

† This research was supported by the Hungarian National Foundation for Scientific Research, Grant No. T4271

* The author is partially supported by the NSF grant CCR-92-00788, Hungarian National Fund for Scientific Research (OTKA) grant T4271, and a grant of the "Magyar Tudományért" Foundation

Consider the well known fact, that in the adaptive case it is sufficient and necessary to ask $\lceil \log n \rceil$ questions of type "is $x \leq b$?" for some $b \in X$ to find an unknown element x in an n-element ordered set X. But what can we say if X is partially ordered rather than ordered? G. O. H. Katona [2] asked this problem for partial orders that are direct products of chains i. e. when X is the set of bounded integer vectors of a fixed dimension and we must find an unknown vector $\mathbf{x} \in X$ by asking queries of the form "is $\mathbf{x} \leq \mathbf{b}$?". (See next section for formal definition.)

In [1] M. Ruszinkó considered this problem in dimension two and solved it in most cases. In this paper we consider the same problem in arbitrary dimension d. So now we want to find an unknown integer vector $\mathbf{x} = (x_1, \ldots, x_d)$ bounded by $1 \leq x_i \leq a_i$ for $i = 1, \ldots, d$. The queries allowed are of the form "is $\mathbf{x} \leq \mathbf{b}$?" for an integer vector $\mathbf{b} = (b_1, \ldots, b_d)$ where answer "yes" means $x_i \leq b_i$ for all $i = 1, \ldots, d$. It is easy to see that it is necessary to ask $\lceil \sum_{i=1}^{d} \log a_i \rceil$ questions and it is sufficient to ask $\sum_{i=1}^{d} \lceil \log a_i \rceil$ queries (see Proposition 2.1). Thus the gap between these two trivial bounds is at most $d-1$. But which bound is closer to the correct value?

Chapter 2 contains the notations, definitions and trivial cases. In Chapter 3 we improve on the trivial upper bound and prove that in "most" lattices the lower bound is tight. On the other hand we show, that for a certain sequence of lattices $\Omega(d)$ more queries are needed than the trivial lower bound. In Chapter 4 we prove a theorem for the special 2-dimensional case generalizing Theorem 3.3 of [1] giving lots of examples where the lower bound is not tight. In Chapter 5 some open problems are posed.

2. Notations, Definitions, Trivial Cases

For $c \in \mathbf{R}$ we denote by $\lceil c \rceil$ the smallest integer $\geq c$, and by $\lfloor c \rfloor$ the largest integer $\leq c$, $\{c\} = c - \lfloor c \rfloor$. We denote by log the logarithm of base 2.

Let $d \geq 1$ be a fixed integer and let us consider d-dimensional integer vectors $\mathbf{a} = (a_1, \ldots, a_d)$. We say that $\mathbf{a} \leq \mathbf{b}$ if $a_i \leq b_i$ for every $i = 1, \ldots, d$. Let $\mathbf{1} = (1, \ldots, 1)$. We consider the following search problem for a vector $\mathbf{a} \geq \mathbf{1}$.

Problem. An unknown integer vector $\mathbf{1} \leq \mathbf{x} \leq \mathbf{a}$ has to be found by asking questions of the form "is $\mathbf{x} \leq \mathbf{b}$?" for an arbitrary integer vector $\mathbf{b}$. The questions can be asked adaptively, i. e. later questions can depend on the answers for earlier questions. What is the minimum number of questions needed to find the unknown vector $\mathbf{x}$ in the worst case?

Let us denote this minimum by $f(\mathbf{a})$, i. e. it is the minimum for all search strategies of the maximum for all vectors $\mathbf{1} \leq \mathbf{x} \leq \mathbf{a}$ of the number of queries used to find $\mathbf{x}$.

