Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Branch-and-Bound Methods: General Formulation and Properties

L. G. Mitten,

BRANCH-AND-BOUND METHODS: GENERAL FORMULATION AND PROPERTIES

L G Mitten

University of British Columbia, Vancouver, Canada

(Received November 12, 1968)

The branch-and-bound procedure is formulated in rather general terms and necessary conditions for the branching and bounding functions are precisely specified Results include the standard properties for finite procedures, plus several convergence conditions for infinite procedures Discrete programming which includes integer programming and combinatorial optimization problems, is discussed and Fibonacci search is presented as an example of a nonfinite branch-and-bound procedure employing an optimal convergence rule

BRANCH-AND-BOUND methods have found application in the solution of a variety of important optimization problems—e g , integer programming (references 2, 3, 8, 9, 12, 16, 22), nonlinear assignment problems (references 11, 13, 17, 24), scheduling problems (references 7, 14, 20), network problems[4 21] the knapsack problem,[15] the travelling-salesman problem,[19] plant location problems,[10] etc A survey of some of the literature will be found in reference 18

Several papers (e g , references 1, 5, 6, 23, 25) have been devoted to explorations of some of the basic characteristics of branch-and-bound methods Of these, the work of BERTIER AND ROY[6] and BALAS[5] has been particularly illuminating The present paper seeks to extend these results and to provide a somewhat more general theoretical framework for branch-and-bound methods

In essence, branch-and-bound methods are enumerative schemes for solving optimization problems The utility of the method derives from the fact that, in general, only a small fraction of the possible solutions need actually be enumerated, the remaining solutions being eliminated from consideration through the application of bounds that establish that such solutions cannot be optimal

The name 'branch and bound' arises from the two basic operations

- *Branching*, which consists of dividing collections of sets of solutions into subsets
- *Bounding*, which consists of establishing bounds on the value of the objective function over the subsets of solutions

The branch-and-bound procedure involves recursive application of the branching and bounding operations, with provision made for deleting subsets known not to contain an optimal solution

In subsequent sections, the basic elements of branch-and-bound methods are outlined, the required branching and bounding functions are defined, and the recursive procedure is introduced as a rather simple set function Next, some of the important properties of branch-and-bound methods will be developed, and, finally, several illustrations of applications of branch-and-bound techniques will be presented

The present development is based on reference 5, but extends previous work in several ways The explicit introduction of a lower bound on the value of the optimal solution represents a generalization of current methods and has potential for improving the efficiency of computational procedures In addition, sets are not required to be finite and branching is not confined to a single set Finally, some of the properties of the recursive procedure are more general than those previously mentioned

BASIC ELEMENTS

LET S be a given arbitrary set whose elements are called *feasible solutions* and let f $S \rightarrow R$ be a specified real-valued (return) function and let

$$\sup_{x \epsilon S} f(x) = f^*$$

The objective in the basic optimization problem is to find the set of *optimal feasible solutions* $S^* = \{x^* | x^* \epsilon S \text{ and } f(x^*) = f^*\}$ In anomalous situations it may happen that $S = \phi$ or that f^* is not attained in S, in these cases, S^* will be empty

To facilitate the branching-and-bounding operations, the original problem is frequently embedded in a larger problem involving less restrictive assumptions (e g , an integer restriction in a programming problem may be relaxed to permit easier calculation of bounds) For this purpose a non-empty superset T (with $S \subset T$) is introduced along with a bounded extension g $T \rightarrow R$ of the function f with the requirements that (a) $g(x) = f(x)$ whenever $x \epsilon S$, and (b) there exists an $x \epsilon T$ such that $g(x) = f^*$ A good bit of the art of applying branch-and-bound methods consists of choosing T and g so as to yield computationally efficient procedures

