Reprinted from JOURNAL OF COMPUTER AND SYSTEM SCIENCES
All Rights Reserved by Academic Press, New York and London

Vol. 16, No. 3, June 1978
*Printed in Belgium*

# A Lower Bound of $\frac{1}{2}n^2$ on Linear Search Programs for the Knapsack Problem*

DAVID DOBKIN AND RICHARD J. LIPTON

*Department of Computer Science, Yale University, New Haven, Connecticut 06520*

Received June 17, 1976; revised June 10, 1977

## 1. INTRODUCTION

The purpose of this paper is to establish the following theorem:

THEOREM. *For each $n$, any linear search tree that solves the $n$-dimensional knapsack problem requires at least $\frac{1}{2}n^2$ comparisons.*

Previously the best known lower bound on this problem was $n \log n$ [1]. The result presented here is the first lower bound of better than $n \log n$ given for an *NP*-complete problem for a model that is actually used in practice. Previous non-linear lower bounds have been for computations involving only monotone circuits [8] or fanout limited to one. Our theorem is derived by combining results on linear search tree complexity [4] with results from threshold logic [11]. In Section 2, we begin by presenting the results on linear search trees and threshold logic. Section 3 is devoted to using these results to obtain our main theorem.

## 2. BASIC CONCEPTS

In this section we introduce the basic concepts necessary to the understanding of our main theorem. To begin, we present the model for which our bound holds. It has previously been studied in [6, 7, 10].

DEFINITION. A *linear search tree* program is a program consisting of statements of one of the forms:

(a) $L_i$: if $f(x) > 0$ then go to $L_j$ else go to $L_k$;
(b) $L_i$: halt and accept input $x$;
(c) $L_j$: halt and reject input $x$.

* Portions of this research were supported by the Office of Naval Research under Grant N00014-75-C-0450 to the first author and the National Science Foundation under Grant DCR-74-12870 to the second author.



0022-0000/78/0163-0413$02.00/0

In (a) $f(x)$ is an affine function (i.e., $f(x) = \sum_{i=1}^{\infty} a_i x_i + a_0$ for some $a_0, a_1, \ldots, a_n$) of the input $x = (x_1, \ldots, x_n)$ which is assumed to be from some euclidean space $E^n$. Moreover, the program is assumed to be loop free.

In a natural way each linear search tree program computes some predicate on $E^n$. The complexity of such a program on a given input is the number of statements executed on this input.

In proving our results, we shall make use of the following theorem which is proved here for completeness.

THEOREM [4]. *Any linear search tree program that determines membership in the set*

$$\bigcup_{i \in I} A_i$$

*where the $A_i$ are pairwise disjoint nonempty open subsets of $E^n$ requires at least* $\log_2 |I|$ *queries for almost all inputs.*

*Proof.* We prove that any such search tree $T$ with leaves $D_1, \ldots, D_r$ has $r \geqslant |I|$ and hence a path of depth $\geqslant \log_2 |I|$. The leaves partition $E^n$ and, for each $j$, $D_j$ is an accepting leaf if $D_j \subseteq \bigcup_{i \in I} A_i$ and a rejecting leaf otherwise. The theorem then follows from the observation for each $j$, $D_j$ can intersect at most one $A_i$, since any convex region containing points of $A_i$ and $A_{i'}$ ($i \neq i'$) must contain points not in $\bigcup_{i \in I} A_i$. ∎

In this paper we shall study the complexity of linear search trees for the $n$-dimensional knapsack problem, which we state as a geometric problem. It should be noted, however, that our methods can be applied to many other problems. We may state two equivalent versions of this problem.

KNAPSACK PROBLEM (KSn). (i) Given a point $(x_1, \ldots, x_n) \in E^n$, does there exist a subset $I$ such that $\sum_{i \in I} x_i = 1$?

(ii) Given the hyperplanes $H_\alpha$, $\alpha \in \{0, 1\}^n$ where

$$H_\alpha \triangleq \left\{ (y_1, \ldots, y_n) \in E^n \;\middle|\; \sum_{i=1}^{n} \alpha_i y_i = 1 \right\},$$

does $(x_1, \ldots, x_n)$ lie on some hyperplane?

Clearly, these two formulations are equivalent and they both correspond to the usual knapsack problem which is *NP*-complete [5].

The lower bound established here is proved by appealing to results from theshold logic. Before defining the necessary terms from this field, we demonstrate our method and the chief obstacle in applying it.

Let $\Gamma = \{0, 1\}^n - \{0^n\}$. Say a point $x$ is *above* (*below*) the hyperplane $H_\alpha$ with $\alpha \in \Gamma$ provided

$$\sum_{i=1}^{n} \alpha_i x_i - 1$$

is positive (negative). Also let $R_I$ for $I \subseteq \Gamma$ be the set

$$\{x \in E^n \mid x \text{ is above } H_\alpha \text{ with } \alpha \in I \text{ and below } H_\alpha \text{ with } \alpha \notin I\}.$$

Intuitively, $R_I$ is one of the regions formed by the hyperplanes. There are $2^{2^n-1}$ possible such regions; however, many of these regions are empty. For example,

$$x_1 + x_2 > 1, \qquad x_3 + x_4 > 1, \qquad x_1 + x_3 < 1, \qquad x_2 + x_4 < 1$$

is empty. This example shows that the key problem is to determine how many regions are formed by the hyperplanes $\{H_\alpha\}\alpha \in \Gamma$.

