NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

OPTIMAL PROBLEM-SOLVING SEARCH:

ALL-OR-NONE SOLUTIONS

Herbert A. Simon
and
Joseph B. Kadane

Carnegie-Mellon University

ABSTRACT

Optimal algorithms are derived for satisficing problem-solving search, that is, search where the goal is to reach any solution, no distinction being made among different solutions. This task is quite different from search for best solutions or shortest path solutions.

Constraints may be placed on the order in which sites may be searched. This paper treats satisficing searches through partially ordered search spaces where there are multiple alternative goals.

(1) Simon was supported in part by Research Grant MH-07722 from the National Institute of Mental Health and in part by the Advanced Research Projects Agency of the Office of the Secretary of Defense (F44620-70-C-0107) which is monitored by the Air Force Office of Scientific Research; Kadane was supported in part by the Office of Naval Research under contract number N00014-67-A-0314-002.

OPTIMAL PROBLEM-SOLVING SEARCH:

ALL-OR-NONE SOLUTIONS

Herbert A. Simon
and
Joseph B. Kadane

Carnegie-Mellon University (1).

In the representation of problem solving as a search through a tree or directed graph, several different cases must be distinguished. In one case (Best-Value Search), values are associated with terminal nodes, and the aim of the search is to discover the node bearing the highest value. This case is only interesting if information becomes available during the course of the search which, by excluding some portions of the search space, makes exhaustive search unnecessary.

In the second case (Shortest-Path Search), which can be treated as a special case of the first, the value associated with a terminal node is the length of the shortest path from the starting point to that terminal, and the aim is to find the terminal with the smallest value (i.e., the terminal closest to the starting point). In either the first or second case, a further requirement may be imposed on the search algorithm that it be the algorithm for finding the best value or shortest path, as the case may be, that minimizes the expected search effort for attaining its goal.

In a third case (Satisficing Search), there is a designated subset of terminal nodes called goals, and the aim of the search is to reach any of these goals. No distinction is

made between the values of different goals -- all goals are equally desirable. In one variant of this case, there is a single, unique goal, and only one path leading to it. In either variant of this third case, we are interested in search algorithms that minimize the expected search effort for reaching the first goal (or, if the goal is unique, for reaching that goal). It is this third case which is the subject of the present paper.

Tasks of all of these three types are common in the literature of artificial intelligence. Game-playing programs are concerned with discovering "best" moves, hence involve best-value search. Programs for solving certain scheduling problems -- for example, the Traveling Salesman problem -- search for shortest paths. Theorem-proving programs, however, and most problem-solving programs are concerned with satisficing search.

Search algorithms designed to handle tasks of the different types may need to be vastly different, and the search effort required to find solutions may respond to quite different parameters of the task environments. Suppose, for example, that needles of varying sharpness have been distributed randomly throughout a haystack of size $\underline{S}$. A best-value search algorithm designed to find the sharpest needle in the haystack will have to search the entire stack, and will require an effort proportional to $\underline{S}$. A satisficing algorithm to find a needle sharp enough for sewing will only have to search until it discovers one such needle. Its expected search effort will be inversely proportional to the average density of "sharp enough" needles in the stack, and independent of $\underline{S}$.

If the needles are not distributed with uniform density, then the kind of information about the distribution that would be useful to guide a best-value search may be quite different from the information that would be useful to guide a satisficing

search. Intuitively, one can see that for best-value search it would be helpful to be able to set an upper bound on the sharpness of the needles to be found in any particular region. For satisficing search, one would want to know the probability of finding a sharp-enough needle in any given region. From this simple example, therefore, we see that we need separate theories of search algorithms for best-value and satisficing search, respectively.

Several cases of satisficing search have already been treated in the literature. These include both the case where sites may be searched in any order, without constraint, and the case where sites are partitioned into a number of classes, and there is a specified order in which sites in each class must be searched (but no between-class constraints). We call this case, "parallel search."

In the literature of artificial intelligence (Chang and Slagle, 1971; Kowalski, 1969, 1972; Nilsson, 1971; Pohl, 1971), algorithms are to be found for best-value and shortest-path search through trees. A recent paper by Garey (1973) provides an algorithm for satisficing search through trees, but neither the algorithm nor its method of derivation encompasses general partial orderings. It is the purpose of this paper to fill this gap by extending our results to satisficing searches where the ordering constraints are typical of those in problem-solving tasks: that is, to searches in partial orderings.

In the first two sections, we will review the optimal search algorithms for unconstrained and parallel (satisficing) searches, respectively. In the third section, we will extended our results to satisficing searches through partially ordered search spaces in the case where there are multiple alternative goals.

1. Satisficing Search Without Ordering Constraints

An unknown number of chests of Spanish treasure have been buried on a random basis at some of n sites, at a known depth of three feet. For each site there is a known unconditional probability, $p(i)$, $i=1,\ldots,n$, that a chest was buried there, and the cost of excavating site i is $q(i)$.

A strategy, t, is a permutation of any subset of the integers from 1 to n. Suppose a subset of sites is searched in the order given by t, under the condition that the search is terminated as soon as one treasure is found. then we can associate with the strategy t an expected cost $V(t)$ of this terminating search, and a probability $P(t)$ that a treasure will be found. $P(t) = 1-S(t)$, where $S(t)$ is the probability that there is treasure at none of the sites of t. We assume that $V(t) > 0$ and $S(t) > 0$ for all t. That is, no site can be excavated without cost and no site contains a treasure with certainty.