We call the bounds on the function f given by the following proposition the *trivial lower and upper bounds.*

Proposition 2.1. *For any* $\mathbf{a} = (a_1, \ldots, a_d)$ *the following inequalities hold*

$$\left\lceil \sum_{i=1}^{d} \log a_i \right\rceil \leq f(\mathbf{a}) \leq \sum_{i=1}^{d} \lceil \log a_i \rceil .$$

Proof. The lower bound follows from the observation that the number of queries has to be at least the logarithm of size of the set we are searching in and in this case it is $\prod_{i=1}^{d} a_i$. A binary search for each coordinate of $\mathbf{x}$ yields the upper bound. ∎

In the following section we prove some stronger bounds on the function $f(\mathbf{a})$.

3. Sharper upper and lower bounds

The following theorem says that the gap between f and the trivial lower bound can be large. Let us denote the d-dimensional vector with all coordinates 3 by $\mathbf{3}_d$.

Theorem 3.1. $f(\mathbf{3}_d) = 2d$.

The main idea of the proof is that in the middle of the search, instead of dealing with the set of vectors $\mathbf{x}$ consistent with the answers so far we deal with a "nice" subset of it.

Definition. We call a set S of d dimensional integer vectors *nice* if it is a direct product of factors of the following type:

(0) A single point $\{2\}$ or $\{3\}$ in a coordinate,

(1) Two points $\{1, 2\}$ or $\{2, 3\}$ in a coordinate,

(2) The set $\{1, 2, 3\}$ in a coordinate,

(3) The five element set $\{(1, 3), (2, 3), (3, 1), (3, 2), (3, 3)\}$ in two coordinates.

We define the value of a factor of type (i) to be i for $i = 0, 1, 2, 3$ and the *value* $v(S)$ of a nice set S to be the sum of the values of its factors. The proof will be easy after the following lemma.

Lemma 3.1.1. *Let* S *be a nice set of lattice points. For any query of the form "is* $\mathbf{x} \leq \mathbf{b}$*?" there is a nice subset* T *of* S *such that for every element* $\mathbf{x} \in T$ *the question is answered in the same way and* $v(T) \geq v(S) - 1$.

Proof. We call a factor of a nice set S trivial if $\mathbf{b}$'s projection to it is $\geq$ than any element of the factor. The proof is a case analysis.

(a) If a factor of type (1) is nontrivial then we can get T by replacing this factor in S by the one element set $\{2\}$ if it was $\{1, 2\}$ or by $\{3\}$ if it was $\{2, 3\}$.

(b) If $\mathbf{b}$'s projection to a factor of type (2) is ≤ 1 then we can get T by replacing this factor of S by $\{2,3\}$.

(c) If at least two different factors of type 2 are nontrivial then we can now define T by replacing two such factors of S by a factor of type 3.

(d) In case $\mathbf{b}$'s projection to a factor of type 3 is nontrivial then we can get T by replacing this factor of S by a factor $\{3\}$ (of type (0)) in the coordinate $\mathbf{b}$'s projection is less than 3 and by a factor of type 2 in the other coordinate.

(e) In case $\mathbf{b}$'s projection to a factor of type 0 is nontrivial then of course every element of S would yield a "no" answer, so we can take $T = S$.

(f) In case all the factors are trivial then every element in S yields a "yes" answer so we can take $T = S$ again.

(g) Finally, the only remaining case is if all the factors are trivial except for one type 2 factor to which $\mathbf{b}$'s projection is 2. Let us define T by replacing this factor of S by $\{1,2\}$, a factor of type 1.

In the first five cases any vector $\mathbf{x} \in T$ yields "no" answer, in the last two cases any $\mathbf{x} \in T$ yield "yes" answer. We have $v(T) \geq v(S) - 1$ in all of the cases. ∎

Proof of Theorem 3.1. Proposition 2.1 gives $f(\mathbf{3}_d) \leq 2d$. For the lower bound we give an adversary argument. We start with the set of all possible vectors $\mathbf{x}$. This is a nice set of value $2d$. After any question and answer we consider a nice subset of this that is consistent with every answer so far. By Lemma 3.1.1 we can answer the queries of the given form such that the value of the nice set considered decreases by at most 1. This means that after less than $2d$ queries the value is positive so we have more than one possible vectors $\mathbf{x}$ consistent with all the answers, so more queries are needed. Therefore $f(\mathbf{3}_d) \geq 2d$. ∎

Theorem 3.1. gives an example where the value of the function f is big, it is bigger by $\lfloor (2 - \log 3)d \rfloor$ than the trivial lower bound. Here $2 - \log 3 = 0.415\ldots$ Theorem 3.2 states that the difference cannot be much larger. We are going to use the following theorem from [1].

