stop(top). In such a situation, the element top has to be deleted from the stack and more operations are required to generate the next combination. When $k > \text{top} > 2$, one can show that the probability for a specific value of top that $a[\text{top}] = \text{stop(top)}$ is $\alpha(\text{top} + 1)/\alpha(\text{top})$, which reduces to $(k - \text{top} + 1)/(n - \text{top})$. Hence, when k is small compared to n, it is very unlikely that the next combination is generated by using the theoretical maximum number of operations.

If k is very small compared to n, then $P(1)$, the probability that a combination is generated by changing only $a[1]$, is approximately 1. In this case, almost all the combinations are generated by the single statement $a[1] := a[1] - 1$. It is doubtful that any combination algorithm would require less work than this to generate a combination.

Received 8/79; revised 10/81; accepted 11/81

References

1. Bitner, J.R., Ehrlich, G., and Reingold, E.M. Efficient generation of the binary reflected Gray code and its applications. *Comm. ACM, 19*, 9 (Sept. 1976), 517–521.
2. Ehrlich, G. Algorithm 466, Four combinatorial algorithms. *Comm. ACM, 16*, 11 (Nov. 1973), 690–691.
3. Ehrlich, G. Loopless algorithms for generating permutations, combinations and other combinatorial configurations. *J. ACM, 20*, (1973), 500–513.
4. Ives, F.M. Permutation enumeration: Four new permutation algorithms. *Comm. ACM, 19*, 2 (Feb. 1976), 68–72.
5. Reingold, E.M., Nievergelt, J., and Deo, N. *Combinatorial Algorithms: Theory and Practice.* Prentice-Hall, New Jersey, 1977.

Programming Techniques and Data Structures

M. Douglas McIlroy*
Editor

The Solution for the Branching Factor of the Alpha–Beta Pruning Algorithm and its Optimality

Judea Pearl
University of California, Los Angeles

This paper analyzes $N_{n,d}$, the average number of terminal nodes examined by the α-β pruning algorithm in a uniform game tree of degree n and depth d for which the terminal values are drawn at random from a continuous distribution. It is shown that increasing the search depth by one extra step would increase $N_{n,d}$ by a factor (called the *branching factor*) $\mathcal{R}_{\alpha\text{-}\beta}(n) = \xi_n/1 - \xi_n \approx n^{3/4}$ where ξ_n is the positive root of $x^n + x - 1 = 0$. This implies that for a given search time allotment, the α-β pruning allows the search depth to be increased by a factor $\approx 4/3$ over that of an exhaustive minimax search. Moreover, since the quantity $(\xi_n/1 - \xi_n)^d$ has been identified as an absolute lower bound for the average complexity of all game searching algorithms, the equality $\mathcal{R}_{\alpha\text{-}\beta}(n) = \xi_n/1 - \xi_n$ now renders α-β *asymptotically optimal.*

CR Categories and Subject Descriptors: I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods and Search—*graph and tree search strategies, heuristic methods*; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*complexity of proof procedures, sorting and searching*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*graph algorithms, trees*

General Term: Algorithms

Additional Key Words and Phrases: alpha-beta search, game searching, games, minimax algorithms, branch and bound search, average case analysis

This work was supported in part by the National Science Foundation Grants MCS 78-07468 and MCS 78-18924

An early version of this paper was presented at the *8th International Conference on Automata Languages and Algorithms*, Acre, Israel, July 3–17, 1981.

* Former editor of Programming Techniques and Data Structures, of which Ellis Horowitz is the current editor.

Author's Present Address: Judea Pearl, Cognitive Systems Laboratory, Departments of Computer Science and Engineering Systems, University of California, Los Angeles, Los Angeles, CA 90024.

1. Introduction

1.1. Informal Description of the α–β Procedure

The α–β pruning algorithm is the most commonly used procedure in game playing applications, where it serves to speed up game searching without loss of information. The algorithm determines the minimax value of the root of a game tree by traversing the tree in a predetermined order, for example, from left to right, skipping all those nodes that can no longer influence the minimax value of the root.

