View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by NASA Technical Reports Server

NASA CR- 178,162

NASA Contractor Report 178162

ICASE REPORT NO. 86-55

NASA-CR-178162
19860021764

ICASE

EXPECTED PERFORMANCE OF m-SOLUTION BACKTRACKING

David M. Nicol

FOR REFERENCE

NOT TO BE TAKEN FROM THIS ROOM

Contract No. NAS1-18107

August 1986

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

LIBRARY COPY

SEP 15 1986

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

NF00192

Expected Performance of m-Solution Backtracking

David M. Nicol

Institute for Computer Applications in Science and Engineering

Abstract This paper derives upper bounds on the expected number of search tree nodes visited during an m-solution backtracking search, a search which terminates after some pre-selected number m problem solutions are found. The search behavior is assumed to have a general probabilistic structure. Our results are stated in terms of node expansion and contraction. A visited search tree node is said to be *expanding* if the mean number of its children visited by the search exceeds 1, and is *contracting* otherwise. We show that if every node expands, or if every node contracts, then the number of search tree nodes visited by a search has an upper bound which is linear in the depth of the tree, in the mean number of children a node has, and in the number of solutions sought. We also derive bounds linear in the depth of the tree in some situations where an upper portion of the tree contracts (expands), while the lower portion expands (contracts). While previous analysises of 1-solution backtracking have concluded that the expected performance is always linear in the tree depth, our model allows super-linear expected performance. By generalizing previous work in the expected behavior of backtracking, we are better able to identify classes of trees which can be searched in linear expected time.

Keywords: Random search, random trees, backtracking, depth-first search.

This research was supported by the National Aeronautics and Space Administration under NASA Contract Number NAS1-18107 while the author was in residence at ICASE, NASA Langley Research Center, Hampton, VA 23665.

N86-31326 #

1. Introduction

Backtracking (or depth-first search) is a powerful tool for quickly finding solutions to certain types of problems whose potential solution space is huge. Such a search may terminate upon finding one solution, or may attempt to discover all solutions. We will study *m–solution searches*, which terminate once a preselected number m solutions are discovered (or when it is determined that fewer than m solutions exist). Backtracking is used to solve problems which seek an n-vector $<X_1, \ldots, X_n>$ where each X_i is an element of a finite set S_i, such that some condition $C(<X_1, \ldots, X_n>)$ is satisfied. For example, backtracking is often used to solve the n-Queen's problem. In this case, each vector coordinate position represents a chess board column, $X_i = j$ means that a queen is placed in column i and row j, and the condition C is that no queen attacks any other. A backtracking search exploits necessary subconditions for a solution. The necessary subconditions $\{C_j\}$ are chosen so that if $C_j(<X_1, \ldots, X_j>)$ is satisfied, then $C_i(<X_1, \ldots, X_i>)$ is satisfied for i such that $1 \le i < j$. Given a candidate partial solution $<X_1, \ldots, X_k>$, the search checks condition $C_k(<X_1, \ldots, X_k>)$. If that condition is satisfied, the partial solution is extended to $<X_1, \ldots, X_{k+1}>$ for some $X_{k+1} \in S_{k+1}$, and the extended candidate partial solution is tested against C_{k+1}. If on the other hand $C_k(<X_1, \ldots, X_k>)$ is not satisfied, then an alternate value $\hat{X}_k \in S_k$ which has not yet been appended to $<X_1, \ldots, X_{k-1}>$ (if one exists) is chosen, and the partial solution $<X_1, \ldots, X_{k-1}, \hat{X}_k>$ is checked against C_k. If all elements of S_k have already been appended to $<X_1, \ldots, X_{k-1}>$ and have failed to satisfy C_k, the algorithm "backtracks" and attempts new extensions to $<X_1, \ldots, X_{k-2}>$. Most basic algorithm texts discuss the details of backtracking, for example, see [5].

The search space of a backtracking search can be represented by a tree. A tree node is uniquely identified by the path between it and the tree root; a path of length j identifies a unique partial solution with j coordinate assignments; a node's set of children represent all possible extensions to the partial solution represented by the node. Leaf nodes represent potential full solutions. Viewed in this way,

backtracking is a depth first search of the tree with pruning. Pruning occurs if a tested partial solution fails to satisfy the appropriate condition: the entire subtree rooted in the failing node is pruned from the search space, and the search "backtracks". Backtracking's success lies in this (problem dependent) ability to dynamically prune large portions of the search tree. However, backtracking's worst case performance is exponential in the size of the problem; nevertheless, many problems do not exhibit this worst case performance. A more meaningful measure is the *expected* performance. The usual method for determining the expected or average performance of an algorithm is to assume that problem parameters are random, or that the outcome of a decision is random (for examples, see [6]). One way of introducing randomness into backtracking is to suppose that the success or failure of a visited node is random. This approach is adopted by [12], and is the one we will explore. While a probabilistic model may not describe any particular backtracking search very accurately, it is useful for determining performance as a function of general search space characteristics. As pointed out in [7], such a model can also reveal the sensitivity of performance to the search space parameters.

A number of researchers have considered the expected performance of backtracking, or related problems. Extensive analysis of branch and bound methods on random search trees is found in [11]. That work employs the theory of branching processes [4], which provides a very uniform probability structure for the search tree. In [KaPe83], the problem of finding a least cost path in a random binary tree is considered; this analysis also employs the theory of branching processes. Backtracking searches which seek all solutions to random CNF satisfiability problems are considered in [2]. The randomness introduced to the search model there is intimately related to the problem type (CNF satisfiability), allowing a tight analysis of that problem. Close approximations for the mean and variance of the number of nodes visited in a depth-first search which terminates with one solution are derived in [12]. Our search model closely resembles theirs, but our approach is quite different. The analysis in [12] assumes that the search tree is binary, and that the randomness in node evaluation outcomes is uniform across the entire tree; every visited node has probability p of being a successful partial solution. The

search tree has an extremely uniform structure under these assumptions, allowing [12] to identify recurrence relations, and closely approximate their solution. Renewal theory is employed in [13] to derive approximate expected bounds on memory requirements for general branch-and-bound methods.

