for such i, $\binom{i}{j+2^{n-1}} = \binom{i}{j}$. By Lemma 3, the well-known combinatorial identity

$$\binom{i+2^{n-1}}{j+2^{n-1}} = \sum_{k=0}^{j+2^{n-1}} \binom{i}{k}\binom{2^{n-1}}{j+2^{n-1}-k}$$

reduces in GF(2) to

$$\binom{i+2^{n-1}}{j+2^{n-1}} = \binom{i}{j} + \binom{i}{j+2^{n-1}}. \tag{1}$$

But $\binom{i}{j+2^{n-1}}$ is periodic with period 2^n, so that

$$\binom{i+2^{n-1}}{j+2^{n-1}} = \binom{i-2^{n-1}}{j+2^{n-1}}. \tag{2}$$

Note that $0 \le i - 2^{n-1} < 2^{n-1} \le j + 2^{n-1}$, so that the right-hand side of (2) is 0, and (1) becomes, in GF(2),

$$\binom{i}{j+2^{n-1}} = \binom{i}{j}. \qquad \square$$

Corollary 5: For $\boldsymbol{d} = (d_0, d_1, \cdots, d_{2^{n-1}-1}) \neq \mathbf{0}$, let $\boldsymbol{d}' = (\mathbf{0} \vdots \boldsymbol{d}), \mathbf{0} \in \mathrm{GF}(2)^{2^{n-1}}$, then $(\boldsymbol{d}') \in B(2^n)$ has $c(\boldsymbol{d}') = 2^{n-1} + c(\boldsymbol{d})$.

Proof: For $i \ge 0$, suppose

$$d_i = \sum_{j=0}^{c(\boldsymbol{d})-1} a_j \binom{i}{j}$$

with $1 \le c(\boldsymbol{d}) \le 2^{n-1}$ and $a_{c(\boldsymbol{d})-1} = 1$. Then Lemma 4 implies

$$d'_i = \sum_{j=0}^{c(\boldsymbol{d})-1} a_j \binom{i}{j+2^{n-1}}. \qquad \square$$

Theorem 6: Let $\boldsymbol{s} = (s_0, s_1, \cdots, s_{2^n-1}) = (L(\boldsymbol{s}) \vdots R(\boldsymbol{s}))$ be a binary vector with associated sequence $(\boldsymbol{s}) \in B(2^n)$, $n \ge 1$. Form $\boldsymbol{d} = R(\boldsymbol{s}) - L(\boldsymbol{s})$, so that $(\boldsymbol{d}) \in B(2^{n-1})$.

a) If $\boldsymbol{d} = \mathbf{0}$, then $c(\boldsymbol{s}) = c(L(\boldsymbol{s}))$.
b) If $\boldsymbol{d} \neq \mathbf{0}$, then $c(\boldsymbol{s}) = 2^{n-1} + c(\boldsymbol{d})$.

Proof: If $\boldsymbol{d} = \mathbf{0}$, then $R(\boldsymbol{s}) = L(\boldsymbol{s})$, so $(\boldsymbol{s}) = ((L(\boldsymbol{s}) \vdots L(\boldsymbol{s}))) = (L(\boldsymbol{s}))$ and $c(\boldsymbol{s}) = c(L(\boldsymbol{s}))$. If $\boldsymbol{d} \neq \mathbf{0}$, then write

$$\begin{aligned}\boldsymbol{s} &= (L(\boldsymbol{s}) : R(\boldsymbol{s})) \\ &= \big(L(\boldsymbol{s}) \vdots L(\boldsymbol{s})\big) + \big(\mathbf{0} \vdots R(\boldsymbol{s}) - L(\boldsymbol{s})\big) \\ &= \big(L(\boldsymbol{s}) \vdots L(\boldsymbol{s})\big) + (\mathbf{0} \vdots \boldsymbol{d}).\end{aligned}$$