The method is demonstrated in Fig. 1 which shows a binary game tree of depth $d = 4$ with nodes at maximizing levels (called MAX nodes) and at minimizing levels (called MIN nodes) represented by squares and circles, respectively. The numbers inside the terminal squares represent evaluations of the game positions at the frontier of the search tree, while those at higher levels are the minimax values computed by the α–β procedure. The heavy branches represent the search tree actually generated by the α–β procedure as it traverses the game tree from left to right. Nodes not on that search tree are skipped (or "cutoff") by α–β, as they cannot provide useful information.

The rationale for node skipping can be explained by examining the nodes labeled A, B, and C, in Fig. 1. The purpose of exploring node B has been to find out if the value of A can be reduced below 10, which is the value established for A's leftmost son. However, the fact that one of B's sons has already attained the value 14 and that B is a MAX node imply that the value of B must be greater than 14, regardless of any information that C may provide. Therefore, any exploration of C cannot alter the fact that the value of A is exactly 10, so C can be cut off from the search. A precise formulation of the α–β algorithm and its cutoff conditions can be found in [3].

Clearly, the efficiency of this search method depends on the order of the terminal values. For the values shown in Fig. 1, only 7 terminal nodes are examined by a left-to-right search, whereas all 16 terminal nodes would have to be examined by a right-to-left search. In complex games, the difference between the best case and the worst case can be quite substantial, amounting to a factor of 2 in the depth of the look-ahead tree that a given computer system can afford to explore. This disparity warrants analysis of the *average* performance of α–β under the assumption that the terminal values are randomly ordered.

1.2 Previous Analytical Results

Although experiments show that the exponential growth of game tree searching is slowed significantly by the α–β pruning algorithm, quantitative analyses of its effectiveness have been frustrated for over a decade. One reason for this concern has been to determine whether the average performance of the α–β algorithm is optimal over that of other game searching procedures.

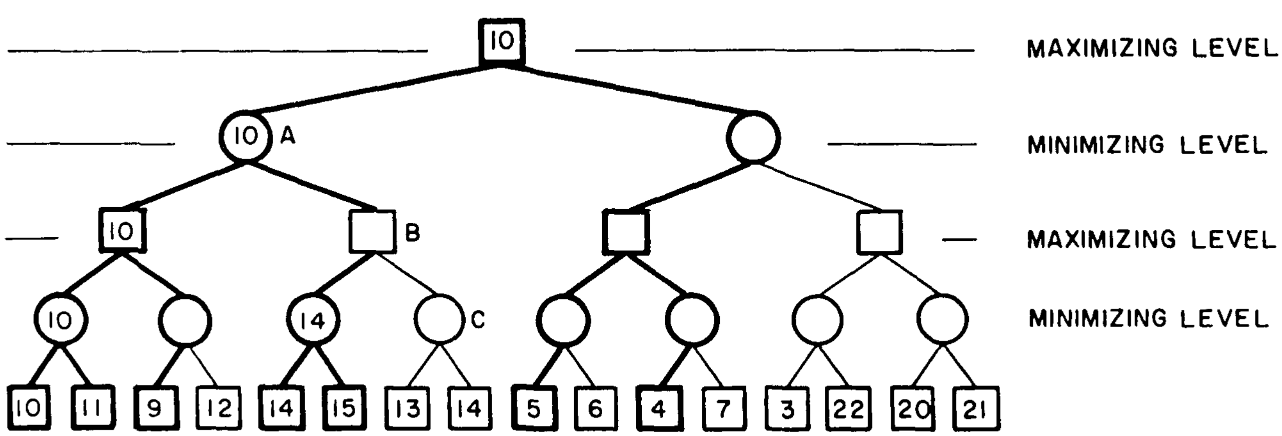
The model most frequently used for evaluating the performance of game searching methods consists of a uniform tree of depth d and degree n, where the terminal positions are assigned random, independent, and identically distributed values. $N_{n,d}$, the average number of terminal nodes examined during the search, has become a standard yardstick for the complexity of the search method. Additionally, the significant parameter for very deep trees is the *branching factor*

$$\mathscr{R}_{\alpha-\beta} = \lim_{d\to\infty} (N_{n,d})^{1/d}$$

which measures the effective number of branches actually explored by α–β from a typical node of the search tree.