All of the research mentioned above introduces randomness in a very uniform way. This is accepted practice, and is usually required for analytic tractability. For the purposes of general behavioral description, uniform randomness is adequate, unless the model deviates strongly from known behavior. For backtracking, this is exactly the case. Intuition suggests that in general, the larger the size of the partial solution, the harder it will be to extend that solution. Yet under the assumption of uniform randomness, the likelihood of pruning a node and its subtree is the same anywhere in the search tree. Furthermore, the assumption of uniform randomness has led to surprising results. Both [12] and [11] derive bounds on the expected number of search nodes visited during a 1-solution search; these bounds are linear in the tree depth *regardless of the probability parameter p.* Under their modeling assumptions, we must conclude that the average number of nodes visited by a 1-solution search is always linear in the depth of the search tree.

Uniform randomness does not accurately model our expectations about pruning behavior; furthermore, its assumption produces results whose strength is counter-intuitive. These observations have led us to consider a more general probabilistic model of backtracking's behavior. In doing so, we reaffirm much of the strength of the results derived in [12] and [11]. However, unlike the models in [12] and [11], our model exposes a class of searches having super-linear complexity in the depth of the search tree. By generalizing the probabilities which model searching behavior, we are thus better able to classify searches which will have linear expected performance. We will allow a general probability structure on a tree with depth D, whose nodes have random numbers of children. Defining natural notions of node expansion and contraction, we will show that if all nodes expand, or if all nodes contract, then there exist linear upper bounds on the number of nodes visited in a backtracking search for m solu-

tions. These bounds are derived in terms of the probability structure, the number of solutions sought, and the depth of the search tree. We then derive upper bounds for some search trees having both expanding and contracting nodes. These bounds are also linear in the depth of the search tree. Finally, we give an example of a search with non-zero extension probabilities which has super-linear complexity.

2. Model Definition

A backtracking search is a depth-first search, with the provision that it does not traverse any subtree rooted in a failed node. We model the decision to prune by supposing that every explored node has an *extension probability* of representing a successful partial solution. The extension probability for a leaf node is the probability that the leaf node represents a full solution. We also assume that a given visited node's success is independent of any other node's at the same depth. It is important to observe that this probability is conditioned on the event of the node being visited by the search. Given that a node is successful, we suppose that it has a random number of children, and that the mean number of children is n (this assumption is similar to one in [11]). We allow each node's distribution of children to be unique, but assume that each distribution is "new better than used in expectation", or NBUE [10]. A nonnegative random variable Y is NBUE if $E[Y - a \mid Y > a] \leq E[Y]$ for all $a \geq 0$. Common examples of continuous NBUE distributions are hypo-exponentials, normals, and certain classes of gamma and Weibull distributions. An appropriately constructed discrete approximation to a continuous NBUE random variable will be NBUE, and the degenerate constant random variable is NBUE. We say that a node with extension probability p *expands* if $p > 1/n$, because given that the node is visited, the expected number of its children which are visited is

$$pn + (1 - p)\cdot 0 > 1.$$

We say that a search tree is expanding if each of its nodes expands. Likewise, if $p \leq 1/n$, we say that the node contracts; we say that the search tree contracts if every one of its nodes contracts.

Figures 1, 2, and 3 illustrate some of these ideas. Figure 1 depicts a small search tree, where each node is labeled with its extension probability. Figure 2 shows a possible realization of the nodes searched, where the entire left subtree is traversed, but the right child of the root fails, so that none of its children are visited. Assuming that $n = 2$, we see that the root's left child is expanding ($0.6 > 1/2$) while the root's right child is contracting ($0.4 < 1/2$). It is important to note that the definitions of expanding and contracting nodes are in terms of the *mean* value n, not the number of children actually realized by a parent. Figure 3 depicts the nodes searched during a 1-solution search which discovers that the second leaf node visited is a successful full solution.

We will alternately consider two types of backtracking searches. A *full* search is one which terminates only after all solutions have been found. We let N denote the random number of nodes visited during a full search, we also let N_j denote the random number of nodes visited at depth j. For many problems, we are satisfied with finding only one solution, and may substantially reduce the number of nodes visited by stopping with the first solution. We call a search which terminates with the first solution (or exhausts the search tree) a 1-solution search. Let T denote the random number of nodes visited during a 1-solution search, and let T_j denote the random number of nodes visited at depth j during a 1-solution search. Our approach also supports generalization of 1-solution searching, an m-solution search. An m-solution search might be used in a situation where it is too costly to search for all solutions, but a sizable sampling of solutions is desired. The random variables $T(m)$ and $T_j(m)$ are the immediate m-solution analogs to T and T_j. We will use $E[N]$ and $E[T(m)]$ as measures of backtracking's average complexity. As noted in [11] and [12], the average *time* complexity of searching may be larger, being dependent on the complexity of the evaluation of the necessary subconditions.