Let (ab) be the strategy consisting of executing strategy a, followed by b, where the subsets of a and b are non-overlapping. Then by our definitions, we have:

(1.1) $V(i) = q(i)$,

where i is the strategy of excavating the ith site.

(1.2) $V(ab) = V(a) + S(a)V(b)$

(1.3) $S(ab) = S(a)S(b)$.

Equation (1.2) states that the expected cost of a terminating search over (ab) is the expected cost, $V(a)$, of a terminating search over a plus the product of the expected cost of a terminating search over b by the probability, $S(a)$, that the latter search is necessary (i.e., that treasure was not found in a). Equation (1.3) states that the probability of not finding a treasure in (ab) is the product of the probabilities of not finding treasures in a and b, respectively.

The functional equations (1.2) and (1.3) are studied in Kadane (1969).

It is evident that $\underline{S}$ is associative and commutative, so that $\underline{S}((\underline{ab})\underline{c}) = \underline{S}(\underline{a}(\underline{bc}))$ and $\underline{S}(\underline{ab}) = \underline{S}(\underline{ba})$; while $\underline{V}$ is associative, but not commutative. Defining $\underline{A}_i = (\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_i)$, and $\underline{S}(\underline{A}_0) = \underline{S}(\underline{a}_0) = 1$, we find readily that:

(1.4) $$V(A_r) = V(a_1, a_2, \ldots, a_r) = \sum_{i=1}^{r} \prod_{j=0}^{i-1} S(a_j)V(a_i) = \sum_{i=1}^{r} S(A_{i-1})V(a_i)$$

We also note for later reference that:

(1.5) $$P(ab) = P(a) + P(b) - P(a)P(b) = P(a) + S(a)P(b).$$

We consider now the effect upon the expected cost of search of excavating the same set of sites but in different orders -- by strategies $(\underline{abcd})$ and $(\underline{acbd})$, say, where $\underline{a}$ and $\underline{d}$ may be empty, $\underline{S}(\underline{x})$, with $\underline{x}$ empty, equals 1, and $\underline{V}(\underline{x})$, with $\underline{x}$ empty, equals zero.

(1.6) $$V(abcd) - V(acbd) = V(a) + S(a)V(bcd) - V(a) - S(a)V(cbd)$$

(1.7) $$= S(a)[V(bcd) - V(cbd)]$$

(1.8) $$= S(a)[V(bc) + S(bc)V(d) - V(cb) - S(cb)V(d)]$$

But, since $\underline{S}(\underline{bc}) = \underline{S}(\underline{cb})$, Equation (1.8) simplifies to:

(1.9) $$V(abcd) - V(acbd) = S(a)[V(bc) - V(cb)]$$

(1.10) $$= S(a)[V(b)P(c) - V(c)P(b)]$$

In particular, if $\underline{b}$ and $\underline{c}$ consist of the single sites $\underline{i}$ and $\underline{j}$, respectively, then it will be cheaper to excavate $\underline{i}$ before $\underline{j}$ iff $\emptyset(\underline{i}) > \emptyset(\underline{j})$, where $\emptyset(\underline{t}) = \underline{P}(\underline{t})/\underline{V}(\underline{t})$. Moreover, this result holds for all $\underline{i}$ and $\underline{j}$. The optimal strategy, therefore, for finding a single treasure is to excavate sites in descending order of $\emptyset(\underline{i})$ until a treasure is discovered.

Similar results were obtained by Dean (1965), Kadane (1969), Joyce (1971), and Mitten (1960).

2. Search With Parallel Ordering

Suppose, now, that the Spanish treasures are buried, as before, but that neither the sites nor the depths of burial are known with certainty. At each site a sequence of one-foot slices can be excavated, and a treasure may be disclosed by the removal of any one of these slices. The probability that a treasure lies just below any specified slice is known. Designate the probability that the treasure lies below slice f of site i as $p(i,f)$.

A particular slice (i,g) can only be searched after all the other slices above it, (i,f), $f < g$, have been searched. Hence, an admissible search strategy will be an ordering of a subset of slices such that (i,g) does not precede (i,f) if $g > f$. For a given strategy, let $t(i,f)$ be the order number of the slice (i,f).

Now, we can define quantities, $V(t)$, $P(t)$, and $S(t)$ exactly as before, so that Equations (1.1) through (1.10) are again valid for the admissible strategies. We wish to find the strategy that minimizes $V(t)$ where t ranges over permutations of the entire set of n integers subject to the order constraint that $t(i,f) > t(i,g)$ if $g > f$.

A bloc of slices is a set of slices belonging to the same site that are consecutive in that site. (The members of a bloc need not be consecutive in any particular strategy, since they may be interspersed with one or more slices from other sites.)

In a strategy (abc), b is (weakly) monotonic decreasing if for each pair of segments, with order numbers i, j in b, $\emptyset(j) \leq \emptyset(i)$ if $i < j$. The substrategy b is monotonic increasing if, for each i, j in the sequence $\emptyset(j) \geq \emptyset(i)$ if $i < j$.

In a strategy $(abcd)$, b and c are interchangeable if permuting the strategy into $(acbd)$ does not violate the order constraints (i.e., does produce another strategy).