Let $\boldsymbol{d}' = (\mathbf{0} \vdots \boldsymbol{d})$, so that $(\boldsymbol{s}) = (L(\boldsymbol{s})) + (\boldsymbol{d}')$. Note that $c(L(\boldsymbol{s})) \le 2^{n-1}$ and, by Corollary 5, $c(\boldsymbol{d}') = 2^{n-1} + c(\boldsymbol{d}) > 2^{n-1}$, so that the term $\binom{i}{c(\boldsymbol{d})-1+2^{n-1}}$ occurs in the basis expansion of $s_i = L(\boldsymbol{s})_i + d'_i$. Thus $c(\boldsymbol{s}) = c(\boldsymbol{d}') = 2^{n-1} + c(\boldsymbol{d})$. $\square$

The algorithm of Section II applies the result of Theorem 6 recursively, starting with the initial vector $\boldsymbol{s}$ of length 2^n, and stopping when the vector (0) (of complexity 0) or the vector (1) (of complexity 1) is encountered.

Conclusion

For a given sequence of arbitrary period N, the Massey algorithm [3] accepts the sequence sequentially and at each stage computes the connection polynomial for the shortest LFSR that generates the encountered portion of the sequence. The Massey algorithm may have to run through more than one period of length N of the sequence before it stabilizes on the correct connection polynomial. In practice, additional iterations are required to ensure that the algorithm has in fact stabilized. The algorithm given in this correspondence works only for a sequence with period of length $N = 2^n$ and computes the complexity c in $\log N = n$ steps. The connection polynomial $f(E)$ then must be $(E-1)^c$ in this case. The storage requirements of the Massey algorithm depend directly on the eventual complexity of the sequence, while the present algorithm must always store a single period of the sequence, making the algorithm inappropriate for very long periods.

References

[1] E. J. Groth, "Generation of binary sequences with controllable complexity," *IEEE Trans. Inform. Theory*, vol. IT-17, pp. 288–296, 1971.
[2] E. L. Key, "An analysis of the structure and complexity of non-linear binary sequence generators," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 732–736, Nov. 1976.
[3] J. L. Massey, "Shift registers synthesis and BCH decoding," *IEEE Trans. Inform. Theory*, vol. IT-15, pp. 122–127, Jan. 1969.
[4] S. W. Golomb, *Shift Register Sequences*. San Francisco: Holden-Day, 1967.

The Largest Super-Increasing Subset of a Random Set

EHUD D. KARNIN, STUDENT MEMBER, IEEE, AND MARTIN E. HELLMAN, FELLOW, IEEE

***Abstract*—It is shown that the longest super-increasing sequence which can be constructed from a set of n independent uniformly distributed random variables is almost surely asymptotic to $\log_2 n$. Some extensions of this result, as well as the implications for the security of knapsack-based cryptographic systems, are discussed.**

I. Introduction and Motivation

Given a set A of n positive numbers u_i, $i = 1, 2, \cdots, n$ and a positive number S, the knapsack problem [1] is to find a binary solution vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ such that

$$\sum_{i=1}^{n} x_i u_i = S \qquad x_i \in \{0, 1\}. \tag{1}$$

The associated decision problem, i.e., determining whether (1) has a solution, is in the class of NP-complete problems [2]. Therefore, it is believed that solving (1) is very hard in general. However, there are sets A for which the problem is extremely easy.

Consider the case when the sequence u_i $i = 1, 2, \cdots, n$ is *super-increasing*:

$$u_j > \sum_{i=1}^{j-1} u_i \qquad j = 2, 3, \cdots, n. \tag{2}$$

When (2) holds, (1) is solvable in time which grows only linearly in n (cf. [1]). Also when a large subset of A can be ordered as a super-increasing sequence, the effort of solving (1) is substantially reduced (as discussed in Section V).

The knapsack problem forms the basis for a public key cryptographic system [1]. The public key is generated by choosing a

Manuscript received December 7, 1981; revised March 7, 1982. This research was supported in part by the National Security Agency under Contract MDA904-81-C-0414, and in part by Joint Services Electronics Program under Contract DAAG29-81-0057.

The authors are with the Department of Electrical Engineering, Informations System Laboratory, Stanford University, Stanford, CA 94305.