Our results are stated in terms of bounds on the maximal and minimal extension probabilities among all search tree nodes. We let $p_j^{\max}$ denote the maximum extension probability among all nodes at depth j; we similarly define the depth dependent minimum $p_j^{\min}$. For every depth J (recall that the

tree has depth D) we define

$$p_{+J}^{\max} = \max_{1 \le j \le J} \{ p_j^{\max} \}, \qquad p_{+J}^{\min} = \min_{1 \le j \le J} \{ p_j^{\min} \},$$

$$p_{J+}^{\max} = \max_{J < j \le D} \{ p_j^{\max} \}, \qquad p_{J+}^{\min} = \min_{J < j \le D} \{ p_j^{\min} \},$$

and finally, $p^{\min} = p_{+D}^{\min}$ and $p^{\max} = p_{+D}^{\max}$. Table I summarizes our model definitions.

Under our model assumptions, it is possible to find extension probabilities which lead to exponentially complex 1-solution searches. Suppose that the extension probabilities for all nodes at depths 1,2, $\cdots$, D–1 are equal to 1, and that the solution probabilities at depth D are all equal to 0. The 1-solution search will never find a solution, but will visit every node in the search tree. This degenerate case clearly has complexity which grows exponentially in the depth of the tree. While this example is not likely to represent any interesting problem particularly well, it does illustrate that the complexity of a 1-solution search can be high. By identifying classes of trees which yield good expected complexity, we are better able to identify classes of trees which do not.

Model Parameters	
Notation	Definition
D	Depth of search tree
N	Number of nodes visited in full search
m	Number of solutions sought
$T(m)$	Number of nodes visited in m-solution search
n	Mean number of children of successful node
$p_{+J}^{\max}$	Maximum extension probability at depths up to J
$p_{+J}^{\min}$	Minimum extension probability at depths up to J
$p_{J+}^{\max}$	Maximum extension probability at depths greater than J
$p_{J+}^{\min}$	Minimum extension probability at depths greater than J
$p^{\max}$	Maximum extension probability among all nodes
$p^{\min}$	Minimum extension probability among all nodes

Table I

3. Summary of Results

Our analysis will consider four classes of search trees. We first treat the class of contracting trees ($p^{max} \leq 1/n$), and show by Lemma 1 that $E[N] \leq nD$. We then examine the class of expanding trees ($p^{min} > 1/n$), and give in Lemma 2 an upper bound on $E[T(m)]$. This bound is linear in D, in m, and in n. Together, Lemmas 1 and 2 address the model assumptions considered in [12]. While our bounds aren't as tight as the results in [12], they are considerably more general. Lemmas 3,4, and 5 treat trees with mixtures of expanding and contracting nodes. In Lemma 3 we give a bound which is marginally linear in D, m, and n for the situation where an upper portion (near the root) of the tree has contracting nodes, and the lower portion has expanding nodes. This indication of good performance is not surprising; contracting nodes near the root tend to prune large portions of the tree. These results serve as analytic vindication of the search rearrangement strategy studied in [1] and [3] (a formal analysis of average behavior under this strategy for the CNF satisfiability problem is reported in [9]). Search rearrangement strives to reorder the sequence of variable assignments so as to increase the likelihood of pruning at shallow levels of the search tree. Finally, Lemmas 4 and 5 treat intuitively appealing situations where it is relatively easier to extend small partial solutions than it is to extend larger partial solutions. We can model this phenomenon with search trees having an expanding upper portion, and contracting lower portion. When the extent and degree of contraction is bounded independently of the tree depth, the bound on $E[T(m)]$ is marginally linear in D, polynomial in n, and exponential in the extent of contraction. Thus we see that our general model allows linear performance in a situation we expect to encounter in practice, but which is not well described by previous average performance analysis. Furthermore, the constraints required to achieve this linearity are a guide towards identifying trees which do not have linear expected performance. We discuss that issue further in section 7.

Our results are stated below. Derivations are given in following sections.

LEMMA 1 (*Contracting Trees*) :

If $p^{\max} \leq 1/n$, then $E[N] < nD$.

□

LEMMA 2 (*Expanding Trees*) :

If there is an $\varepsilon > 1$ such that $p^{\min} \geq \varepsilon/n$, then

$$E[T(m)] < \frac{n(D+m-1)}{\varepsilon - 1}.$$

□

LEMMA 3 (*Contracting/Expanding Trees*) :

If there is an $I \geq 1$ and $\varepsilon > 1$ such that $p_{+I}^{\max} \leq 1/n$ and $p_{I+}^{\min} \geq \varepsilon/n$, then

$$E[T(m)] < In + \frac{n(D-I+m-1)}{\varepsilon - 1}.$$

□

LEMMA 4 (*Expanding/Contracting Trees*) :

If there is a $\delta > 1$ and $J \geq 1$ such that $p_{+(D-J)}^{\max} \leq \delta/n$, and $p_{(D-J)+}^{\max} \leq 1/n$, then

$$E[N] \leq n\left[\frac{\delta^{D-J} - 1}{\delta - 1}\right] + nJ\delta^{D-J-1}.$$

□

LEMMA 5 (*Expanding/Contracting Trees*) :