Theorem [Rusz. 3.2.] *If* $\{\log a_1\} + \{\log a_2\} \leq 0.8$ *then* $f(a_1, a_2) = \lceil \log a_1 + \log a_2 \rceil$.

Theorem 3.2. *For any vector* $\mathbf{a} \geq \mathbf{1}$, $\mathbf{a} = (a_1, \ldots, a_d)$ *we have* $f(\mathbf{a}) \leq \lceil \sum_{i=1}^{d} \log a_i \rceil + 0.6d$.

Proof. First suppose, that for any pair of components i, j $(1 \leq i < j \leq d)$ we have $\{\log a_i\} + \{\log a_j\} \geq 0.8$. In this case the gap between the trivial upper and lower bounds is obviously at most $0.6d$.

Now suppose, that the previous condition does not hold. In this case we will prove the

statement by induction on d using Theorem 3.2. of [1].

For $d = 1, 2$ the inequality obviously holds. Now suppose, that $\{\log a_1\} + \{\log a_2\} \leq 0.8$. Then

$$f(\mathbf{a} = a_1, \ldots, a_d) \leq f(a_1, a_2) + f(a_3, \ldots, a_d)$$

$$\leq \lceil \log a_1 + \log a_2 \rceil + \left\lceil \sum_{i=3}^{d} \log a_i \right\rceil + 0.6(d-2) \leq \left\lceil \sum_{i=1}^{d} \log a_i \right\rceil + 0.6d$$

∎

The following lemma of [1] considers a special two-dimensional case where the lower bound of Proposition 2.1 is tight. We are going to use this in the following theorem to prove that this bound is almost always tight. We present the simple proof to be self-contained.

Lemma [Rusz 3.1.3.] *For the two-dimensional vector* $\mathbf{v} = (2^k - 1, 2^k + 1)$ *where* $k \geq 1$ *integer we have* $f(\mathbf{v}) = 2k$.

Proof. As $2k$ is the lower bound in Proposition 2.1 it is enough to give a strategy that finds the unknown vector $\mathbf{x} = (x_1, x_2)$ asking at most $2k$ queries. We ask queries "is $\mathbf{x} \leq \mathbf{b}$?" with $\mathbf{b} = (2^k - 2^{k-i}, 2^k)$ for $i = 1, 2, \ldots, k$ but only till we hear the first "yes" answer. If this happens after the i^{th} question then $2^k - 2^{k-i-1} < x_1 \leq 2^k - 2^{k-i}$ and so it can be found in $k - i$ queries by binary search, while $1 \leq x_2 \leq 2^k$ can be found in k queries by binary search. In case we get all "no" answers to our first k queries, than $x_2 = 2^k + 1$ and $1 \leq x_1 \leq 2^k - 1$ can be found in k queries by binary search. ∎

The following theorem implies that for arbitrary fixed dimension d if every coordinate of the vector $\mathbf{a}$ is big enough then the lower bound of Proposition 2.1 is off by not more than 1 from the real value of $f(\mathbf{a})$. It also implies that the fraction of those vectors $\mathbf{a}$ for which this lower bound does not give the value of $f(\mathbf{a})$ exactly among all integer vectors $\mathbf{1} \leq \mathbf{a} \leq \mathbf{c}$ tends to 0 as all coordinates of $\mathbf{c}$ tends to infinity. The consequence of this statement for the dimension 2 answers problems 2 and 3 of [1].

Theorem 3.3. *For every dimension* $d \geq 1$ *and every* $\epsilon > 0$ *there is an* $n > 0$ *such that for any integer vector* $\mathbf{a} = (a_1, \ldots, a_d)$ *with all coordinates greater than* n *we have* $f(\mathbf{a}) \leq \lceil \sum_{i=1}^{d} \log a_i + \epsilon \rceil$.

