Abstract: By the method of region counting, a lower bound of $n \log_2 n$ queries is obtained on linear search tree programs that solve the n-dimensional knapsack problem. The region counting involves studying the structure of a subset of the hyperplanes defined by the problem. For this subset of hyperplanes, the result is shown to be tight.

A Non-linear Lower Bound on Linear Search Tree Programs for Solving Knapsack Problems

David Dobkin

Research Report #52

October 1975

Portions of this research were supported by the Office of Naval Research under Grant N00014-75-C-0450 and by the National Science Foundation under Travel Grant DCR-75-15624.

1. Introduction

In a previous paper [1], we showed that any linear search tree that determines membership in a union of a disjoint family of k open sets requires at least $\log_2 k$ queries. This result can actually be extended [2] to show that any search tree using queries that are polynomials of degree $\le p$ requires at least $\frac{1}{p} \log_2 k$ queries to determine membership in a union of a disjoint family of k open sets. In the present paper we will use these results to show that any linear search tree for solving the knapsack problem of dimension n requires at least $n \log_2 n$ queries. This result will follow by showing that a subset of this problem gives rise to at least $\frac{1}{2} n!$ regions. Although the linear search tree model does not appear to have all of the power of the Turing machine model for which the P vs. NP model was first proposed, this non-linear lower bound is derived on the model that is actually used in practice for solving knapsack-type problems. To begin, we review some definitions from [1] to set notation for this paper. The knapsack problems of dimension n is commonly stated as follows: Given an n+1-tuple $\{x_1,\ldots,x_n, b\}$, does there exist a vector $(a_1,\ldots,a_n)$ all of whose components are 0 or 1 such that

$$\sum_{i=1}^{n} a_i x_i = b?$$

We can restate this problem in a geometric fashion by observing that determining whether the n+1-tuple $\{x_1,\ldots,x_n, b\}$ gives rise to a solvable knapsack problem is equivalent to determining whether the point $(x_1/b, x_2/b, \ldots, x_n/b)$ lies on any of a set of hyperplanes in $\mathbb{R}^n$, n-dimensional Euclidean space. In particular, if $1 \le i \le 2^n-1$ is represented in its binary expansion as $i = i_n \cdot 2^{n-1} + i_{n-1} \cdot 2^{n-2} + \ldots + i_2 \cdot 2 + i_1$, we can represent the *ith* of these hyperplanes as $H_i(\underline{v}) - 1 = 0$ where

$$H_i(\underline{v}) = \sum_{j=1}^{n} i_j v_j .$$

In all that follows we shall use this second formulation, which is trivially equivalent to the first.

The linear search tree model is defined as the set of programs consisting of statements of three types:

L_i: If $f(x)\ \mathcal{R}\ 0$ then go to L_j; else go to L_k

L_m: Halt and accept

L_n: Halt and reject

where $f(x)$ is a linear form in the components of the n-vector x and $\mathcal{R}$ is one of the relations >, <, and =. This model can be extended to polynomial search trees of degree ≤ p by allowing $f(x)$ to be a polynomial of degree ≤ p. The result of [1] can then be stated as follows:

Theorem 1: Any linear search tree for determining membership in $\bigcup_{i\in I} A_i$ where each A_i is an open subset of $\mathbb{R}^n$ and the A_i are pairwise disjoint requires at least $\log_2 |I|$ queries in the worst case.

And, as observed, we have the corollary:

Corollary: Any polynomial search tree of degree ≤ p for determining membership in $\bigcup_{i\in I} A_i$ for $\{A_i\}_{i\in I}$, a family of open subsets of $\mathbb{R}^n$ that are pairwise disjoint, requires $\frac{1}{p} \log_2 |I|$ queries in the worst case.

Our main result will be to show that a subset of the knapsack hyperplanes divide $\mathbb{R}^n$ into at least $\frac{1}{2}\, n!$ disjoint open sets, so that solving the knapsack problem (i.e. determining membership in these sets) requires at least $n \log_2 n$ queries in the worst case. The hyperplanes that we will consider are those H_i such that the binary expansion of i has exactly 2 ones in it corresponding to solutions of the problem: Given $\{x_1,\ldots,x_n, b\}$, do there exist $i \neq j$ such that $x_i + x_j = b$? Since the number of regions generated by all the hyperplanes is at least as many as those generated by this subset, this lower bound is a lower bound on the entire problem. We will also show that this lower bound is tight for the problem at hand by demonstrating an appropriate algorithm.

2. A Lower Bound

In this section, we give a characterization of the regions generated by the hyperplanes in the problem under consideration. To begin, we observe that we are seeking to determine how many regions exist that can be expressed as the intersections of halfspaces of the form $x_i + x_j > 1$ or $x_i + x_j < 1$ for $1 \le i < j \le n$. We may represent these spaces by $\underline{S}_{ij} = \{\underline{x} \in \mathbb{R}^n \mid x_i + x_j < 1\}$ and $\overline{S}_{ij} = \{\underline{x} \in \mathbb{R}^n \mid x_i + x_j > 1\}$ for $1 \le i < j \le n$. We may define $\mathcal{T}_n$ as the set of all pairs of the form (i,j) for $1 \le i < j \le n$ and then, for each subset of K of $\mathcal{T}_n$, we ask whether

$$\delta_K = \bigcap_{(i,j)\in K} \underline{S}_{ij} \cap \bigcap_{(i,j)\in K} \overline{S}_{ij}$$