If there are $\varepsilon > 1$, and $J, C \geq 1$ such that $p_{+(D-J)}^{\min} \geq \varepsilon/n$, and $1/Cn \leq p_{(D-J)+}^{\min} \leq 1/n$ then

$$E[T(m)] < \frac{m(Cn)^{J+1} + n(D-J-1)}{\varepsilon - 1} + m(Cn)^{J+1}nJ.$$

□

4. Contracting Trees (Lemma 1)

We first derive an upper bound on $E[N]$, the expected number of nodes visited during a full search. Recalling that N_j is the random number of nodes visited at depth j during a full search, note that

$$E[N] = \sum_{j=1}^{D} E[N_j].$$

In order to calculate $E[N_j]$, observe that if $N_{j-1} = s$ for a given full search, then

$$E[N_j \mid N_{j-1} = s] \leq sp^{\max}n. \tag{1}$$

This follows because every visited node has an extension probability less than $p^{\max}$, and a successful node has n children on the average. Taking the expectation with respect to N_{j-1},

$$E[N_j] \leq E[N_{j-1}]p^{\max}n.$$

In a full search, every node at depth 1 is visited, so $E[N_1] = n$. From inequality (1),

$$E[N_2] \leq n^2p^{\max},$$

and in general,

$$E[N_j] \leq n^j(p^{\max})^{j-1}. \tag{2}$$

The expected number of nodes visited during a full search is thus bounded from above by

$$E[N] \leq \sum_{j=1}^{D} n^j(p^{\max})^{j-1}.$$

Recalling the identity [8]

$$\sum_{j=0}^{k} r^j = \frac{1 - r^{k+1}}{1 - r} \quad \text{for } r \neq 1,$$

we see that

$$E[N] \leq \begin{cases} nD & \text{if } np^{\max} = 1 \\ n\dfrac{1 - (np^{\max})^D}{1 - np^{\max}} & \text{otherwise} \end{cases} \tag{3}$$

It is also clear from equation (2) that our bound is an increasing function of $p^{\max}$.

A contracting search tree is characterized by $p^{max} \leq 1/n$. Thus Lemma 1 follows from (3) above.

LEMMA 1 : If $p^{max} \leq 1/n$, then

$$E[N] \leq nD.$$

□

Thus when the extension probabilities are all small enough, the total number of nodes visited in a full search (and hence any search) is bounded from above by nD. This bound is marginally linear in n and in D. The key reason for this linearity is the high probability of large subtrees being pruned as a result of extension failures at shallow depths.

5. Expanding Trees (Lemma 2)

We next consider an expanding search tree. Inequality (3) provides an exponentially large upper bound on $E[N]$ when $np^{max} > 1$; furthermore, this bound is tight if all extension probabilities are identical. However, the number of nodes visited in an m-solution search may be substantially smaller than that of a full search. Under the assumption that $p^{min} \geq \varepsilon/n$ for some $\varepsilon > 1$, we will next show that the number of nodes visited by an m-solution search is bounded above by a function which is marginally linear in D, in n, and in m.

The tactic we adopt in deriving this bound is to embed the search tree into an infinitely wide tree, and then analyze an m-solution search on the infinite tree from the bottom up. An m-solution search on an infinite tree with expanding nodes will always find m solutions; by analyzing a search there we avoid having to condition on whether or not m-solutions are discovered in the original search tree. The bottom up analysis is supported by two basic observations. One is that we can bound $E[T_D(m)]$ without worrying about the behavior of the search at depths other than D. The second observation is that in an m-solution search of the infinite tree, the knowledge of how many nodes have been visited at

depth j+1 allows us to bound the number of nodes which have been visited at depth j. The bound on $E[T_D(m)]$ allows us to quantify the bound on $E[T_{D-1}(m)]$; repeating this back-substitution we can quantify $E[T_j(m)]$ for each j, and sum these bounds to bound $E[T(m)]$.

Given an expanding search tree, we embed the tree into an infinite tree as follows. Conditioning on the number of children the root has (say K), the original tree consists of K *major* subtrees $B_1, B_2, \cdots, B_K$, each being rooted in a child of the tree root. We define a tree with depth D consisting of a root and a countably infinite number of children, where the first K children are roots for subtrees probabilistically identical to $B_1, ..., B_K$, respectively. All of the other children are roots for arbitrary expanding subtrees. The infinite tree does not correspond to an expansion of a given problem; rather, we use its probabilistic properties as a convenient tool. We can couple the behavior of a search on the original tree with a search on the infinite tree by requiring the evaluation outcomes in the original tree to be identical to the infinite tree's outcomes in its first K subtrees. An m-solution search on the infinite tree will terminate only by finding m "solutions". It is also clear that for every search, at every depth j, the number of nodes visited in the infinite tree at depth j is an upper bound on the number of nodes visited by the search in the finite tree. We will therefore bound $E[T(m)]$ by bounding the expected number of nodes visited in an m-solution search of the infinite tree. At the risk of abusing our notation, we will now let $T_j(m)$ denote the number of nodes visited at depth j in an m-solution search of the infinite tree; $T(m)$ is similarly redefined. Any upper bounds we derive on the infinite tree apply equally well to the finite problem tree. Our notational modification is understood to apply only to this section. For the remainder of this section, we will refer only to searches of the infinite tree; consideration of this tree is also limited to this section.

We initially consider the behavior of an m-solution search at the tree's deepest level D. Any time a leaf node is visited it is found to be a solution with probability no less than $p^{\min}$, it is otherwise found to fail. Because of the independence in leaf node evaluation outcomes, $T_D(m)$ is thus bounded

from above by an m-fold convolution of a geometric random variable $G(p^{\min})$ having success probability $p^{\min}$. Then

$$E[T_D(m)] \leq mE[G(p^{\min})] = \frac{m}{p^{\min}} \leq \frac{mn}{\varepsilon}. \tag{4}$$