0018-9448/83/0100-0146$01.00

sequence of n super-increasing integers, multiplying each by w and reducing modulo m (where w and m are large integers, typically of the order of 2^{2n}). This transformation is the basis of most pseudorandom number generators and produces integers which appear to be uniformly distributed in $\{0,1,\cdots,m\}$. Dividing by m and neglecting quantization effects (since m is very large), we can model the resulting u_i, $i=1,2,\cdots,n$ as being uniformly distributed in $[0,1]$.

Motivated by this we consider the following question: let $A=\{u_i\}_{i=1}^n$ with each u_i independent and uniformly distributed in $[0,1]$. Look at all subsets of A which have elements that can be ordered as a super-increasing sequence. What is the cardinality μ_n of the largest such super-increasing subset?

This problem and some extensions are studied in the following sections.

II. Main Result

Theorem:

$$\frac{\mu_n}{\log_2 n}\to 1 \qquad \text{a.s.} \tag{3}$$

Proof: We establish a lower and an upper bound for $\mu_n/\log_2 n$.

Lower Bound: Consider the events

$$A_i=\left\{\exists u_i\in[2^{-(i-1)}-2^{-m},2^{-(i-1)}]\right\},\qquad i=1,2,\cdots,m.$$

Then

$$P(\mu_n\ge m)\ge P\left(\bigcap_{i=1}^m A_i\right),$$

which is equivalent to

$$P(\mu_n<m)\le P\left(\bigcup_{i=1}^m A_i^c\right)\le\sum_{i=1}^m P(A_i^c)=m(1-2^{-m})^n.$$

Let $m=(1-\epsilon)\log_2 n$, $\epsilon>0$. Then

$$\begin{aligned}P\left(\frac{\mu_n}{\log_2 n}<1-\epsilon\right)&\le[(1-\epsilon)\log_2 n][1-2^{-(1-\epsilon)\log_2 n}]^n\\&=[(1-\epsilon)\log_2 n]\left[\left(1-\frac{1}{n^{1-\epsilon}}\right)^{n^{1-\epsilon}}\right]^{n^\epsilon}\\&\to[(1-\epsilon)\log_2 n]e^{-n^\epsilon}\to 0.\end{aligned}$$

Moreover these terms are summable, i.e.,

$$\sum_{n=2}^\infty P\left(\frac{\mu_n}{\log_2 n}<1-\epsilon\right)<\infty.$$

Consequently, by the Borel–Cantelli lemma

$$P\left(\liminf\frac{\mu_n}{\log_2 n}<1-\epsilon\right)=0.$$

Since ϵ is arbitrary we get

$$\liminf\frac{\mu_n}{\log_2 n}\ge 1 \qquad \text{a.s.} \tag{4}$$

Upper bound:

$$P(\mu_n\ge m)\le P\{\exists u_i \text{ such that } u_i\le 2^{2-m}\}. \tag{5}$$

We prove this claim by contradiction. From (2), any super-increasing sequence starting with u_i should grow faster than

$$u_i,u_i,2u_i,4u_i,\cdots,2^{m-2}u_i.$$

But $u_i>2^{2-m}$ implies that the last term is greater than one, a contradiction. Relation (5) can be further bounded by

$$P(\mu_n\ge m)\le n2^{2-m}=4n2^{-m}.$$

Let $m=(1+\epsilon)\log_2 n$, $\epsilon>0$. Then

$$P\left(\frac{\mu_n}{\log_2 n}\ge 1+\epsilon\right)\le 4n2^{-(1+\epsilon)\log_2 n}=4n^{-\epsilon}\to 0.$$

To establish the almost surely convergence consider the subsequence $n_i=2^i$, and observe that

$$\sum_{i=1}^\infty P\left(\frac{\mu_{n_i}}{\log_2 n_i}\ge 1+\epsilon\right)\le\sum_{i=1}^\infty 4\cdot 2^{-\epsilon i}<\infty.$$