Theorem 2.1: In $(abcd)$, b and c are interchangeable iff no

site represented by one or more slices in $\underline{b}$ is represented by any slices in $\underline{c}$.

Proof: The order constraints apply to pairs of slices belonging to the same site. Under the conditions of the theorem, interchange of $\underline{b}$ with $\underline{c}$ will not reverse the order of any pair of slices belonging to the same site, hence will not violate any order constraints.

Conversely, if slice $\underline{t}(\underline{i},\underline{f})$ in $\underline{b}$ and slice $\underline{t}(\underline{i},\underline{g})$ in $\underline{c}$ belong to the same site, then the order constraints require that $\underline{f} < \underline{g}$ (since $\underline{t}(\underline{i},\underline{f}) < \underline{t}(\underline{i},\underline{g})$). But, in $(\underline{cb})$ we will have $\underline{t}(\underline{i},\underline{f}) > \underline{t}(\underline{i},\underline{g})$, which violates the constraints.

Q. E. D.

Theorem 2.2: If $\underline{b}$ and $\underline{c}$ are interchangeable in $(\underline{abcd})$, and if $\emptyset(\underline{c}) > \emptyset(\underline{b})$, then the strategy can be improved by interchanging $\underline{b}$ and $\underline{c}$, hence is not optimal.

Proof: Using Equation (1.10), we have:

(2.1) $$V(abcd) - V(acbd) = S(a)[V(b)P(c) - V(c)P(b)]$$

(2.2) $$= S(a)V(b)V(c)[\emptyset(c) - \emptyset(b)] > 0.$$

Q. E. D.

Theorem 2.3: If $\underline{b}$ and $\underline{d}$ are strategies that are also consecutive blocs of site $\underline{i}$, with $\underline{d}$ following $\underline{b}$, and if $\emptyset(\underline{d}) > \emptyset(\underline{b})$, then a strategy of $(\underline{abcde})$ with $\underline{c}$ non-null is not optimal.

Proof: By the ordering constraints, $\underline{d}$ cannot precede $\underline{b}$. Suppose it does not follow immediately ($\underline{c}$ is non-null). Since $(\underline{bd})$ is a bloc, no member of $\underline{c}$ belongs to site $\underline{i}$, for all other slices of $\underline{i}$ must, by the ordering constraints, precede $\underline{b}$ or follow $\underline{d}$. Either $\emptyset(\underline{c}) > \emptyset(\underline{b})$, or $\emptyset(\underline{c}) \leq \emptyset(\underline{b}) < \emptyset(\underline{d})$.

In the first case, the strategy can be improved (Theorem 2.2) by interchanging c with b, hence is not optimal. In the second case, the strategy can be improved by interchanging c with d, hence is not optimal.

Q. E. D.

We call the bloc (bd) of Theorem 2.3 an indivisible bloc. Each indivisible bloc is made up of consecutive slices from a single site, and its indivisibility depends only on the $\emptyset$'s of strategies from that site. Hence we can now proceed, for each individual site, to determine its maximal indivisible blocs by joining blocs that satisfy the conditions of Theorem 2.3 until the $\emptyset$'s for all the separate blocs that remain are monotonic decreasing. We can state this result as a corollary:

Corollary 2.3.1: An optimal solution consists of a sequence of maximal indivisible blocs such that the $\emptyset$'s of the successive blocs of any given site are monotonic decreasing.

We are now ready for the main theorem.

Theorem 2.4: If a strategy consists of a sequence of maximal indivisible blocs, and if the $\emptyset$'s associated with these blocs are monotonic decreasing, then the strategy is optimal.

Proof: 1. Corollary 2.3.1 guarantees that any optimal strategy must consist of a sequence of maximal indivisible blocs, and that the $\emptyset$'s of the subsequence of blocs belonging to any given site are monotonic decreasing.

2. The strategy defined in Theorem 2.4 is unique up to trivial interchanges of segments with equal $\emptyset$, which do not change the value of $\underline{V}$.

3. The only allowable permutations that preserve the maximal

indivisible blocs and their order involve interchanges of strategies belonging to different sites and not separated by a strategy belonging to either of their sites.

4. Suppose a strategy is optimal, but that $\emptyset$ is not monotonic decreasing. Consider the first instance where $\emptyset(\underline{b}) > \emptyset(\underline{a})$, with $\underline{a}$ immediately preceding $\underline{b}$. Since the blocs belonging to a given site are monotonic decreasing in value, $\underline{a}$ and $\underline{b}$ must belong to different sites, hence are interchangeable, by Theorem 2.1. Therefore, by Theorem 2.2, the strategy would be improved by interchanging $\underline{a}$ and $\underline{b}$, contrary to the hypothesis that the strategy is optimal.

Q. E. D.

Theorem 2.4 tells us that the optimal strategy in digging for doubloons is to calculate the average yield (per foot of digging) for each maximal indivisible bloc of a site, then excavate the successive blocs in decreasing order of yield.

The evaluation functions $\emptyset$, take into account not only the potential return, $\underline{p}(i)$, and cost, $\underline{q}(i)$, from the slice being executed, but the prospective value of getting closer to underlying slices that have larger $\emptyset$ values than the current slice. This characteristic of the evaluation function adds a certain "depth-first" tendency to the strategy. For example, suppose that it is known that the treasure is buried not less than five feet below the ground. Then, if the optimal strategy calls for excavation to begin at the $\underline{i}$th site, it will continue at that site until the excavation has reached a depth of at least five feet.