Slagle and Dixon [8] showed that the number of terminal nodes examined by α–β must be at least $n^{\lfloor d/2 \rfloor} + n^{\lceil d/2 \rceil} - 1$ but may, in the worst case, reach the entire set of n^d terminal nodes. The analysis of expected performance using uniform trees with random terminal values began with Fuller, Gaschnig, and Gillogly [2] who obtained formulas by which the average number of terminal examinations $N_{n,d}$ can be computed. Unfortunately, the formula would not facilitate asymptotic anal-

Fig. 1. A Binary Game Tree of Depth 4 Traversed from Left-to-Right by the Alpha–Beta Procedure.



ysis; simulation studies led to the estimate $\mathcal{R}_{\alpha-\beta} \approx (n)^{0.72}$.

Knuth and Moore [3] analyzed a less powerful but simpler version of the α–β procedure by ignoring deep cutoffs. They showed that the branching factor of this simplified model is $O(n/\log n)$ and speculated that the inclusion of deep cutoffs would not alter this behavior substantially. A more recent study by Baudet [1] confirmed this conjecture by deriving an integral formula for $N_{n,d}$ (deep cutoffs included), from which the branching factor can be estimated. In particular, Baudet shows that $\mathcal{R}_{\alpha-\beta}$ is bounded by $\xi_n/1 - \xi_n \le \mathcal{R}_{\alpha-\beta} \le M_n^{1/2}$, where ξ_n is the positive root of $x^n + x - 1 = 0$ and M_n is the maximal value of the polynomial $P(x) = (1 - x^n/1 - x)$ $[1 - (1 - x^n)^n/x^n]$ in the range $0 \le x \le 1$. Pearl [5] has shown both that $\xi_n/1 - \xi_n$ lower bounds the branching factor of every directional game searching algorithm and that an algorithm exists (called SCOUT) that actually achieves this bound. Tarsi [10] has very recently shown that $\xi_n/1 - \xi_n$ also lower bounds the branching factor of nondirectional algorithms. Thus, the enigma of whether α–β is optimal remains contingent upon determining the exact magnitude of $\mathcal{R}_{\alpha-\beta}$ within the range delineated by Baudet.

This paper now shows that the branching factor of α–β indeed coincides with the lower bound $\xi_n/1 - \xi_n$, thus establishing the asymptotic optimality of α–β over the class of all game searching algorithms.

2. Analysis

2.1 An Integral Formula for $N_{n,d}$

Our starting point will be an examination of the conditions under which an arbitrary node J is generated by the α–β algorithm. If all terminal values to the left of J are given, one can perform a simple test to determine whether or not J will be generated. For a MAX node J, form the path leading from the root to J, and define the following quantities:

$A(J)$ = the highest minimax value among all left-siblings of odd ancestors of J

$B(J)$ = the lowest minimax value among all left-siblings of even ancestors of J

J will be generated by α–β if and only if

$$A(J) < B(J).$$

The same criterion holds when J is a MIN node, except that $A(J)$ is computed over even ancestors and $B(J)$ over odd ancestors of J. A special definition is required to include so-called *critical* nodes for which the corresponding sets of left-siblings are empty [7].

The reader can easily verify that in Fig. 1 all nodes generated satisfy the criterion above while all those satisfying $A(J) \ge B(J)$ can provide no information beyond that which has already been gathered by the search and will be cut off. For example, for the rightmost leaf node we have:

$$A(J) = \max[\min(3, 22), 10] = 10$$

$$B(J) = \min\{20, \max[\min(5, 6), \min(4, 7)]\} = 5$$

and since $A(J) > B(J)$, it is not generated by the α–β search.

The criterion above was first derived by Fuller et al. [2] and is a useful tool for computing $N_{n,d}$, the average number of terminal nodes examined by α–β. One need only compute the probability $P[A(J) < B(J)]$ for every node J, then sum these probabilities over all terminal nodes.

$$N_{n,d} = \sum_{J \text{ terminal}} P[A(J) < B(J)]$$

This procedure may seem like a major undertaking. Fortunately, when the terminal values are drawn independently from a common distribution function $f_0(x) = P[V_0 \le x]$, very simple propagation rules govern the distributions of the minimax values at higher levels of the tree. For example, if V_k stands for the minimax value of a MIN node at level k of the tree, then its distribution f_k is related to that of its direct descendents by

$$f_k(x) = 1 - [1 - f_{k-1}(x)]^n$$

and to that of its grandsons by

$$f_k(x) = 1 - \{1 - [f_{k-2}(x)]^n\}^n$$

From these recursions one can compute the distributions $F_{A(J)}(x)$ and $F_{B(J)}(x)$ of the random variables $A(J)$ and $B(J)$ for any terminal node J. Moreover, since $A(J)$ and $B(J)$ are independent and continuous (for noncritical nodes) we have