Proof. The proof does not use anything but the above lemma, the monotonity of f, and the observation that $f(\mathbf{ab}) \leq f(\mathbf{a}) + f(\mathbf{b})$ where $\mathbf{ab}$ stands for the coordinate-wise product of the positive d-dimensional vectors $\mathbf{a}$ and $\mathbf{b}$. To see this last inequality note that the vector $\mathbf{1} \leq \mathbf{x} \leq \mathbf{ab}$ can uniquely be written in the form $\mathbf{x} = (\mathbf{y} - \mathbf{1})\mathbf{a} + \mathbf{z}$ where $\mathbf{1} \leq \mathbf{y} \leq \mathbf{b}$ and $\mathbf{1} \leq \mathbf{z} \leq \mathbf{a}$. We can find first $\mathbf{y}$ in $f(\mathbf{b})$ queries then find $\mathbf{z}$ in $f(\mathbf{a})$ queries.

Take a positive integer $k > \log(6d/\epsilon)$. Let $\mathbf{v}_{ik}$ be the d-dimensional vector with all 1 coordinates except for the first coordinate being $2^k - 1$ and the i^{th} coordinate being $2^k + 1$ $(i = 2, \ldots, d)$. By Lemma [Rusz 3.1.3] we have $f(\mathbf{v}_{ik}) = 2k$. Let $l_i = \lceil \{\log a_i\}/\{\log(2^k + 1)\} \rceil$ and $\mathbf{w} = (2^{m_1}, \ldots, 2^{m_d})$ where $m_i = \lceil \log a_i - l_i \log(2^k + 1) \rceil$ for $i = 2, \ldots, d$ and $m_1 = \left\lceil \log a_1 - \sum_{i=2}^{d} l_i \log(2^k - 1) \right\rceil$. We use here that all the coordinates of $\mathbf{a}$ are big enough to ensure that $m_i \geq 0$ and to make $\mathbf{w}$ an integer vector. As $l_i \leq 2^k + 1$ it is enough to have $\log a_i \geq dk2^k$ so we can choose $n = 2^{dk2^k}$.

We have $f(\mathbf{w}) = \sum_{i=1}^{d} m_i$ since the upper and lower bounds in Proposition 2.1 are the same in this case. Let us define $\mathbf{b} = \mathbf{w} \prod_{i=2}^{d} \mathbf{v}_{ik}^{l_i}$ a coordinate-wise product of $\sum_{i=2}^{d} l_i + 1$ vectors. By the definition of $\mathbf{w}$ we have $\mathbf{a} \leq \mathbf{b}$ and therefore

$$f(\mathbf{a}) \leq f(\mathbf{b}) \leq 2k \sum_{i=2}^{d} l_i + \sum_{i=1}^{d} m_i$$

by the monotonity of f and the product rule. We claim that

$$2k \sum_{i=2}^{d} l_i + \sum_{i=1}^{d} m_i \leq \left\lceil \sum_{i=1}^{d} \log a_i + \epsilon \right\rceil .$$

This is clearly enough to prove the theorem.

First we observe that

$$k < \log(2^k + 1) < k + \log e/2^k$$

where e is the base of the natural logarithm. Therefore $\{\log(2^k + 1)\} < \log e/2^k$ and for $i = 2, \ldots, d$ by the definition of l_i the value $\log a_i - l_i \log(2^k + 1)$ is close to an integer, it is less than by $\log e/2^k$ less than its upper integer part, which is m_i. The i^{th} coordinate of $\mathbf{b} = (b_1, \ldots, b_d)$ is $b_i = (2^k + 1)^{l_i} 2^{m_i}$ so $0 \leq \log b_i - \log a_i < \log e/2^k$ for $i = 2, \ldots d$. By the definition of m_1 for the first coordinate we have $0 \leq \log b_1 - \log a_1 < 1$. The product of the coordinates of $\mathbf{b}$ is the product of the same for all factors. Taking logarithm we obtain

$$\sum_{i=1}^{d} \log b_i = \log(2^{2k} - 1) \sum_{i=2}^{d} l_i + \sum_{i=1}^{d} m_i.$$