In the special case in which $\underline{g}>\underline{f}$ implies $\emptyset(\underline{i},\underline{g}) < \emptyset(\underline{i},\underline{f})$, Kadane (1969) finds the optimal ordering to be according to $\emptyset$. The results of this section are new when the above condition is not satisfied.

3. Search Through a Partially-Ordered Space

Our next task is to extend the results of the previous section to a search through a partially-ordered set of nodes, or cycle-free graph, which includes the familiar case of search through a tree. Theorem-proving and problem-solving searches are commonly representable as searches through trees or, more generally through partial orderings. In such a search, a new node is obtained by applying an operator to branch from some node reached previously. In this section, we will prove a theorem (Theorem 3.1) for optimal search through a partial ordering which is analogous to Theorems 2.3 and 2.4 for parallel orderings. As before, the key role is played by an evaluation function, $\emptyset''$, which can be assigned to each branch at each node already reached in such a way that it is always optimal to search next the branch with largest $\emptyset''$.

The proof of optimality for a partial ordering is a great deal more complex than the proof for a parallel ordering, mainly because $\emptyset$ for a node now has to be maximized over all the alternative sequences descendant from that node. The notion of the "best set" of a node (the set of nodes descendant from that node for which $\emptyset$ is maximum) replaces the "maximum indivisible bloc" of the previous section. The $\underline{D}''$ of Theorem 3.1 is this best set.

As before, we assume that for each node, $\underline{i}$, of the set a value $\underline{p}(\underline{i})$ is given, representing the probability that a solution will be found at that node. The cost of excavating each node from one immediately before it is $\underline{q}(\underline{i})$.

A strategy $\underline{t}(\underline{D})$ for a set of nodes, $\underline{D}$, is any ordering of the nodes of $\underline{D}$ that satisfies the order constraints on those nodes. Let $\underline{A}$ and $\underline{B}$ be two mutually exclusive sets of nodes, and $\underline{C}$ their set sum. The $\underline{A}$ and $\underline{B}$ are interchangeable iff there exists a strategy $\underline{c} = (\underline{ab})$ and a strategy $\underline{c}'' = (\underline{b}''\underline{a}'')$, where $\underline{c}$ and $\underline{c}''$ are strategies on $\underline{C}$, $\underline{a}$ and

$\underline{a}''$ strategies on $\underline{A}$, and $\underline{b}$ and $\underline{b}''$ strategies on $\underline{B}$. Clearly, if $\underline{A}$ and $\underline{B}$ are interchangeable, if $\underline{a}$ is any strategy of $\underline{A}$ and if $\underline{b}$ is any strategy of $\underline{B}$, then $(\underline{ab})$ and $(\underline{ba})$ are strategies of $\underline{C}$.

Corresponding to the notion of a bloc in the previous section, we introduce the concepts of <u>initial</u> and <u>terminal blocs</u> of a set of nodes. Let $\underline{A}$ be a partially ordered set of nodes, and let it contain $\underline{B}$ and $\underline{C} = \underline{A}-\underline{B}$. Then $\underline{B}$ is an initial bloc of $\underline{A}$ iff there exist strategies $\underline{b}$ on $\underline{B}$ and $\underline{c}$ on $\underline{C}$ such that $\underline{a} = (\underline{bc})$ is a strategy on $\underline{A}$. $\underline{B}$ is a terminal bloc of $\underline{A}$ iff there exist strategies $\underline{b}$ on $\underline{B}$ and $\underline{c}$ on $\underline{C}$ such that $\underline{a} = (\underline{cb})$ is a strategy on $\underline{A}$.

As before, we can define for each strategy the quantities $\underline{V}(\underline{t})$, $\underline{P}(\underline{t})$, $\underline{S}(\underline{t})$ and $\emptyset(\underline{t}) = \underline{P}(\underline{t})/\underline{V}(\underline{t})$, all of which depend only on the subset, $\underline{D}$, and its ordering, $\underline{t}$, independently of the remaining nodes in the entire set. Note that $\underline{P}(\underline{t})$ and $\underline{S}(\underline{t})$ are constant over all strategies of a given set, $\underline{D}$, hence are functions of $\underline{D}$; while $\underline{V}(\underline{t})$ and $\emptyset(\underline{t})$ depend upon the strategy, $\underline{t}$, as well as the set, $\underline{D}$.

A strategy of a set of nodes, $\underline{D}$, for which $\emptyset$ assumes its greatest value for that set will be called a best strategy of $\underline{D}$ and will be designated by $\underline{t}'(\underline{D})$ and its $\underline{V}$ by $\underline{V}'(\underline{D})$. An initial bloc, $\underline{D}$, of set $\underline{I}$ for which $\emptyset'(\underline{D})$ is maximal over all initial blocs of $\underline{I}$ will be called a best set of $\underline{I}$, and the $\emptyset$ of its best strategy will be designated by $\emptyset''(\underline{I})$.

We now prove four lemmas that are needed for our main theorem.

Lemma 3.1. Let $\underline{A}$ and $\underline{A}\cup\underline{B}$ be initial sets of $\underline{I}$ such that: (1) $\underline{A}$ is the best set of $\underline{I}$, with best strategy $\underline{t}'(\underline{A}) = \underline{a}$; and (2) $\underline{b}$ is any strategy for $\underline{B}$.