$$P[A(J) < B(J)] = \int_{x=-\infty}^{\infty} F_{A(J)}(x) F'_{B(J)}(x)\, dx$$

and $N_{n,d}$ becomes

$$N_{n,d} = \int_{x=-\infty}^{\infty} \left[\sum_{J \text{ terminal}} F_{A(J)}(x) F'_{B(J)}(x) \right] dx + n^{\lceil d/2 \rceil} + n^{\lfloor d/2 \rfloor} - 1$$

where the terms added to the integral represent the number of critical nodes, all of which are examined. The summation inside the integral can be performed using the recursion relations above (see Roizen [7]) and lead to the following theorem

THEOREM 1. Let $f_0(x) = x$, and, for $i = 1, 2, \ldots,$ define

$$f_i(x) = 1 - \{1 - [f_{i-1}(x)]^n\}^n$$

$$r_i(x) = \frac{1 - [f_{i-1}(x)]^n}{1 - f_{i-1}(x)}$$

$$s_i(x) = \frac{f_i(x)}{[f_{i-1}(x)]^n}$$

$$R_i(x) = r_1(x) \times \cdots \times r_{\lceil i/2 \rceil}(x)$$

$$S_i(x) = s_1(x) \times \cdots \times s_{\lfloor i/2 \rfloor}(x)$$

The average number $N_{n,d}$ of terminal nodes examined by the α–β pruning algorithm in a uniform game tree of degree n and depth d for which the bottom values are drawn from a continuous distribution is given by

$$N_{n,d} = n^{\lfloor d/2 \rfloor} + \int_0^1 R'_d(t) S_d(t)\, dt \qquad (1) \quad \square$$

An identical expression for $N_{n,d}$ was first derived by Baudet ([1], Theorem 4.2) starting with discrete terminal values and progressively refining their quantization levels.

2.2 Evaluation of $\mathcal{R}_{\alpha-\beta}$

The difficulty in estimating the integral in Eq. (1) stems from the recursive nature of $f_i(x)$ which tends to obscure the behavior of the integrand. We circumvent this difficulty by substituting for $f_0(x)$ another function $\phi(x)$ which makes the regularity associated with each successive iteration more transparent.

The value of the integral in Eq. (1) does not depend on the exact nature of $f_0(x)$ as long as it is monotone from some interval $[a, b]$ onto the range $[0, 1]$. This is evident by noting that by substituting $f_0(x) = \phi(x)$ the integral becomes

$$\int_{x=a}^{b} \frac{dR_d[\phi(x)]}{dx} S_d[\phi(x)]\, dx = \int_{\phi=0}^{1} \frac{dR_d(\phi)}{d\phi} S_d(\phi)\, d\phi$$

which is identical to that in Eq. (1). This invariance reflects the fact that the search procedure depends only on the relative order of the d^n terminal values, not on their magnitudes, and since any continuous distribution of the terminal values generates all ranking permutations with equal probabilities, $N_{n,d}$ will not be affected by the *shape* of that distribution. Consequently, $f_0(x)$ which represents the terminal values' distribution, may assume an arbitrary form, subject to the usual constraints imposed on continuous distributions.

A convenient choice for the distribution $f_0(x)$ would be a characteristic function $\phi(x)$ that would render the distributions of the minimax value of every node in the tree identical in shape. Such a characteristic distribution indeed exists [6] and satisfies the functional equation

$$\phi(x) = g[\phi(ax)] \qquad (2)$$

where

$$g(\phi) = 1 - (1 - \phi^n)^n \qquad (3)$$

and a is a real-valued parameter to be determined by the requirement that Eq. (2) possess a nontrivial solution for $\phi(x)$. This choice of $\phi(x)$ renders the functions $\{f_i(x)\}$ in Theorem 1 identical in shape, save for a scale factor. Accordingly, we can write

$$f_i(x) = \phi\,(x/a^i) \qquad (4)$$

$$r_i(x) = r\,(x/a^{i-1}) \qquad (5)$$

$$s_i(x) = s\,(x/a^{i-1}) \qquad (6)$$

where

$$r(x) = \frac{1 - [\phi\,(x)]^n}{1 - \phi\,(x)} \qquad (7)$$

and

$$s(x) = \frac{1 - \{1 - [\phi\,(x)]^n\}^n}{[\phi\,(x)]^n} \qquad (8)$$

Equation (2), known as the Poincaré Equation [4], has a nontrivial solution $\phi(x)$ with the following properties [6]:

(*i*) $\phi(0) = \xi_n$ where ξ_n is the root of
$$x^n + x - 1 = 0 \qquad (9)$$

(*ii*) $$a = \frac{1}{g'(\xi_n)} = \left[\frac{\xi_n}{n(1 - \xi_n)}\right]^2 < 1 \qquad (10)$$

(*iii*) $\phi'(0)$ can be chosen arbitrarily, for example, $\phi'(0) = 1$

(*iv*) $$x(\phi) = \lim_{k\to\infty} a^k[g^{-k}(\phi) - \xi_n]$$

$$\phi(x) \underset{x\to\infty}{\approx} 1 - (n)^{-n/n-1} \exp[-(x)^{-\ln(n)/\ln(a)}]$$

$$\phi(x) \underset{x\to-\infty}{\approx} (n)^{-1/n-1} \exp[-(x)^{-\ln(n)/\ln(a)}]$$

However, only properties (9) and (10) will play a role in our analysis. Most significantly, parameter a, which is an implicit function of n, remains lower than 1 for all n.

Substituting Eqs. (4), (5), and (6) into Eq. (1) and considering, without loss of generality, the case where d is an even integer, $d = 2h$, we obtain

$$N_{n,d} = n^h + \int_{x=-\infty}^{\infty} \pi_h(x)\left(\sum_{i=1}^{h} \frac{r'_i(x)}{r_i(x)}\right) dx \qquad (11)$$

where

$$\pi_h(x) = \prod_{i=0}^{h-1} p(x/a^i), \qquad (12)$$

$$p(x) = r(x)s(x) = P[\phi(x)], \qquad (13)$$

and

$$P(\phi) = \frac{1 - \phi^n}{1 - \phi}\,\frac{1 - (1 - \phi^n)^n}{\phi^n} \qquad (14)$$

Using Eqs. (5) and (7), it can be easily shown that $r'_i(x)/r_i(x)$ satisfies

$$\frac{r'_i(x)}{r_i(x)} \leq n\,\phi'(x/a^{i-1})1/a^{i-1} \qquad (15)$$

and consequently, Eq. (11) becomes

$$N_{n,d} \leq n^h + n \int_{-\infty}^{\infty} \pi_h(x) \cdot \left[\sum_{i=1}^{h} \phi'(x/a^{i-1})1/a^{i-1}\right] dx \qquad (16)$$

We now wish to bound the term $\pi_h(x)$ from above.

An examination of $p(x) = P[\phi(x)]$ [Eqs. (13) and (14)] reveals that $p(x)$ is unimodal in x, $p(0) = [\xi_n/1 - \xi_n]^2$, and that $p(x)$ lies above the asymptotes $p(-\infty) = p(+\infty) = n$. Moreover, the maximum of $P(\phi)$ occurs below $\phi = \xi_n$ and, consequently, $p(x)$ attains its maximum M_n below $x = 0$.

At this point, were we to use the bound $\pi_h(x) \le M_n^h$ in (16), it would result in $N_{n,d} < n^h + nhM_n^h$ and lead to Baudet's bound $\mathcal{R}_{\alpha-\beta} \le M_n^{1/2}$. Instead, a tighter bound can be established by exploiting the unique relationships between the factors of $\pi_h(x)$.

LEMMA 1. Let $x_0 < 0$ be the unique negative solution of $p(x_0) = p(0)$. $\pi_h(x)$ attains its maximal value in the range $a^{h-1}x_0 \le x \le 0$.