We next derive upper bounds on $E[T_j(m)]$ for each j, bounding $E[T_j(m)]$ in terms of $E[T_{j+1}(m)]$. Following that, we will use (4) to quantify these upper bounds, and sum the bounds over all depths to bound $E[T(m)]$. Our discussion is facilitated by another definition. Let $S_j(m)$ be the number of *successful* nodes visited in an m-solution search. We will first bound $E[T_j(m)]$ in terms of $E[S_j(m)]$. Note that the last node visited by an m-solution search is always a successful leaf node, implying that the last node visited at depth j by a search is successful (being an ancestor of the last solution). Given that $S_j(m) = k$, it follows that $E[T_j(m)] \leq k/p^{\min}$, since the mean number of nodes at depth j visited between successful visits at that depth is no greater than $1/p^{\min}$. It follows immediately that

$$E[T_j(m)] \leq \frac{E[S_j(m)]}{p^{\min}}. \tag{5}$$

We now bound $E[S_j(m)]$ from above in terms of $E[T_{j+1}(m)]$. Call a node *activated* if its parent has been explored and is successful. Note that nodes are activated in groups, the group size being the number of children spawned by the parent. Furthermore, group sizes are independent random variables, each with mean n. If we focus our attention during a search on the activity at depth j+1, we will see a succession of groups activated, with group sizes being random variables $\{Y_k\}$. From this viewpoint, the search behavior at depth j+1 is very much like a renewal process [10], except that inter-renewal times need not have identical distributions. We call this stochastic process a *quasi-renewal* process. "Time" in this process is the number of nodes visited by the search at depth j+1, minus 1 (so that time begins at 0); a quasi-renewal occurs at "time" t if node t (the t+1st node visited) is the first node in its group to be visited (excluding node 0). Now define $S_j(m,t)$ to be the number of successful nodes visited at depth j by "time" t. As the search progresses, at any given "time" t the number of quasi-renewals

which have occurred is exactly the number of parents of depth j+1 nodes already visited, minus 1 : $S_j(m,t) - 1$. The renewal function $R(t)$ studied by renewal theory is the expected number of renewals which occur by time t. Thus for the problem at hand, $R(t) = E[S_j(m,t)] - 1$. In [10,p.275] it is shown that the renewal function for a quasi-renewal process with NBUE inter-quasi-renewal times each having mean n is bounded from above by the renewal function $R(t) = t/n$ of a Poisson process with mean inter-renewal time n. Since the $\{Y_k\}$'s are NBUE by assumption, it follows that

$$E[S_j(m,t)] \leq \frac{t}{n} + 1.$$

"Time" stops at value $T_{j+1}(m)$, when the search terminates. The inequality above then implies that

$$E[S_j(m)] \leq \frac{E[T_{j+1}(m)]}{n} + 1.$$

By substituting this bound into inequality (5) and noting that $p^{\min} \geq \varepsilon/n$, we find that

$$E[T_j(m)] \leq \frac{E[T_{j+1}(m)]}{\varepsilon} + \frac{n}{\varepsilon}, \tag{6}$$

which establishes our goal of a bound on $E[T_j(m)]$ in terms of $E[T_{j+1}(m)]$.

Inequality (6) gives us a sequence of bounds as we vary j from D to 2. We quantify these inequalities by employing the bound on $E[T_D(m)]$ given by (4), and then repeatedly apply inequality (6) to obtain bounds on $E[T_j(m)]$ for $j = D-1, D-2, \cdots, 2$. We know that

$$E[T_D(m)] \leq \frac{nm}{\varepsilon}.$$

Substituting this bound for $E[T_D(m)]$ into inequality (6) we obtain

$$E[T_{D-1}(m)] \leq \frac{nm}{\varepsilon^2} + \frac{n}{\varepsilon}.$$

Repeating this process, we find that for $0 \leq k \leq D-1$

$$E[T_{D-k}(m)] \leq \frac{nm}{\varepsilon^{k+1}} + n\sum_{j=1}^{k}\left\lceil\frac{1}{\varepsilon^j}\right\rceil$$

$$= \frac{nm}{\varepsilon^{k+1}} + \frac{n\left\lceil 1 - \frac{1}{\varepsilon^k}\right\rceil}{\varepsilon - 1}. \qquad (7)$$

We can then bound $E[T(m)]$ by summing the different depths' upper bounds given by (7).

$$E[T(m)] \leq \sum_{k=0}^{D-1}\left\lceil\frac{nm}{\varepsilon^{k+1}} + \frac{n\left\lceil 1 - \frac{1}{\varepsilon^k}\right\rceil}{\varepsilon - 1}\right\rceil$$

$$= \frac{1}{\varepsilon-1}\left[nm(1 - \frac{1}{\varepsilon^D}) + n(D-1) - n\left\lceil\frac{\varepsilon}{\varepsilon-1}\right\rceil\left[1 - \frac{1}{\varepsilon^D}\right]\right].$$

A less strict, but less involved bound follows immediately from the fact that $\varepsilon > 1$:

LEMMA 2 : If there is an $\varepsilon > 1$ such that $p^{\min} \geq \varepsilon/n$, then

$$E[T(m)] < \frac{n(D + m - 1)}{\varepsilon - 1}. \qquad (8)$$

□

This bound is marginally linear in n, D, and m. Furthermore, its change with respect to m is interesting. Doubling m increases the bound by less than a factor of two. This supports our intuitive understanding that after one solution is found, the additional cost of finding a second solution is (in expectation) less. The bound given in (8) is very sensitive to ε when ε is close to 1. Yet for $\varepsilon \geq 1.1$, $E[T(m)]$ is no greater than $10n(D + m - 1)$, which seems quite reasonable.

6. Contracting/Expanding Trees (Lemma 3)

Lemmas 1 and 2 give us means of bounding the expected performance of an m-solution search on contracting, and on expanding trees. We next employ these bounds to derive bounds for trees which contract at shallow depths, and expand at deep depths. As expected, we obtain these bounds by considering the expanding and contracting regions separately, and combine the results for an overall bound. We derive a bound on $E[T(m)]$ which is linear in n, in D, and in m.