Hence by the Borel–Cantelli lemma,

$$P\left(\limsup_{i\to\infty}\frac{\mu_{n_i}}{\log_2 n_i}\ge 1+\epsilon\right)=0$$

which implies, since ϵ is arbitrary, that

$$\limsup_{i\to\infty}\frac{\mu_{n_i}}{\log_2 n_i}\le 1 \qquad \text{a.s.}$$

Now consider n in the gap $2^{i-1}<n\le 2^i$. The cardinality μ_n is monotonically increasing, so $\mu_n\le\mu_{2^i}$. Also, $\log_2 n>\log_2 2^{i-1}=i-1$. Hence,

$$\begin{aligned}\frac{\mu_n}{\log_2 n}&\le\frac{\mu_{2^i}}{i-1}=\frac{\mu_{2^i}}{i(1-1/i)}\\&\le\frac{\mu_{2^i}}{\log_2 2^i}(1-1/i_0)^{-1},\qquad i\ge i_0.\end{aligned}$$

Therefore

$$\begin{aligned}\limsup_{n\to\infty}\frac{\mu_n}{\log_2 n}&\le\limsup_{i\to\infty}\frac{\mu_{n_i}}{\log_2 n_i}(1-1/i_0)^{-1}\\&\overset{\text{a.s.}}{\le}(1-1/i_0)^{-1}.\end{aligned}$$

Since i_0 may be taken arbitrarily large, this implies

$$\limsup\frac{\mu_n}{\log_2 n}\le 1 \qquad \text{a.s.} \tag{6}$$

Finally, (4) and (6) yield (3), which completes the proof.

III. Higher Dimensions

In a k-dimensional vector space we define a relation: $\boldsymbol{a}>\boldsymbol{b}$ if each component of $\boldsymbol{a}$ is greater than the corresponding component of $\boldsymbol{b}$. This relation is a partial ordering, and we use it to define a super-increasing sequence of vectors as

$$\boldsymbol{u}_j>\sum_{i=1}^{j-1}\boldsymbol{u}_i,\qquad j=2,3,\cdots,n, \tag{7}$$

where addition of vectors is, as usual, on a component by component basis.

As before let $\{\boldsymbol{u}_i\}_{i=1}^n$ be independent, uniformly distributed in $[0,1]^k$, and let μ_n be the length of the largest super-increasing sequence which can be constructed from them.

Corollary 1:

$$\frac{\mu_n}{\log_2 n}\to\frac{1}{k} \qquad \text{a.s.} \tag{8}$$

Proof: For a lower bound take

$$A_i=\left\{\exists\boldsymbol{u}_i\in[2^{-(i-1)}-2^{-m},2^{-(i-1)}]^k\right\}$$

and proceed as in the proof of the theorem. For the upper bound,

observe that

$$P(\mu_n \geq m) \leq P\left(\exists \boldsymbol{u}_i \text{ such that } \boldsymbol{u}_i < \left(2^{2-m}, 2^{2-m}, \cdots, 2^{2-m}\right)\right)$$

and then proceed as before.

IV. Nonbinary Knapsack

Suppose we modify the knapsack problem by asking for a solution of (1) such that

$$x_i \in \{0, 1, \cdots, B-1\}$$

instead of $x_i \in \{0, 1\}$. Such "compact" knapsacks are of interest in cryptography, because they allow a reduction in the size of the public key [1]. The analog of an easily solved super-increasing knapsack is now

$$u_j > (B-1) \sum_{i=1}^{j-1} u_i \qquad j = 2, 3, \cdots, n, \tag{9}$$

which clearly reduces to (2) for the special case $B = 2$. When (9) holds the non binary knapsack is easily solved, so we ask about μ_n, the cardinality of the largest subset satisfying (9). Again we assume u_i independent, uniformly distributed in [0, 1], and obtain the following.

Corollary 2:

$$\frac{\mu_n}{\log_B n} \to 1 \qquad \text{a.s.} \tag{10}$$

Proof: For a lower bound consider the events

$$A_i = \left\{\exists u_i \in \left[B^{-(i-1)} - B^{-m}, B^{-(i-1)}\right]\right\}$$

and proceed as in the theorem. For the upper bound, observe that

$$P(\mu_n \geq m) \leq P\left(\exists u_i \text{ such that } u_i \leq \frac{B^2}{B-1} B^{-m}\right)$$

and proceed as before. (For the subsequence argument take $n_i = B^i$.)