Then, $\emptyset'' \geq \emptyset(\underline{b})$.

Proof. Since $\underline{A}$ is the best set of $\underline{I}$, $\emptyset'' = \emptyset(\underline{a}) \geq \emptyset(\underline{ab})$.

Now, using Equation 1.5,

(3.1) $V(ab)\emptyset(ab) = P(ab) = P(a) + S(a)P(b)$

(3.2) $= V(a)\emptyset(a) + S(a)V(b)\emptyset(b)$

But $V(ab)\emptyset(a) \geq V(ab)\emptyset(ab)$, so that,

(3.3) $V(ab)\emptyset(a) \geq V(a)\emptyset(a) + S(a)V(b)\emptyset(b)$

Expanding the left-hand side, we get:

(3.4) $[V(a) + S(a)V(b)]\emptyset(a) \geq V(a)\emptyset(a) + S(a)V(b)\emptyset(b)$

(3.5) $S(a)V(b)\emptyset(a) \geq S(a)V(b)\emptyset(b).$

so that, since we have postulated that $S(a) \neq 0$ and $V(b) \neq 0$,

(3.6) $\emptyset(a) = \emptyset'' \geq \emptyset(b)$

Q. E. D.

Lemma 3.2. Let $\underline{A}$ be a set consisting of the mutually exclusive subsets of nodes $\underline{B}$, $\underline{C}$, and $\underline{D}$, where $\underline{B}$ is an initial bloc of $\underline{A}$, while $\underline{C}$ and $\underline{D}$ are interchangeable, hence also both terminal blocs. Let the best strategy, $\underline{t}'(\underline{A})$ be:

$$\underline{t}'(\underline{A}) = (\underline{b}\underline{c}_1\underline{d}_1 \ldots \underline{c}_k\underline{d}_k),$$

where $\underline{b}$ is a strategy for $\underline{B}$, $\underline{c} = (\underline{c}_1 \ldots \underline{c}_k)$ is a strategy for $\underline{C}$ and $\underline{d} = (\underline{d}_1 \ldots \underline{d}_k)$ is a strategy for $\underline{D}$.

Then $\emptyset(\underline{c}_1) \geq \emptyset(\underline{d}_1) \geq \ldots \geq \emptyset(\underline{c}_k) \geq \emptyset(\underline{d}_k)$

Proof: Suppose $\emptyset(\underline{d}_i) < \emptyset(\underline{c}_{i+1})$. Then, by Equation (1.10), $\underline{t}'(\underline{A})$ could be improved by exchanging $\underline{d}_i$ and $\underline{c}_{i+1}$, contrary to the hypothesis that $\emptyset(\underline{A})$ is maximal. But the exchange is admissible, since $\underline{C}$ and $\underline{D}$ are interchangeable. Similarly, the supposition that $\emptyset(\underline{c}_i) < \emptyset(\underline{d}_i)$ leads to a contradiction.

Q. E. D.

Lemma 3.3. Given $\underline{A}$, $\underline{B}$, $\underline{C}$ and $\underline{D}$ as in Lemma 3.2, with $\underline{c} = (\underline{c}_1 \ldots \underline{c}_k)$ and $d = (\underline{d}_1 \ldots \underline{d}_k)$, suppose that $\underline{A}$ is a best set of $\underline{I}$, so that $\emptyset'(\underline{A}) = \emptyset''$. Then: $\emptyset(\underline{d}) \geq \emptyset'(\underline{A})$, and therefore $\emptyset'(\underline{D}) \geq \emptyset'(\underline{A}) = \emptyset''$.

Proof: Define,

$$\underline{e} = (\underline{b}\underline{c}_1\underline{d}_1 \ldots \underline{c}_k)$$

Then,

$$\emptyset'' = \frac{\underline{P}(\underline{e})+\underline{S}(\underline{e})\underline{P}(\underline{d}_k)}{\underline{V}(\underline{e})+\underline{S}(\underline{e})\underline{V}(\underline{d}_k)}$$

If $\emptyset(\underline{d}_k) = \underline{P}(\underline{d}_k)/\underline{V}(\underline{d}_k) < \emptyset''$, then $\emptyset(\underline{e}) > \emptyset''$. But $\underline{e}$ is a strategy for an initial bloc of $\underline{A}$, and also of $\underline{I}$. Since $\emptyset''$ is maximal over all such blocs, the inequality is a contradiction. Therefore $\emptyset(\underline{d}_k) \geq \emptyset''$.

But, by Lemma 3.2,

$$\emptyset(\underline{d}_k) \leq \emptyset(\underline{c}_k) \leq \ldots \leq \emptyset(\underline{d}_1) \leq \emptyset(\underline{c}_1).$$

$$\emptyset(\underline{d}) = \frac{\underline{P}(\underline{d}_1)+\underline{S}(\underline{d}_1)\underline{P}(\underline{d}_2)+\ldots+\underline{S}(\underline{d}_1)\ldots\underline{S}(\underline{d}_{k-1})\underline{P}(\underline{d}_k)}{\underline{V}(\underline{d}_1)+\underline{S}(\underline{d}_1)\underline{V}(\underline{d}_2)+\ldots+\underline{S}(\underline{d}_1)\ldots\underline{S}(\underline{d}_{k-1})\underline{V}(\underline{d}_k)}$$