Branch-and-bound methods deal with subsets of T and with collections of subsets of T To introduce these elements let τ be the set of all subsets of T and let $\boldsymbol{T}$ be the set of all collections of subsets of T—i e , $\tau = 2^T$ and $\boldsymbol{T} = 2^\tau$ Thus, any $T_i \epsilon \tau$ is a subset of T and any $\boldsymbol{t} \epsilon \boldsymbol{T}$ is a collection of subsets of T For brevity, a collection of subsets $\boldsymbol{t}$ will sometimes be referred to simply as a 'collection '

In the following, $\{x\}$ will always denote the set containing the single element x—i e , $\{x\}$ is a singleton subset As a matter of notational convenience the union of all the subsets in any collection $\boldsymbol{t}$ will be denoted by $U(\boldsymbol{t})$—i e, $U(\boldsymbol{t})=\bigcup T_i$ where the union is over all subsets $T_i\epsilon\boldsymbol{t}$

An important element in subsequent developments is the set $\boldsymbol{T}^*=\{\boldsymbol{t}^*|\boldsymbol{t}^*\epsilon\boldsymbol{T}$ and $U(\boldsymbol{t}^*)=S^*\}$, which consists of all collections of subsets whose elements include all optimal feasible solutions The aim of the branch-and-bound procedure is to identify explicitly one of the elements of $\boldsymbol{T}^*$ If $S^*=\phi$, then, of course, $\boldsymbol{T}^*=\phi$

In order to be able to eliminate nonfeasible solutions it is necessary that such solutions be identified This is basically a requirement imposed on the computational procedure employed and is introduced explicitly through the following assumptions

(I) *The computational procedure specifies a collection* $\boldsymbol{t}_0$ *with the following properties (a) The elements of* $\boldsymbol{t}_0$ *contain only nonfeasible solutions—i e* $U(\boldsymbol{t}_0)\subset T-S$, *and (b) all singleton nonfeasible subsets are included in* $\boldsymbol{t}_0$—*i e, if* $x\epsilon T-S$, *then* $\{x\}\ \epsilon\boldsymbol{t}_0$

It is assumed, of course, that $\boldsymbol{t}_0$ is well defined so that for any subset $T_i\subset T$ it can be determined whether or not $T_i\epsilon\boldsymbol{t}_0$ Note that (b) is the minimum prescription—i e, given a solution $x\epsilon T$, it must be possible to determine whether it is feasible or nonfeasible If it happens that methods are available for determining whether subsets of more than one element contain all nonfeasible solutions, then $\boldsymbol{t}_0$ will contain more subsets than the minimum prescribed in (b), and it may be possible to improve the efficiency of the branch-and-bound procedure as a consequence

BRANCHING

THE BRANCHING operation divides the elements of a given collection into subsets Necessary properties of the branching operation are given below, but responsibility for devising specific rules for a particular situation falls on the analyst

(II) *The branching rule is a function* $\beta\ \boldsymbol{T}\rightarrow\boldsymbol{T}$ *with the following properties (a)* $U[\beta(\boldsymbol{t})]=U(\boldsymbol{t})$, *(b)* $T_i'\epsilon\beta(\boldsymbol{t})$ *only if* $T_i'\subset T_i\epsilon\boldsymbol{t}$, *and (c)* $\beta(\boldsymbol{t})\neq\boldsymbol{t}$ *if and only if* $\boldsymbol{t}$ *contains a subset* T_i *consisting of more than one element*

Conditions (a) and (b) state that the branching rule β divides the elements of the collection $\boldsymbol{t}$ into subsets which collectively include precisely the same points as the original collection $\boldsymbol{t}$ Condition (c) ensures that at least one divisible subset in $\boldsymbol{t}$ (if such exists) is actually divided into proper subsets

BOUNDING

BRANCH-AND-BOUND methods employ two types of bounds—a lower bound on f^* and upper bounds on the value of $g(x)$ over subsets of T The necessary properties of these bounds are given below, but in each particular problem the specification of the method for computing the bounds is the responsibility of the analyst

(III) *The* **upper bounding rule** *is a function* B $\tau \to R$ *with the following properties* (a) $g(x) \leqq B(T_i)$ *for all* $x \epsilon T_i \subset T$, (b) $B(T_i') \leqq B(T_i)$ *if* $T_i' \subset T_i \subset T$, *and* (c) $B(\{x\}) = g(x)$