PROOF. Since $p(x)$ is unimodal we have $p(x) < p(0)$ and $p'(x) > 0$ for all $x < x_0$. Consequently, for all $x < x_0$, any decrease in the magnitude of $|x|$ would result in increasing $p(x)$, that is, $p(cx) > p(x)$ for all $0 \le c < 1$. Now consider $\pi_h(ax)$.

$$\pi_h(ax) = p(x/a^{h-2})p(x/a^{h-3}) \cdots p(x)p(ax)$$
$$= \pi_h(x)p(ax)/p(x/a^{h-1});$$

for all x' satisfying $x'/a^{h-1} < x_0$ we must have $p(ax') > p(x'/a^{h-1})$ (using $c = a^h < 1$) and $\pi_h(ax') > \pi_h(x')$, implying that $\pi_h(x')$ could not be maximal. Consequently, for $\pi_h(x')$ to be maximal, x' must be in the range $x_0a^{h-1} \le x' \le 0$. □

LEMMA 2. $\pi_h(x)$ can be bounded by

$$\pi_h(x) \le A(n)[p(0)]^h \qquad (17)$$

where $A(n)$ is a constant multiplier independent on h.

PROOF. Since $p(x)$ is continuous, there exists a positive constant α such that $p(x) \le p(0) - \alpha x$ for all $x \le 0$. Consequently, using Lemma 1, we can write

$$\max_x \pi_h(x) = \max_{a^{h-1}x_0 \le x \le 0} \pi_h(x)$$

$$\le \max_{a^{h-1}x_0 \le x \le 0} \prod_{i=0}^{h-1} (p(0) - \alpha x/a^i)$$

$$\le [p(0)]^h \max_{a^{h-1}x_0 \le x \le 0} \exp\left(\sum_{i=0}^{h-1} -\frac{\alpha x}{a^i p(0)}\right)$$

$$= [p(0)]^h \exp\left[\frac{-\alpha x_0}{p(0)} a^{h-1} \sum_{i=o}^{h-1} 1/a^i\right]$$

$$\le [p(0)]^h \exp\left[\frac{-\alpha x_0}{p(0)(1-a)}\right]$$

Selecting $A(n) = \exp[-\alpha x_0/p(0)(1-a)]$ proves the Lemma. □

THEOREM 2. The branching factor of the α–β procedure for a uniform tree of degree n is given by

$$\mathcal{R}_{\alpha-\beta} = \frac{\xi_n}{1 - \xi_n} \qquad (18)$$

where ξ_n is the positive root of the equation $x^n + x - 1 = 0$.

PROOF. Substituting (17) in (16) yields

$$N_{n,d} \le n^h + n\,A(n)[p(0)]^h \cdot \int_{-\infty}^{\infty} \sum_{i=0}^{h-1} (1/a^i)\phi'(x/a^i)\,dx$$

$$\simeq n^h + n\,A(n)[p(0)]^h h$$

Finally, using $p(0) = (\xi_n/1 - \xi_n)^2 > n$, we obtain

$$\mathcal{R}_{\alpha-\beta} = \lim_{h\to\infty} (N_{n,d})^{1/2h} \le \xi_n/1 - \xi_n \qquad (19)$$

This, together with Baudet's lower bound $\mathcal{R}_{\alpha-\beta} \ge \xi_n/1 - \xi_n$, completes the proof of Theorem 2. □

3. Conclusions

The asymptotic behavior of $\mathcal{R}_{\alpha-\beta}$ is $O(n/\log n)$, as predicted by Knuth's analysis [3]. However, for moderate values of n ($n \le 1000$), $\xi_n/1 - \xi_n$ is fitted much better by the formula $(0.925)n^{0.747}$ (see Fig. 4 of [5]), which vindicates the simulation results of Fuller et al. [2]. This approximation offers a more meaningful appreciation of the pruning power of the α–β algorithm. Roughly speaking, a fraction of only $(0.925)n^{0.747}/n \approx n^{-1/4}$ of the legal moves will be explored by α–β. Alternatively, for a given search time allotment, the α–β pruning allows the search depth to be increased by a factor $\log n/\log \mathcal{R}_{\alpha-\beta} \approx 4/3$ over that of an exhaustive minimax search.

The establishment of the precise value of $\mathcal{R}_{\alpha-\beta}$ for continuous-valued trees, together with a previous result that $\mathcal{R}_{\alpha-\beta} = n^{1/2}$ for almost all discrete-valued trees [5], completes the characterization of the asymptotic behavior of α–β and settles the question of its optimality. The fact that α–β is asymptotically optimal (that is, achieves the lowest possible branching factor) over the class of directional algorithms follows directly from Eq. (18) and a previous result [5] that $\xi_n/1 - \xi_n$ lower bounds the branching factor of any directional algorithm. However, the possible existence of some nondirectional algorithm outperforming α–β and exhibiting a branching factor lower than $\xi_n/1 - \xi_n$ has remained unsettled until very recently. Indeed, Stockman [9] introduced a nondirectional algorithm called *SSS** which consistently examines fewer nodes than α–β. Hopes were then raised that the superiority of Stockman's algorithms reflected an improved branching factor over that of α–β.