But, by definition, $\emptyset(\underline{D}) \geq \emptyset(\underline{d})$, whence,

$$\emptyset'(\underline{D}) \geq \emptyset''.$$

Q. E. D.

Lemma 3.4. Let $(\underline{acd})$ and $(\underline{ac}''d)$ be strategies over the same set of nodes. Then,

(3.7) $V(acd) - V(ac''d) = S(a)[V(c) - V(c'')]$

Proof:

(3.8) $V(acd) - V(ac''d) = V(a) + S(a)V(cd) - V(a) - S(a)V(c''d)$

(3.9) $= S(a)[V(c)+S(c)V(d)-V(c'')-S(c)V(d)]$

(3.10) $= S(a)[V(c) - V(c'')]$

Q. E. D.

In order to state and prove the main theorem, we need to introduce some additional notation. Let $(\underline{a}_1 \ldots \underline{a}_r)$ and $(\underline{b}_1 \ldots \underline{b}_r)$ be strategies. Define $\underline{A}'_i$ as the best permutation of $(\underline{a}_1 \ldots \underline{a}_r)$, $\underline{B}_i = (\underline{b}_1 \ldots \underline{b}_i)$, $\underline{C}_i = (\underline{b}_1 \underline{a}_1 \ldots \underline{b}_i \underline{a}_i)$, $\underline{A}_i = (\underline{A}'_{i-1} \underline{a}_i)$, $\underline{A}'^*_i$ as the best permutation of $(\underline{a}_{i+1} \ldots \underline{a}_r)$, $\underline{B}^*_i = (\underline{b}_{i+1} \ldots \underline{b}_r)$, $\underline{C}^*_i = (\underline{b}_{i+1} \underline{a}_{i+1} \ldots \underline{b}_r \underline{a}_r)$, $\underline{A}^*_i = (\underline{a}_{i+1} \underline{A}'^*_{i-1})$

Define $\underline{C}_0 = \underline{C}^*_r = \underline{A}_0 = \underline{A}^*_r = \underline{B}_0 = \underline{B}^*_r = \lambda$, the null strategy, with $\underline{S}(\lambda) = 1$, $\underline{V}(\lambda) = 0$.

Consider a strategy over the set $\underline{I}$, having the form $(\underline{f}\underline{g})$, where $\underline{f}$ and $\underline{g}$ are strategies over the non-overlapping sets $\underline{E}$ and $\underline{G}$, respectively. Let $\emptyset''$ be the maximum of $\emptyset$ over all strategies of $\underline{G}$, and let $\underline{D}''$ be an initial bloc of $\underline{G}$, and $\underline{t}'(\underline{D}'')$ a strategy for $\underline{D}''$ such that $\emptyset(\underline{t}'(\underline{D}'') = \emptyset''$. Finally, let $\underline{h}$ be the initial segment of $\underline{g}$, consisting of $\underline{t}'(\underline{D}'')$ possibly interspersed with other nodes of $\underline{G}$ not belonging to $\underline{D}''$, and having the last element of $\underline{t}'(\underline{D}'')$ as its last element. We now prove the theorem:

Theorem 3.1: If $\underline{h}$ contains any nodes not belonging to $\underline{D}''$, then $(\underline{f}\underline{g})$ can be improved (weakly) by moving these "intruding" nodes beyond the last node of $\underline{D}''$, that is, by bringing the nodes of $\underline{D}''$ to the front of $\underline{h}$ with the remaining nodes of $\underline{h}$ following them.

Stating the theorem in the notation previously introduced, we designate $\underline{t}'(\underline{D}'')$ by $\underline{A}_r = (\underline{a}_1 \underline{a}_2 \ldots \underline{a}_r)$, and $\underline{h}$ by $\underline{C}_r = (\underline{b}_1 \underline{a}_1 \ldots \underline{b}_r \underline{a}_r)$, so that $\underline{B}_r = (\underline{b}_1 \ldots \underline{b}_r)$ is a strategy on the intruding nodes. Finally, we define $\underline{m}$ such that $\underline{g} = \underline{C}_r \underline{m}$. By definition of $\underline{D}''$, $\underline{A}'^*_0 = \underline{A}^*_0$. The permutation of $\underline{h}$, asserted by the Theorem to be an improvement, is then $\underline{h}'' = (\underline{A}_r \underline{B}_r) = \underline{A}^*_0 \underline{B}^*_0$.

If $\underline{V}(\underline{f}\underline{g})$ is infinite, there is nothing to prove. On the other hand, if $\underline{V}(\underline{f}\underline{g})$ is finite, then

(3.11) $V(fg) = V(f) + S(f)V(C_r) + S(f)S(C_r)V(m) < \infty$

so that $\underline{V}(\underline{f})$, $\underline{V}(\underline{C}_r)$ and $\underline{V}(\underline{m})$ are all finite. Now,

(3.12) $V(fA^*_0B^*_0m) = V(f) + S(f)V(A^*_0B^*_0) + S(f)S(A^*_0B^*_0)V(m)$

Now $\underline{S}$ is commutative, so $\underline{S}(\underline{A}^*_0\underline{B}^*_0) = \underline{S}(\underline{C}_r)$. Then,

(3.13) $V(fg) - V(fA^*_0B^*_0m) = S(f)[V(C_r) - V(A^*_0B^*_0)]$.