Here

$$\log(2^{2k} - 1) > 2k - \log e/(2^{2k} - 1)$$

and

$$\sum_{i=2}^{d} l_i \leq (d - 1)(2^k + 1).$$

Putting all this together we get

$$2k \sum_{i=2}^{d} l_i + \sum_{i=1}^{d} m_i < \sum_{i=1}^{d} \log b_i + \log e(d - 1)(2^k + 1)/(2^{2k} - 1)$$

$$< \sum_{i=1}^{d} \log a_i + \log e(d-1)/2^k + \log e(d-1)/(2^k-1) + 1$$

$$< \sum_{i=1}^{d} \log a_i + \epsilon + 1 \leq \left\lceil \sum_{i=2}^{d} \log a_i + \epsilon \right\rceil + 1.$$

The third inequality follows from the choice of k. Here the first and the last expressions are integers and therefore they differ by at least 1. This gives the inequality we need to finish the proof. ∎

The threshold value $n = n(d, \epsilon)$ we obtain from this proof is $(6d/\epsilon)^{6d^2/\epsilon}$. This can be improved to $(d/\epsilon)^{6d/\epsilon}$ with no effort.

4. A note for the 2-dimensional case

Theorem 3.3 shows, that the lower bound of Proposition 3.1 is almost always tight. There exist however plenty of vectors for which it is not tight even in dimension 2. In [1] it was established that there are infinitely many such vectors. We extend this result in the following theorem which implies that for a positive fraction of all positive integers u there are infinitely many positive integers v such that the lower bound in Proposition 3.1 for the vector (u, v) is not tight.

Theorem 4.1. *If u is a positive odd integer with $2 - \log 3 < \{\log u\} < \log 3 - 1$ then there are infinitely many positive integers v such that $f((u, v)) = \lceil \log u + \log v \rceil + 1$.*

Proof. Let u be an integer satisfying the conditions of Theorem 4.1. Take an integer $k \geq 1$ with $2^k \equiv 1 \pmod{w}$ simultaneously for all odd integers $1 \leq w \leq u$. We claim that $v = (2^k - 1)/u$ satisfies the statement of the theorem. This is enough for the proof since k can be chosen in infinitely many different ways. By Euler's theorem it can be any multiple of $u!$.

By Proposition 2.1 we have to prove only $f((u, v)) \geq \lceil \log u + \log v \rceil + 1 = k + 1$. Suppose that there exists a search strategy finding $\mathbf{x}$ in k queries. Since in $k - 1$ queries we cannot find an element of a set if it has more than 2^{k-1} elements, the first query "is $\mathbf{x} \leq (a, b)$?" has to be such that at most 2^{k-1} different vectors $\mathbf{x}$ leads to any given answer "yes" or "no". We may suppose here that $1 \leq a \leq u$ and $1 \leq b \leq v$. As the total number of possible vectors $\mathbf{x}$ is $uv = 2^k - 1$ this means that the number of vectors giving "yes" answer ab is 2^{k-1} or $2^{k-1} - 1$. In case $ab = 2^{k-1} - 1$ a is a divisor of $2^{k-1} - 1$ therefore it is odd. On the other hand by choosing v in the way above, we get, that a is a divisor of $2^k - 1$, too. Since these two numbers are co-primes, a has to be 1. Since ab is too small in this case we have to have the other possibility $ab = 2^{k-1}$. This case is possible but clearly in one way only, namely a has to be the largest power of 2 smaller than u and b has to be the largest power

of 2 smaller than v. Let us note here that by assumption $2 - \log 3 < \{\log u\} < \log 3 - 1$ we have $u < (3/2)a$ and $v < (3/2)b$.