is empty or not. There are $2^{\binom{n}{2}}$ such subsets and at first one is tempted to believe that each subset gives rise to an open subset of $\mathbb{R}^n$. We may observe, however, that for $n = 4$, if $K = \{(1,2),(3,4)\}$, then $\delta_K = \underline{S}_{12} \cap \underline{S}_{34} \cap \overline{S}_{13} \cap \overline{S}_{24} \cap \overline{S}_{14} \cap \overline{S}_{23}$ is empty since a point in this set would satisfy $x_1 + x_2 < 1$, $x_3 + x_4 < 1$ as well as $x_1 + x_3 > 1$, $x_2 + x_4 > 1$, a contradiction. Furthermore, it is clear that if K_1 and K_2 are different subsets of $\mathcal{T}_n$ such that δ_{K_1} and δ_{K_2} are non-empty then these intersections are disjoint open sets. Openness follows since these sets are the intersections of finite collections of open sets, and to show that they are disjoint we observe that there is an (i,j) that belongs to K_1 and not K_2 (or to K_2 and not K_1) so that every point in δ_{K_1} satisfies $x_i + x_j < 1$ and every point in δ_{K_2} satisfies $x_i + x_j > 1$.

Now we wish to determine conditions on $K \subset \mathcal{T}_n$ such that δ_K is nonempty. We begin with one such set and apply the appropriate permutations to generate at least $\frac{1}{2}n!$ others.

Lemma: If $K = \{(i,j) \mid i + j \le n + 1\}$, then δ_K is non-empty.

Proof: Let $x_i = \dfrac{i}{n + 3/2}$ for $1 \le i \le n$; then $x_i + x_j < 1$ if and only if $i + j < n + 3/2$ if and only if $(i,j) \in K$. Hence the point $(x_1,\ldots,x_n)$ is an element of K, which is therefore non-empty. □

We observe that K is a subset of T_n consisting of $\lceil \frac{n^2}{2} \rceil$ elements and that i occurs in exactly n - i pairs of K if $i \le \frac{n}{2}$ and n + 1 - i pairs if $i > \frac{n}{2}$. For each permutation π of n elements we define $\pi(K)$ as $\{(\pi(i),\pi(j)) \mid (i,j) \in K\}$ and observe that if π_1 and π_2 are two permutations such that π_1 and π_2 do not send $\lfloor \frac{n}{2} \rfloor$ and $\lfloor \frac{n}{2} \rfloor + 1$ into the same pair of integers then $\pi_1(K)$ and $\pi_2(K)$ are necessarily different subsets of T_n. Thus,

Theorem 2: There are at least $\frac{1}{2}$ n! different subsets K of T_n that give rise to non-empty sets δ_K.

Using this theorem in concert with Theorem 1, we obtain the main result of this paper.

Theorem 3: Any linear search tree for solving the n-dimensional knapsack problem (or even the n-dimensional knapsack problem restricted to solutions of the form $x_i + x_j = b$) must require $\frac{1}{2}$ n log n - l.o.t. queries in the worst case.

Furthermore, since polynomial queries can be simulated by linear queries, we have:

Corollary: Any polynomial search tree of degree $\le$ p requires at least $\frac{1}{p}$ n $\log_2$ n queries in the worst case to solve the n-dimensional knapsack problem.

3. An Upper Bound

In this section, a method is given to match the lower bound given in the previous section for the restricted knapsack problem mentioned there. Given an input $\{x_1,\ldots,x_n, b\}$, we wish to determine whether there exist distinct integers i and j such that $x_i + x_j = b$. To do so, we may use the following:

Algorithm KS2:

I. Sort $x_1,\ldots,x_n$ to yield a sorted list $y_1,\ldots,y_n$ such that, for $i > j$, $y_i \ge y_j$.

II. Test to determine whether any y_i is $\frac{1}{2}$. If two or more are, halt and accept the input. If one is, drop this element from the list and proceed to Step III. If none is, proceed to Step III.

III. For each i, $1 \le i \le n$, determine the least j_i such that $y_i + y_{j_i} < 1$. If $y_i + y_{j_i - 1} = 1$, halt and accept; otherwise, continue.

IV. Halt and reject.

Step I in this algorithm requires $n \log_2 n$ steps, Step II can be done in $\log_2 n$ steps by binary search, and Step III can be done in at most 2n steps by a merging strategy. Therefore the entire algorithm requires $n \log_2 n$ + l.o.t. steps matching the leading term in the upper bound.

4. Conclusions and Research Problems

The major results of this paper are a lower bound of n log n queries for the solution of the knapsack problem with a linear search tree and a similar upper bound for the restricted version of the problem under consideration. These results may possibly be improved by considering less restricted versions of the problem. For example, it is reasonable to conjecture that a better lower bound could be obtained by considering the regions generated by all 2^n knapsack hyperplanes rather than merely a subset of $\binom{n}{2}$ such hyperplanes. Unfortunately, Strassen [2] has observed that no better lower bound than $O(n^2)$ can be obtained by applying Theorem 1, as at most $O(2^{n^2})$ regions will be generated by a set of 2^n hyperplanes in $\mathbb{R}^n$. Such a lower bound has been given for a more general hyperplane search problem in [1].

References

[1] D. Dobkin and R. Lipton.
On the complexity of computations under varying sets of primitive operations.
Proceedings of the 2nd GI Symposium on Automata Theory and Formal Languages, Springer-Verlag. To appear.

[2] V. Strassen.
Personal communication, May 1975.