Therefore, we wish to prove that:

(3.14) $V(C_r) - V(A^*_0B^*_0) \geq 0$.

Note that $\underline{a}$'s may be advanced forward, interchanging them with $\underline{b}$'s, since $\underline{D}''$ is an initial bloc of $\underline{M}$.

Proof: By identity, and remembering that

$A^*_r = B^*_r = \lambda$, $A'^*_0 = A^*_0$,

(3.15) $V(C_r) - V(A^*_0B^*_0) = \sum_{i=1}^{r} [V(C_iA'^*_iB^*_i)-V(C_{i-1}A'^*_{i-1}B^*_{i-1})]$

Considering the individual terms of the summation, we have:

(3.16) $V(C_iA'^*_iB^*_i)-V(C_{i-1}A'^*_{i-1}B^*_{i-1})$

$= V(C_{i-1}b_iA^*_{i-1}B^*_i)-V(C_{i-1}A'^*_{i-1}b_iB^*_i)$

(3.17) $= V(C_{i-1}b_iA^*_{i-1}B^*_i)-V(C_{i-1}b_iA'^*_{i-1}B^*_i)$

$+V(C_{i-1}b_iA'^*_{i-1}B^*_i)-V(C_{i-1}A'^*_{i-1}b_iB^*_i)$

But, by Lemma 3.4, $\underline{V}(\underline{C}_{i-1}b_i\underline{A}^*_{i-1}\underline{B}^*_i)-\underline{V}(\underline{C}_{i-1}b_i\underline{A}'^*_{i-1}\underline{B}^*_i) \geq 0$.

Therefore,

(3.18) $V(C_iA'^*_iB^*_i) - V(C_iA'^*_{i-1}B^*_{i-1})$

$\geq V(C_{i-1}b_iA'^*_{i-1}B^*_i) - V(C_{i-1}A'^*_{i-1}b_iB^*_i)$.

Applying Equation (1.10), with $\underline{C}_{i-1} = \underline{a}$,

$\underline{b}_i = \underline{b}$, $\underline{A}'^*_{i-1} = \underline{c}$, $\underline{B}^*_i = \underline{d}$,

we get,

$$(3.19)\quad V(C_iA'^*_iB^*_i) - V(C_iA'^*_{i-1}B^*_{i-1})$$

$$\geq S(C_{i-1})[P(A'^*_{i-1})V(b_i) - P(b_i)V(A'^*_{i-1})], \text{ whence}$$

$$(3.20)\quad V(C_r) - V(A^*_0B^*_0) \geq \sum_{i=1}^{r} S(C_{i-1})V(b_i)V(A'^*_{i-1})(\emptyset(A'^*_{i-1})-\emptyset(b_i))$$

$$= T_1 - T_2$$

where,

$$(3.21)\quad T_1 = \sum_{i=1}^{r} S(C_{i-1})V(b_i)V(A'^*_{i-1})\emptyset(A'^*_{i-1})$$

$$(3.22)\quad T_2 = \sum_{i=1}^{r} S(C_{i-1})V(b_i)V(A'^*_{i-1})\emptyset(b_i)$$

Consider I_1. $\underline{A}'_{i-1}$ is the best strategy of a terminal bloc of $\underline{D}''$.

Hence, by Lemma 3.3, $\emptyset(\underline{A}'^*_{i-1}) \geq \emptyset''$. so that

$$(3.23)\quad T_1 \geq \sum_{i=1}^{r} S(C_{i-1})V(b_i)V(\underline{A}'^*_{i-1})\emptyset''$$

Next, consider I_2. Factoring $\underline{S}(\underline{C}_{i-1}) = \underline{S}(\underline{B}_{i-1})\underline{S}(\underline{A}_{i-1})$,

in (3.22), we obtain,

$$(3.24)\quad T_2 = \sum_{i=1}^{r} S(B_{i-1})V(b_i)\emptyset(b_i)S(A_{i-1})V(A'^*_{i-1})$$

Since $\underline{V}(\underline{A}^*_r) = 0$, we have the identity:

$$(3.25)\quad S(A_{i-1})V(A'^*_{i-1}) = \sum_{j=1}^{r} Z_j, \text{ where}$$

$$(3.26)\quad Z_j = S(A_{j-1})V(A'^*_{j-1}) - S(A_j)V(A'^*_j)$$

Using Equation (3.25) in (3.24), and then changing

the order of summation, we find,

$$(3.27)\quad T_2 = \sum_{i=1}^{r} \sum_{j=1}^{r} S(B_{i-1})V(b_i)\emptyset(b_i)Z_j$$

(3.28) $$= \sum_{j=1}^{r} Z_j \sum_{i=1}^{j} S(B_{i-1})P(b_i)$$

(3.29) $$= \sum_{j=1}^{r} Z_j P(B_j) = \sum_{j=1}^{r} Z_j V(B_j)\emptyset(B_j)$$

But $\underline{B}_j$ satisfies the conditions of $\underline{B}$ of Lemma 3.1, with $\underline{A}'_r$ as $\underline{A}$.