These possibilities have all been eliminated by a more recent result by Tarsi [10]. Considering a standard bi-valued game tree in which the terminal nodes are assigned the values 1 and 0 with the probabilities ξ_n and $1 - \xi_n$, respectively, Tarsi's result states that *any* algorithm which solves such a game tree must, on the average, examine at least $(\xi_n/1 - \xi_n)^d$ terminal positions. At the same time the task of solving any bi-valued game tree is equivalent to the task of verifying an inequality proposition regarding the minimax value of a continuous-valued game tree [5] of identical structure, and, consequently, the former cannot be more complex than the latter. Thus, the quantity $(\xi_n/1 - \xi_n)^d$ should also lower bound the expected number of nodes examined by any algorithm searching a continuous-valued game tree. This, together with Eq. (18), establishes the asymptotic optimality of α–β.

Received 5/80; revised 7/81; accepted 11/81

References

1. Baudet, G.M. On the branching factor of the alpha-beta pruning algorithm. *Artificial Intelligence 10*, 2 (April 1978), 173–199.
2. Fuller, S.H., Gaschnig, J.G., and Gillogly, J.J. An analysis of the alpha-beta pruning algorithm. Department of Computer Science Report, Carnegie-Mellon University, (1973).
3. Knuth, D.E., and Moore, R.N. An analysis of alpha-beta pruning. *Artificial Intelligence 6* (1975), 293–326.
4. Kuczma, M. *Functional Equations in a Single Variable.* Polish Scientific Publishers, Warszawa, (1968), p. 141.
5. Pearl, J. Asymptotic properties of minimax trees and game-searching procedures. *Artificial Intelligence 14*, 2 (Sept. 1980), 113–138.
6. Pearl, J. A space-efficient on-line method of computing quantile estimates *J. of Algorithms 2*, 2 (June 1981) 24–28.
7. Roizen, I. On the average number of terminal nodes examined by alpha-beta. UCLA-ENG-CSL-8108, Cognitive Systems Laboratory, University of California, Los Angeles, (1981).
8. Slagle, J.R., and Dixon, J.K. Experiments with some programs that search game trees. *JACM 16*, 2 (April 1969) 189–207.
9. Stockman, G. A minimax algorithm better than alpha-beta? *Artificial Intelligence 12*, 2 (Aug. 1979), 179–196.
10. Tarsi, M. Optimal searching of some game trees. UCLA-ENG-CSL-8108, Cognitive Systems Laboratory, University of California, Los Angeles, (1981). (To appear in *JACM.*)

Programming Techniques
And Data Structures

Ronald L. Rivest*
Editor

Heuristics for the Line Division Problem in Computer Justified Text

Hanan Samet
University of Maryland

Measures for evaluating solutions to the line division problem in computer justified text are presented. They are based on the belief that documents tend to have a more pleasing visual appearance when the deviation between interword breaks in a paragraph is reduced. This effect is achieved by not placing the maximum number of words on each line. The measures are variations on the variance of the number of extra spaces per interword break in a paragraph. They are applicable to both fixed and variable width fonts. One of the measures is examined in greater detail. It has the property that a lower bound can be computed, thereby indicating when further rearrangement of the text is futile. Several text rearrangement algorithms are proposed that make use of this measure.

CR Categories and Subject Descriptors: I.7.2 **[Text Processing]**: Document Preparation—*format and notation, photocomposition*; H.4 **[Information Systems Application]**: Office Automation—*word processing*

General Term: Algorithms

Additional Key Words and Phrases: line division, text justification, typesetting, layout, spacing, line breaking

1. Introduction

The dramatic rise in the use of interactive computer facilities has been coupled with a rise in the use of text editing programs. This has in turn led to the development of document processing systems whose role is to trans-

* Former editor of Programming Techniques and Data Structures, of which Ellis Horowitz is the current editor.

Author's present address: Hanan Samet, Department of Computer Science, University of Maryland, College Park MD 20742.