The answer to this problem lies in threshold logic. We will now sketch the relevant results. Further details appear in [9].

DEFINITION. Let $A$ be a subset of $\{0, 1\}^n$. Then the partition of $\{0, 1\}^n$ into $A$ and $\{0, 1\}^n - A$ corresponds to a *threshold function* provided there exist *weights* $w_1, \ldots, w_n$ such that

(1) $x_1 \cdots x_n \in A$ iff $w_1x_1 + \cdots + w_nx_n > 1$.

(2) $x_1 \cdots x_n \notin A$ iff $w_1x_1 + \cdots + w_nx_n < 1$.

Note that (2) does not follow from (1).

Let $N(n)$ be the number of such threshold functions, then [11] shows that

$$2^{\frac{1}{2}n^2} \leqslant N(n) \leqslant 2^{n^2}.$$

In the next section we use this result to obtain our lower bound.

## 3. MAIN RESULTS

In this section we prove our main result, i.e., that any linear search tree for $KS_n$ requires at least $\frac{1}{2}n^2$ comparisons. We first state a technical lemma:

LEMMA. (1) *$R_I$ is an open set for $I \subseteq \Gamma$.*

(2) *$R_{I_1} = R_{I_2}$ implies that $I_1 = I_2$ for $I_1, I_2 \subseteq \Gamma$.*

The proof of this is elementary and is omitted. This lemma shows (part (2)) that we need only prove that $R_I$ is nonempty for many sets $I$ in order to prove our theorem. The next lemma does this.

LEMMA. *Suppose that $A$ partitions $\{0, 1\}^n$ and gives rise to a threshold function. Then $R_A$ is nonempty.*

*Proof.* Let $w_1, \ldots, w_n$ be weights for $A$. Now we claim that $w = (w_1, \ldots, w_n) \in R_A$.

(a) Let $\alpha$ be in $A$. Then $w$ is above $H_\alpha$ since

$$\sum_{i=1}^{n} \alpha_i w_i > 1$$

by the definition of threshold function.

(b) Let $\alpha$ be in $\{0, 1\}^n - A$. Then $w$ is below $H_\alpha$ since

$$\sum_{i=1}^{n} \alpha_i w_i < 1$$

and again this follows by the definition of threshold function. Thus we have shown that $w < R_A$. ∎

In summary we have shown that there are at least $2^{1/2n^2}$ distinct open sets $R_I$'s. An appeal to our earlier theorem [4] yields the claimed lower bound.

Finding an upper bound on the linear search tree complexity of knapsack problem appears to be a nontrivial problem. Two possible methods of attack are available. In the first, an algorithm is sought that works uniformly in $n$. That is, we seek a single method of solving knapsack problems of all dimensions. The existence of such an algorithm that runs in polynomial time is unlikely because this would imply that $P = NP$. But, for each $n$, it may be possible to construct a linear search tree that solves all $n$-dimensional knapsack problems. To construct such a tree; it is necessary to study partitions of the set of knapsack regions by new hyperplanes in order to determine appropriate tests at each stage of the algorithm. Based on considerations of the structure of the regions of the knapsack problem, we conjecture that a polynomial-time algorithm does exist for this problem. The existence of such an algorithm would resolve an open question posed in [3] but would not show that $P$ and $NP$ are equal for the reason given there.

*Note added in proof.* We have recently discovered an algorithm requiring polynomial expected time for the knapsack problem studied here.

## References

1. D. Dobkin, A Non-linear Lower Bound on Linear Search Tree Programs for Solving Knapsack Problems, Yale Computer Science Research Report No. 52, 1975.
2. D. Dobkin and R. Lipton, The complexity of searching lines in the plane, to appear.
3. D. Dobkin and R. Lipton, On some generalizations of binary search, ACM Symposium on the Theory of Computing, Seattle, Washington, May 1974.
4. D. Dobkin and R. Lipton, On the complexity of computations under varying sets of primitive operations, *in* "Automata Theory and Formal Languages," Springer–Verlag Lecture Notes in Computer Science, No. 33, Springer–Verlag, Berlin/New York, 1975.
5. R. Karp, Reducibilities among combinatorial problems, *in* "Complexity of Computer Computations" (R. Miller and J. Thatcher, Eds.), pp. 85–104, Plenum, New York, 1972.

6. M. Rabin, Proving the simultaneous positivity of linear forms, *J. Comput. System Sci.* 6 (1972), 639–650.
7. E. Reingold, Computing the maximum and the median, Twelfth Annual Symposium on Switching and Automata Theory, 1971.
8. C. Schnorr, A lower bound on the number of additions in monotone computations of monotone rational polynomials, Unpublished manuscript.
9. E. L. Sheng, "Threshold Logic," Academic Press, New York, 1969.
10. P. Spira, Complete linear proofs of systems of linear inequalities, *J. Comput. System Sci.* 6 (1972), 205–216.
11. S. Yajima and T. Ibaraki, A lower bound on the number of threshold functions, *IEEE EC* 14 (1965), 926–92.9

Printed by the St Catherine Press Ltd., Tempelhof 37, Bruges, Belgium.