We suppose there is a depth I such that $p_{+I}^{\max} \leq 1/n$; and $p_{I+}^{\min} \geq \varepsilon/n$ for some $\varepsilon > 1$. Consider the search tree truncated at depth I; Lemma 1 states that the expected number of nodes visited by a full search of the truncated tree is no greater than nI, and equation (2) shows that the expected number of depth I nodes visited by a full search is bounded by n. It follows that the expected number of successful depth I nodes found in a full search, and hence an m-solution search, is no greater than 1. We can bound the expected number of nodes visited by an m-solution search at depths I+1, $\cdots$,D by considering the subtrees rooted in successful nodes at depth I. Each such subtree has depth $D - I$, and extension probabilities (excluding the subtree root) which all exceed $1/n$. Now consider a *total subtree search* of depths I+1, $\cdots$,D, which conducts an m-solution search of every subtree rooted in a successful node at depth I. Clearly the expected number of nodes visited by such a search is greater than the expected number of visited nodes in a usual m-solution search: the usual m-solution search will stop with the first m solutions, while the total subtree search will continue on to explore all subtrees with successful roots at depth I. The expected number of successful nodes at depth I is no greater than 1, so that the expected number of nodes at depths greater than I visited by a total subtree search is bounded from above by

$$\frac{n(D - I + m - 1)}{\varepsilon - 1}.$$

This bound follows from the application of Lemma 2 to one subtree rooted in depth I. We have already bounded the number of nodes visited at depths 1, $\cdots$,I by nI. The expected number of nodes

visited in an m-solution search is therefore bounded as described in Lemma 3.

LEMMA 3 : If there is an $I \geq 1$ and $\varepsilon > 1$ such that $p_{+I}^{\max} \leq 1/n$ and $p_{I+}^{\min} \geq \varepsilon/n$, then

$$E[T(m)] \leq In + \frac{n(D - I + m - 1)}{\varepsilon - 1}. \tag{9}$$

□

This bound has the same marginal properties in n, D, and m as does the bound given by Lemma 2.

7. Expanding/Contracting Trees (Lemmas 4,5)

We earlier gave an example of a tree which expanded in all but the last depth, and then contracted completely at the leaf nodes. This tree has the worst possible performance because it searches all nodes, but finds no solutions. This tree falls in the class of trees we next consider, trees which expand in upper levels, and contract in lower levels. However, we are able to show that if the degree and extent of contraction is bounded, then we achieve marginal linear performance in the depth of the tree. Our bound is more sensitive to n: it increases as a polynomial in n and in the degree of contraction. This bound is exponential in the extent of the contracting region.

We suppose there is a depth $D - J$ such that $p_{+(D-J)}^{\min} \geq \varepsilon/n$ for some $\varepsilon > 1$, and $1/n \geq p_{(D-J)+}^{\max} \geq 1/Cn$ for some $C \geq 1$. The parameter J describes the extent of the contracting region (J depth levels), and C describes the degree of the contraction. We derive two bounds for this case. Our bound on $E[N]$ is useful when the extension probabilities are small or when the expanding portion of the tree is limited; the other bound is useful if the contracting region is small.

To bound $E[N]$, we make the additional assumption that $p^{\max} \leq \delta/n$. By (3), the expected number of nodes visited at depths $1, \cdots, D-J$ by a full search is bounded from above by

$$n\left[\frac{\delta^{D-J} - 1}{\delta - 1}\right].$$

As a consequence of (2), the expected number of successful nodes at depth $D-J$ is bounded from

above by δ^{D-J-1}. By Lemma 1, the expected number of nodes visited in a subtree rooted in a successful depth $D-J$ node is bounded from above by nJ. Lemma 4 combines these observations.

LEMMA 4 : If there is a $\delta > 1$ and $J \geq 1$ such that $p^{\max}_{+(D-J)} \leq \delta/n$, and $p^{\max}_{(D-J)+} \leq 1/n$, then

$$E[N] \leq n\left[\frac{\delta^{D-J}-1}{\delta-1}\right] + nJ\delta^{D-J-1}. \tag{10}$$

□

The key parameters in this inequality are δ and J. When δ is small, say $\delta < 2$, the exponential growth of (10) in $D-J$ is slow. If $D-J$ is small, then the troublesome expanding portion of the tree is limited, and (10) may yield a reasonably small bound. If neither of these cases is satisfied, we should consider another bound on $E[T(m)]$, which we derive next.

The assumption that $p^{\min}_{(D-J)+} \geq 1/Cn$ bounds the degree to which the search tree contracts. We can use this restriction to bound $E[T(m)]$. The first step is to look at a subtree rooted in a visited depth $D-J$ node, and find a lower bound on the probability p_S of finding a solution in that subtree. One potential path to discovering a solution in this subtree occurs if the first $J+1$ successive visits in a depth-first search of the subtree (we include the root here) each discover successful nodes. The last successful node is the full solution. The probability of this occurrence is at least

$$p_L = \frac{1}{(Cn)^{J+1}} \leq p_S.$$

This bound holds for every subtree (and hence every probability of finding a solution in a subtree). We will say that a depth $D-J$ node is *ultimately successful*, or u-successful, if it is an ancestor of a successful complete solution. p_L is a lower bound on the probability of a visited depth $D-J$ node being u-successful. The behavior of a 1-solution search at depths $1,2,\cdots,D-J$ is probabilistically identical to the behavior of a 1-solution search on a modified tree having depth $D-J$ and the same extension probabilities as the original search tree, except that depth $D-J$ nodes' extension probabilities are replaced by their u-success probabilities. Since a subtree rooted in depth $D-J$ may contain multiple solutions, an

m-solution search of the original tree will visit fewer nodes at depths $1, \cdots, D-J$ than will an m-solution search of the modified tree. The arguments presented in section 5 establish that the mean number of nodes visited at depth $D-J$ by an m-solution search of the modified tree is bounded from above by $\frac{m}{p_L} = m(Cn)^{J+1}$. Then the same type of analysis which leads to equation (8) shows that

$$\sum_{j=1}^{D-J} E[T_j(m)] < \frac{m(Cn)^{J+1} + n(D-J-1)}{\varepsilon - 1}. \tag{11}$$