Note: The extension of this result to a k-dimensional vector space (cf. Section III) is

$$\frac{\mu_n}{\log_B n} \to \frac{1}{k} \qquad \text{a.s.} \tag{11}$$

V. Application and Discussion

In cryptography a knapsack can be used to conceal an n bit message $\boldsymbol{x}$ (see [1]). The cryptanalyst knows u_i, $i = 1, 2, \cdots, n$ and S, and tries to find $\boldsymbol{x}$. Searching over all the 2^n possible $\boldsymbol{x}$ will yield a solution for (1). However, one may also use the following procedure.

For the $n - \mu_n$ u_i's, which are not in the largest super-increasing subset, try all the $2^{n-\mu_n}$ possible x_i. In each trial subtract from S the u_i which correspond to $x_i = 1$. Then see if there is a solution to the associated easy super-increasing knapsack with μ_n components, in $O(\mu_n)$ operations.

By our theorem μ_n is almost surely asymptotic to $\log_2 n$, and the effort of solving (1) is reduced only by a factor of about $2^{\mu_n} = n$, compared to an exhaustive search. Thus the security of a knapsack-based cryptographic system is not substantially decreased by the above procedure.

As a final remark we compare a binary knapsack of length n and a nonbinary knapsack of length N. If both have to conceal the same amount of information, then $n = Nb$, where $b = \log_2 B$.

About $\log_2 n$ bits of information are involved in an "easy" super-increasing subset of the binary knapsack. For the nonbinary knapsack we have by Corollary 2

$$\mu_N \sim \log_B N = \frac{\log_2 (n/b)}{\log_2 B} \sim \frac{\log_2 n}{b}.$$

Since each element carries b bits of information, $b\mu_N \sim \log_2 n$ bits of information are involved in the super-increasing part, which is the same number obtained for the binary knapsack.

Acknowledgment

The authors are thankful to Professor J. Michael Steele of Stanford University, Department of Statistics, for helpful discussions.

References

[1] R. Merkle and M. Hellman, "Hiding information and signatures in trapdoor knapsacks," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 525–530, Sept. 1978.

[2] M. Garey and D. Johnson, *Computers and Intractability—A Guide to the Theory of NP-Completeness*. San Francisco: W. H. Freeman, 1979.

A Combinatorial Approach to Polygon Similarity

DAVID AVIS AND HOSSAM ELGINDY

***Abstract*—A new approach is presented to the classification problem of planar shapes represented by polygons. A shape is abstracted combinatorially by means of its visibility graph, and two shapes are deemed similar whenever their graphs are cyclically isomorphic. Efficient algorithms are presented for performing these operations.**

I. Introduction

In pattern recognition, a common and economical approach to representing planar shapes within a computer is to describe their boundaries by polygons with a finite number of vertices [1, pp. 168–184]. With appropriate selection of vertices on or near the boundary, the shape can be adequately described without affecting its important features. Such polygons may be described by a sequence of edges representing a piecewise approximation to the boundary, or a graph which preserves important properties of the original shape.

The problem of classification requires the partition of the set of all polygons into a number of classes such that the polygons in each class are equivalent (or similar) with respect to a selected equivalence relation. In other words, a polygon can be uniformly mapped onto other polygons in its equivalence class such that certain features are preserved.

A hierarchial approach to the classification problem would involve classifying the polygons into broad equivalence classes, and then using further discrimination criteria within each class.

In this correspondence, we present a purely combinatorial technique. We define a finite graph on the vertices of the polygon and then deal exclusively with the graph. Two polygons are said to be equivalent whenever the corresponding graphs are isomorphic under some cyclic permutation of the vertices.

This equivalence relation decomposes the class of n-vertex polygons into broad equivalence classes. For example, it groups all n-vertex convex polygons in one class, which may be thought of as the structurally most simple class. Further, and more

Manuscript received May 14, 1981; revised April 20, 1982. This work was supported in part by Natural Sciences and Engineering Research Council (N.S.E.R.C.) Grant A3013 and in part by Fonds de Chercheurs et Action Concertée (F.C.A.C.) Grant EQ-1678.

The authors are with the School of Computer Science, McGill University, 805 Sherbrooke Street West, Montreal, PQ, Canada, H3A 2K6.

0018-9448/83/0100-0148$01.00