Condition (*b*) ensures that deleting points from subsets does not lead to higher upper bounds while condition (*c*) ensures that bounds on singleton subsets are not unnecessarily loose As a matter of notational convenience it is also assumed that $B(\phi) = -\infty$ (or an appropriately small finite value)

(IV) *The* **lower bounding rule** *is a function* b $\boldsymbol{T} \to R$ *with the following properties for any collection* $\boldsymbol{t}$ (*a*) $b(\boldsymbol{t}) \leqq f^*$, (*b*) $b[\beta(\boldsymbol{t})] \geqq b(\boldsymbol{t})$ (*c*) $b(\boldsymbol{t}) \geqq f(x)$ *if* $x \epsilon S$ *and* $\{x\} \epsilon \boldsymbol{t}$, *and* (*d*) $b(\boldsymbol{t}') = b(\boldsymbol{t})$ *if* $\boldsymbol{t}' \subset \boldsymbol{t}$, *and, for every* $T_i \epsilon \boldsymbol{t} - \boldsymbol{t}'$ *either* $T_i \epsilon \boldsymbol{t}_0$ *or* $B(T_i) < b(\boldsymbol{t})$

Condition (*b*) states that dividing sets into subsets cannot reduce the value of the lower bound $b(\boldsymbol{t})$ Condition (*c*) ensures that the lower bound on singleton subsets is not unnecessarily loose Condition (*d*) states that a subset T_i, which is known to be either nonfeasible (i e $T_i \epsilon \boldsymbol{t}_0$) or dominated [i e $B(T_i) < b(\boldsymbol{t})$], cannot affect the value of the lower bound $b(\boldsymbol{t})$

For any collection $\boldsymbol{t}$ the symbol $\boldsymbol{t}^-$ will be used to designate the subcollection containing those elements of $\boldsymbol{t}$ which are either dominanted or are known to be nonfeasible—i e , $\boldsymbol{t}^- = (\boldsymbol{t} \cap \boldsymbol{t}_0) \cup \{T_i | T_i \epsilon \boldsymbol{t}$ and $B(T_i) < b(\boldsymbol{t})\}$ Thus, no element of $\boldsymbol{t}^-$ can contain an optimal feasible solution

BRANCH AND BOUND

THE branch-and-bound recursive operation consists of forming new collections of subsets from which have been excluded those elements known not to contain an optimal feasible solution Specifically,

(V) *The* **branch and bound recursive operation** *is a function* $\boldsymbol{B}$ $\boldsymbol{T} \to \boldsymbol{T}$ *with the property that, if* $\beta(\boldsymbol{t}') = \boldsymbol{t}$, *then* $\boldsymbol{B}(\boldsymbol{t}') = \boldsymbol{t} - \boldsymbol{t}^-$

Thus, the branch-and-bound recursive function $\boldsymbol{B}$ uses the branching rule β to divide the elements of the collection $\boldsymbol{t}'$ into subsets Applying the bounding functions B and b to the new collection $\boldsymbol{t}$ defines the collection $\boldsymbol{t}^-$ of subsets of $\boldsymbol{t}$ that are either dominated or known to be nonfeasible Deleting the elements of $\boldsymbol{t}^-$ from $\boldsymbol{t}$ yields the new collection $\boldsymbol{B}(\boldsymbol{t}')$

For any collection $\boldsymbol{t}^1$, the recursive function $\boldsymbol{B}$ defines a sequence $\{\boldsymbol{t}^n\}_{n>0}$ with $\boldsymbol{t}^{n+1} = \boldsymbol{B}(\boldsymbol{t}^n)$ for $n>0$ Note that $\boldsymbol{B}(\phi) = \phi$, and so $\boldsymbol{t}^n = \phi$ for all $n>0$ if $\boldsymbol{t}^1 = \phi$