Therefore, by that lemma, $\emptyset(\underline{B}_j) \leq \emptyset''$. Hence,

(3.30) $$T_2 \leq [\sum_{j=1}^{r} Z_j V(B_j)]\emptyset''$$

(3.31) $$\leq [\sum_{j=1}^{r} Z_j \sum_{i=1}^{j} S(B_{i-1})V(b_i)]\emptyset''$$

(3.32) $$\leq \sum_{i=1}^{r} S(B_{i-1}) \sum_{j=1}^{r} Z_j V(b_i)\emptyset''$$

(3.33) $$\leq \sum_{i=1}^{r} S(B_{i-1})V(b_i)S(A_{i-1})V(A'^*_{i-1})\emptyset'' \quad \text{(by (3.25))}$$

(3.34) $$\leq \sum_{i=1}^{r} S(C_{i-1})V(b_i)V(A'^*_{i-1})\emptyset''$$

Combining (3.23) and (3.31), we have, finally:

(3.35) $$V(C_r) - V(A^*_0B^*_0) \geq T_1 - T_2$$

$$\geq \sum_{i=1}^{r} [S(C_{i-1})V(b_i)V(A'^*_{i-1})](\emptyset'' - \emptyset'') = 0.$$

Comparing (3.35) with Equation (3.14), we see that this is the result we want, and the theorem is proved.

Q. E. D.

4. Conclusion

In conclusion we wish to comment on how our results can be used in constructing search algorithms. We will consider the case of a search through a partial ordering since search through a parallel ordering and independent search can be regarded simply as special cases.

In some applications, we will have, in advance, a map of the entire search graph, together with an estimate of $p(i)$ for each of its nodes. In this case, $\emptyset''$ can be estimated for each node after determining the maximal indivisible blocs. In other applications, the search tree will only evolve in the course of the search itself. Then $\emptyset''$ cannot be determined from the $p(i)$'s, but will have to, somehow, be estimated directly.

In the case where a map of the search graph is given in advance, determination of the $\emptyset''$ may be considerably facilitated by using an algorithm described by Garey (1973). This algorithm provides a method, applicable to the "tree-like" portions of a partial ordering, for reducing the entire set of nodes to a smaller set, essentially by discovering the maximal indivisible blocs and the optimal strategies for them. After Garey's algorithm has been applied to carry the reduction as far as possible, the maximal indivisible blocs for the reduced system can be discovered, and the values of $\emptyset''$ associated with them computed. Once these values have been found, or estimated directly if the search graph is not given in advance, the algorithm described in the next paragraphs can be used to order the nodes optimally.

Suppose that, in a partial ordering, we assign to each branch, B, at each node, N, an ordinal number, $E(B,N)$. Then, we can construct a search algorithm of the familiar search-scan variety, as follows:

Let (B,N) designate branch B at node N, and let L be the list of all (B,N) pairs available for search, ordered according to $E(B,N)$.

1. Choose the first (B,N) pair on L

   and generate the new node, N";

2. If N" is a goal node stop, else;

3. Compute E(N",B) for all branches from N",

   and insert the new (N",B) pairs in their

   appropriate positions in L;

4. Return to Step 1.

Theorem 3.1 shows that if E(N",B) is set equal to the $\emptyset$" defined in the text, then the search determined by the above algorithm is optimal.

The value, $\emptyset$"(N,B), depends on the probabilities of reaching the goal, p(i), at some of the nodes that are descendants of N. In practice, these values of p(i) will usually not be known, and E in the algorithm will have to be a heuristic estimate of $\emptyset$" -- some estimate of the "promise" of searching from N in the direction defined by B. Theorem 3.1 indicates what the nature of this estimate should be.

In estimating $\emptyset$", we must postulate that the search will be continued through a "best set" of nodes, the set of reachable nodes that maximizes the ratio of expected return to cost of search. Search should continue in one direction at least as long as this ratio continues to increase, and in fact, until it becomes lower than the best ratio for some other branch. By this procedure, depth of search is balanced against the expectation of success, so that a modest probability of success after a short search may imply the same $\emptyset$" as a higher probability of success after a longer search.

References

Chang, C. L. & Slagle, J. R. (1971) An admissible and optimal algorithm for searching and-or graphs. Artificial Intelligence, 2:117-128.

Dean, B. V. (1966) Stochastic networks in research planning. (eds. Yovits, M. C. et al.) Research Program Effectiveness, New York: Gordon & Breach.

Garey, M. R. (1973) Optimal task sequencing with precedence constraints. Discrete Mathematics, 4:37-56.

Joyce, W. B. (1971) Organization of unsuccessful R & D programs, IEEE Trans. Eng. Managm., 18, 57-65.

Kadane (1969) Quiz show problems, J. Math. Analysis and Application, 27:609-623.

Kowalski, R. (1969) Search strategies for theorem-proving. Machine Intelligence 5, pp. 181-201. (eds. Meltzer, B. & Michie, D.) Edinburgh: Edinburgh Univ. Press.

Kowalski, R. (1972) And-or graphs, theorem-proving graphs and bi-directional search. Machine Intelligence 7, pp. 167-194. (eds. Meltzer, B. & Michie, D.) Edinburgh: Edinburgh Univ. Press.

Mitten, L. G. (1960) An analytic solution to the least cost testing sequence problem, J. Indust. Eng. 11:17.

Nilsson, N. J. (1971) Problem-solving Methods in Artificial Intelligence. New York: McGraw-Hill.

Pohl, I. (1971) Bi-directional search. Machine Intelligence 6, pp. 127-140 (eds. Meltzer, B. & Michie, D.) Edinburgh: Edinburgh Univ. Press.