Consider what happens after an answer "no" to the first question. There are still $2^{k-1} - 1$ possible vectors $\mathbf{x}$. Out of these vectors 2^{k-2} or $2^{k-2} - 1$ will answer the next question with a "yes" by the same reasons as above. Let the next question be "is $\mathbf{x} \leq (c, d)$?". The following is a case analysis, we find the contradiction by showing that none of the cases can occur.

(a) In case $d \leq b$ the number of possible vectors $\mathbf{x}$ giving answer "yes" is 0 or $(c - a)d \leq (u - a)b < ab/2 - 1 = 2^{k-2} - 1$, which is a contradiction.

(b) In case $c \leq a$ the number of possible vectors $\mathbf{x}$ giving answer "yes" is 0 or $c(d - b) \leq a(v - b) < ab/2 - 1 = 2^{k-2} - 1$, which is a contradiction.

(c) Finally, the only remaining case is $c > a$ and $d > b$. The number of possible vectors $\mathbf{x}$ giving answer "yes" is $cd - ab$ therefore cd must be either $(3/4)2^k$ or $(3/4)2^k - 1$. In the latter case c divides $(3/4)2^k - 1$ and therefore it is odd. Since we chose v in the way above we have that c divides $2^k - 1$, too. As these two numbers are co-primes c has to be 1. Therefore cd cannot be big enough. This contradiction shows that cd has to be $(3/4)2^k$. But now c as a divisor of $(3/4)2^k$ has to be a power of 2 or 3 times a power of 2 but there is no such number $a < c \leq u$.

These contradictions prove the theorem. ∎

5. Open problems

Finally let us consider the following problems. The 2-dimensional vectors (u, v) for which the trivial lower bound on f is not tight are called *loose* in [1].

(1) We do not know if there exist loose vectors (u, u).

This problem appears in [1] and it is the only problem appearing there that is not solved in this paper.

(2) We conjecture that if (u, v) is loose then there are infinitely many integers v' such that (u, v') is loose.

This conjecture is motivated by a case-analysis for small values of u outlined here. For $u \leq 10$ there are no loose vectors (u, v). The first value of u where Theorem 4.1 applies is $u = 11$. (We mention here that this case is already settled in [1].) There are no loose vectors (u, v) with $u = 12, 14, 15, 16, 17, 18, 20, 21$. For $u = 13, 19, 22$ and 23 there are infinitely many loose vectors (u, v). For $u = 13$ and 19 it is possible to ask the first two questions in such a way that the number of vectors $\mathbf{1} \leq \mathbf{x} \leq (u, v)$ consistent with any set of answers is $\leq 2^{\lceil \log uv \rceil - 2}$ i. e. satisfies the "counting criterium", but we cannot ask the third question this way if v is chosen right. For $u = 22$ and 23 even the second question has

to violate the counting criterion for some v. (23 is the second value of u where Theorem 4.1 applies.)

The following stronger conjecture (motivated by the proof of Theorem 4.1) would imply the previous conjecture.

Let u, v, and v' be positive integers. Let $k = \lceil \log uv \rceil$ and $k' = \lceil \log uv' \rceil$. Suppose that for any $w \leq u$ we have $2^k \equiv 2^{k'} \pmod{w}$. If (u, v) is loose and $v < v'$ then (u, v') is also loose.

(3) Find $x = \sup_{\mathbf{a}} \big((f(\mathbf{a}) - \mathrm{tlb}(\mathbf{a}))/d\big)$. Here $\mathbf{a}$ is a positive integer vector, d is its dimension, and $\mathrm{tlb}(\mathbf{a})$ stands for the trivial lower bound of $f(\mathbf{a})$.

By Theorem 3.1 and Theorem 3.2 we have $2 - \log 3 \leq x \leq 0.6$. The upper bound here can be made much closer to $1/2$ with little effort. We conjecture however that the lower bound gives the correct value of x, i. e. the vectors $\mathbf{3}_d$ of Theorem 3.1 are the ones for which the trivial lower bound is worst.

References

[1] M. Ruszinkó, "On a 2-dimensional search problem", *Journal of Statistical Planning and Inference*, Vol. 37 (1993), 371-383.

[2] G. O. H. Katona, personal communication.