Reconsidering the original depth D tree, we use Lemma 1 to see that the expected number of nodes visited by a total search of a subtree rooted in depth $D-J$ is bounded from above by nJ. $m(Cn)^{J+1}nJ$ is thus an upper bound on the expected number of nodes visited at depths $D-J+1,\ldots, D$ in an m-solution search of the original tree. Summing this bound with the one given by (11), we have

LEMMA 5 : If there are $\varepsilon > 1$, and J, $C \geq 1$ such that $p^{\max}_{+(D-J)} \geq \varepsilon/n$, and $1/Cn \leq p^{\min}_{(D-J)+} \leq 1/n$ then

$$E[T(m)] < \frac{m(Cn)^{J+1} + n(D-J-1)}{\varepsilon - 1} + m(Cn)^{J+1}nJ.$$

□

This bound is reasonably small only if J is small. We can expect marginal linear performance in the depth of the tree, provided there are depth independent bounds on the number of tree levels which contract, and the degree to which nodes in that region contract. Lemma 5 also hints at the relative sensitivity of performance to changes in the different model parameters. The bound grows as a polynomial in n and in C; it grows exponentially in J.

Reconsider the example of the tree which visited all nodes, and found no solutions. If we slightly change the example so that a leaf node has some fixed probability $p > 0$ of being a solution, then Lemma 5 gives us marginal linear performance in the depth of the tree (supporting empirical data presented in [StSi85].) This is more of an indictment of asymptotic results than it is an indication of good performance. If J is fixed and quite large, Lemma 5 can still give us linear performance in D.

That does not mean that we can expect to find a solution quickly.

The example highlighting exponential complexity hinged on having extension probabilities equal to zero. There exist trees with all non-zero extension probabilities which have exponential performance asymptotically. It is not difficult to see (or prove) that as the depth of a contracting tree increases, the probability of discovering a solution in that tree approaches zero. Now consider a tree which expands in the upper half of its levels and contracts in the lower half. In the limit of increasing tree depth, a full search of the tree's upper half occurs because the search is always pruned in the lower half before reaching a solution. Since the upper half is expanding, the number of nodes visited there increases exponentially in the tree depth. This example re-illustrates the need for extending past work in average performance analysis of backtracking. Using uniform probability structures, previous performance models conclude that all 1-solution searches have expected linear complexity. The example above shows that under a general model, there are 1-solution searches with super-linear complexity. Our general model is more selective in its identification of trees leading to low complexity, and specifically excludes the example outlined above from the class of searches with linear complexity.

8. Summary

Previous analysis of the average case performance of 1-solution backtracking searches have assumed uniform probability structures, and have subsequently shown that 1-solution backtracking always has complexity linear in the depth of the search tree. However, using a more general probability model, it is possible to construct 1-solution backtracking searches which have expected super-linear complexity. This paper investigates the implications of assuming a general probability structure for the average case analysis. Using the notions of "expanding" and "contracting" nodes, we have derived expected linear complexity bounds on m-solution searches of trees with all nodes contracting, and trees with all nodes expanding. We present linear bounds on trees which have an upper portion contracting with the lower portion expanding. We also derive bounds on trees with an expanding upper portion

and a contracting lower portion. When the size of the contracting region is bounded by a constant, and the degree of contraction is bounded by a constant, this last bound is marginally linear in the depth of the search tree. By extending previous work in the average complexity of backtracking, we have both provided further assurance of good expected performance under appropriate conditions, and indirectly shown where classes of searches with high complexity lie.

Acknowledgements

Use of the the terms "expanding" and "contracting" are due to a discussion with Harold Stone. I also thank Shahid Bokhari and Bob Voigt for suggestions regarding this paper's presentation.

References

[1] J. R. Bitner, E. M. Reingold, Backtrack Programming Techniques, *Communications of the ACM*, 18, 11, (1975) 651-655.

[2] C. A. Brown, P. W. Purdom, An Average Time Analysis of Backtracking, *SIAM Journal on Computing*, 10, 3, (1981), pp. 583-593.

[3] C. A. Brown, P. W. Purdom, An Empirical Comparison of Backtracking Algorithms, *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-4, 3, (1982), 309-316.

[4] T. Harris, *The Theory of Branching Processes*, Springer, Berlin, 1963.

[5] E. Horowitz, S. Sahni, *Fundamentals of Computer Algorithms*, Computer Science Press, Rockville, MD, 1978.

[6] R. M. Karp, The Probabilistic Analysis of some Combinatorial Search Algorithms, *Algorithms and Complexity*, J.F. Traub, ed., Academic Press, New York, 1976, 1-19.

[7] R. M. Karp, J. Pearl, Searching for an Optimal Path in a Tree with Random Costs, *Artificial Intelligence*, 21, 1, (1983), pp. 99-117.

[8] D. E. Knuth, *The Art of Computer Programming, Vol. 1*, Addison-Wesley, Menlo Park, CA, 1973.

[9] P. W. Purdom, Jr., Search Rearrangement Backtracking and Polynomial Average Time, *Artificial Intelligence*, 21, 1, (1983), 117-133.

[10] S. Ross, *Stochastic Processes*, Wiley, New York, 1983.

[11] D. R. Smith, Random Trees and the Analysis of Branch and Bound Procedures, *Journal of the ACM*, 31, 1, (1984), pp. 163-188.