RESULTS

IN THIS section, relations between the branch-and-bound recurvise function $\boldsymbol{B}$ and the set of optimal feasible solutions S^* are established The proofs of the results involve rather straight-forward applications of conditions (I)-(V) and standard results from analysis, and are therefore omitted The results all hold for the anomalous sinutations in which $S^* = \phi$

The first result simply states that $\boldsymbol{B}$ retains optimal feasible solutions

(R1) *If* $S^* \subset U(\boldsymbol{t}^1)$, *then* $S^* \subset U(\boldsymbol{t}^n)$ *for all* $n < 0$

This may be strengthened by considering only collections $\boldsymbol{t}^1$ whose elements cover S

(R2) *If* $S \subset U(\boldsymbol{t}^1)$, *then* $S^* \subset U(\boldsymbol{t}^n)$ *for all* $n > 0$

The following result states the important fixed-point property of $\boldsymbol{B}$

(R3) *If* $\boldsymbol{t} \epsilon \{\boldsymbol{t}^n | S \subset U(\boldsymbol{t}^1)$ *and* $n > 0\}$ *and* $\boldsymbol{B}(\boldsymbol{t}) = \boldsymbol{t}$, *then* $\boldsymbol{t} \epsilon \boldsymbol{T}^*$, *further, if* $\boldsymbol{t} \epsilon \boldsymbol{T}^*$ *then* $\boldsymbol{B}(\boldsymbol{t}) \epsilon \boldsymbol{T}^*$

The result (R3) is of considerable practical significance, since it implies that, if recursive application of $\boldsymbol{B}$ (to an initial point whose elements cover S) yields a fixed point, then that fixed point is in the set of optimal collections $\boldsymbol{T}^*$

In general, assurance that the recursive application of $\boldsymbol{B}$ yields a collection in $\boldsymbol{T}^*$—either in a finite number of steps or in the limit—is possible only by introducing additional structure beyond conditions (I)-(V) Several possibilities are developed below

Most applications of branch-and-bound procedures reported to date have assured the finiteness of the procedure by exploiting the finiteness of some set The following result summarizes the common property of these procedures

(R4) *If* $S \subset U(\boldsymbol{t}^1)$ and $U(\boldsymbol{t}^m)$ *is finite for some finite* $m > 0$, *then* $\boldsymbol{t}^n \epsilon \boldsymbol{T}^*$ *for some finite* $n > 0$

More particularly, under the hypothesis of (R4), there will exist some finite n for which $\boldsymbol{t}^n = \boldsymbol{t}^* = \{\{x^*\} | x^* \epsilon S^*\}$ and $\boldsymbol{B}(\boldsymbol{t}^*) = \boldsymbol{t}^*$—i e, in a finite number of steps the recursive procedure will reach a fixed point $\boldsymbol{t}^*$ that will exhibit all optimal feasible solutions in the form of singleton subsets $\{x^*\}$

To obtain more general conditions under which $\{\boldsymbol{t}^n\}_{n>0}$ converges to some optimal collection, use may be made of the fact that $\boldsymbol{t}' = \boldsymbol{B}(\boldsymbol{t})$ represents an 'improvement' over $\boldsymbol{t}$ if either the bounds for $\boldsymbol{t}'$ are in some sense tighter than for $\boldsymbol{t}$, or the collection $\boldsymbol{t}'$ is in some sense smaller than the collection $\boldsymbol{t}$ The following notation will be required to make these ideas more precise If η is a subsequence of the positive integers (in their natural order), then $\{\boldsymbol{t}^n\}_\eta$ will represent the subsequence of $\{\boldsymbol{t}^n\}_{n>0}$ whose elements are indexed by η, and, if n is an element in η, then n' will denote the next larger element in η