[12] H. S. Stone, P. Sipala, The Average Complexity of Depth-First Search with Backtracking and Cutoff, *IBM Journal of Research and Development*, 30, 3, (1986), 242-258.

[13] B. W. Wah, C. F. Yu, Stochastic Modeling of Branch-and-Bound with Best-First Search, *IEEE Trans. on Software Engineering*, SE-11, 9, (1985), pp. 922-934.

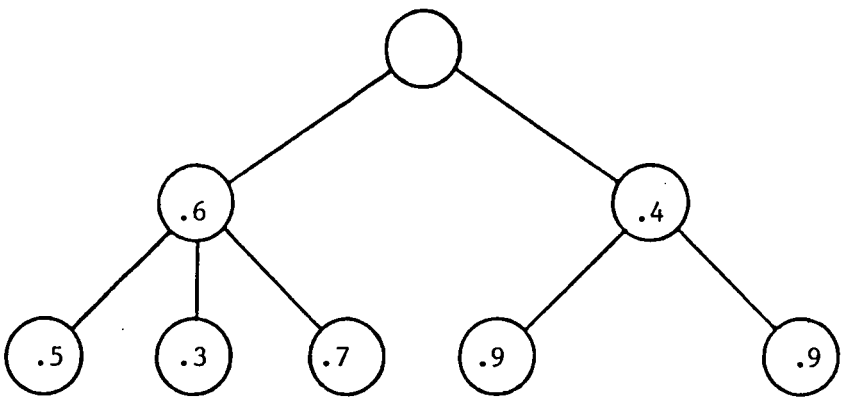

Fig. 1: Search Tree with Extension Probabilities

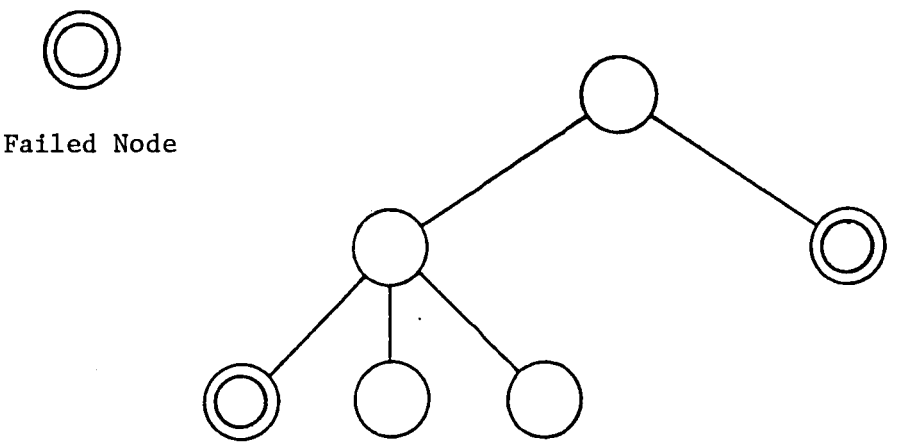

Fig. 2: Realization of Full Search, Two Solutions

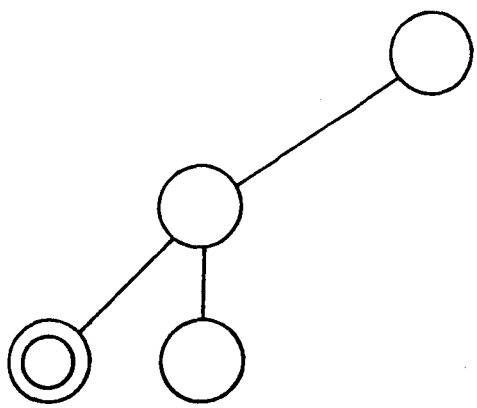

Fig. 3: Realization of 1-Solution Search

Standard Bibliographic Page

1. Report No. NASA CR-178162 ICASE Report No. 86-55	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle EXPECTED PERFORMANCE OF m-SOLUTION BACKTRACKING		5. Report Date August 1986 6. Performing Organization Code
7. Author(s) David M. Nicol		8. Performing Organization Report No. 86-55
9. Performing Organization Name and Address Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225		10. Work Unit No. 11. Contract or Grant No. NAS1-18107
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		13. Type of Report and Period Covered Contractor Report 14. Sponsoring Agency Code 505-31-83-01

15. Supplementary Notes

Langley Technical Monitor:
J. C. South

Submitted to SIAM Journal of Computing

Final Report

16. Abstract

This paper derives upper bounds on the expected number of search tree nodes visited during an m-solution backtracking search, a search which terminates after some preselected number m problem solutions are found. The search behavior is assumed to have a general probabilistic structure. Our results are stated in terms of node expansion and contraction. A visited search tree node is said to be expanding if the mean number of its children visited by the search exceeds 1 and is contracting otherwise. We show that if every node expands, or if every node contracts, then the number of search tree nodes visited by a search has an upper bound which is linear in the depth of the tree, in the mean number of children a node has, and in the number of solutions sought. We also derive bounds linear in the depth of the tree in some situations where an upper portion of the tree contracts (expands), while the lower portion expands (contracts). While previous analysises of 1-solution backtracking have concluded that the expected performance is always linear in the tree depth, our model allows super-linear expected performance. By generalizing previous work in the expected behavior of backtracking, we are better able to identify classes of trees which can be searched in linear expected time.

17. Key Words (Suggested by Authors(s))	18. Distribution Statement
random search, backtracking, depth-first search	61 - Computer Programming and Software Unclassified - unlimited

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	25	A02

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA Langley Form 63 (June 1985)

End of Document