As a measure of the improvement in bounds, let $\lambda(\boldsymbol{t}) = \sup_{T_i \epsilon \boldsymbol{t}} B(T_i) - b(\boldsymbol{t})$, so that $\lambda(\boldsymbol{t}^*) = 0$ if $\boldsymbol{t}^* \epsilon \boldsymbol{T}^*$ The result (R5) is obtained by noting that $\rho(\boldsymbol{t}^m, \boldsymbol{t}^n) = |\lambda(\boldsymbol{t}^m) - \lambda(\boldsymbol{t}^n)|$ is, under the stated conditions, a metric for the set $\{\boldsymbol{t}^n | \boldsymbol{t}^n$ is in $\{\boldsymbol{t}^n\}_\eta\}$

(R5) *If* $S \subset U(\boldsymbol{t}^1)$, *then* $\{\boldsymbol{t}^n\}_{n>0}$ *converges to some* $\boldsymbol{t}^*$ *in* $\boldsymbol{T}^*$ *if there exists a sequence* η *such that* $\lambda(\boldsymbol{t}^n)'/\lambda(\boldsymbol{t}^n)<\delta$ *for some fixed* $\delta<1$ *and all* n *in* η

Turning to the result based on the size of the collections $\boldsymbol{t}^n$, we may note that difficulties may be encountered in finding nontrivial measures defined over the entire sets τ and $\boldsymbol{T}$ To simplify matters, attention may be restricted to a σ-field $\tau'(\subset\tau)$ and the associated set $\boldsymbol{T}'(\subset\boldsymbol{T})$ consisting of all collections of countable unions of the subsets $T_i\epsilon\tau'$ For any completely additive measure μ defined over a σ-field τ', let $\omega(\boldsymbol{t})=\mu[U(\boldsymbol{t})]-\mu[U(\boldsymbol{t}^*)]$ for any $\boldsymbol{t}^*\epsilon\boldsymbol{T}^*\cap\boldsymbol{T}'$ Then (R6) is established using the fact that, under the stated hypotheses, $\rho'(\boldsymbol{t}^m, \boldsymbol{t}^n)=|\omega(\boldsymbol{t}^m)-\omega(\boldsymbol{t}^n)|$ is a metric for the set $\{\boldsymbol{t}^n|\boldsymbol{t}^n \text{ is in } \{\boldsymbol{t}^n\}_\eta\}$

(R6) *If* $S\subset U(\boldsymbol{t}^1)$, *then* $\{\boldsymbol{t}^n\}_{n>0}$ *converges to some* $\boldsymbol{t}^*\epsilon\boldsymbol{T}^*$ *if there exists a sequence* η, *a* σ-*field* $\tau'\subset\tau$, *and a completely additive measure* μ *defined on* τ' *satisfying the requirements* (a) $\boldsymbol{t}^n\epsilon\boldsymbol{T}'$ *for all* n *in* η, (b) $\omega(\boldsymbol{t}^{n'})/\omega(\boldsymbol{t}^n)<\delta$ *for some fixed* $\delta<1$ *and for all* n *in* η *for which* $\omega(\boldsymbol{t}^n)>0$, *and* (c) $\mu(T_i)>0$ *if* T_i *is an infinite set*

Condition (c) could be weakened to allow $\mu(T_i)=0$ for any $T_i\subset S^*$ if this would simplify the task of finding the required measure

Other results may be obtained by postulating various analytic or topological conditions—e g , continuity, completeness, compactness, etc Two examples will perhaps suffice

Standard fixed-point convergence theorems and an appropriate metric structure yield the following result

(R7) *If* $\boldsymbol{B}$ *is a contraction mapping in a complete metric space including all the elements of* $\{\boldsymbol{t}^n\}_{n>0}$, *then* $\{\boldsymbol{t}^n\}_{n>0}$ *converges to some* $\boldsymbol{t}^*\epsilon\boldsymbol{T}^*$

General convergence theorems for algorithms provide an alternative approach For example, $\boldsymbol{B}$ satisfies ZANGWILL's conditions for an algorithmic map, plus a number of other hypotheses for his convergence theorems—*see* (reference 27) The type of assumptions required are illustrated by the following application of Zangwill's Convergence Theorem A to the branch-and-bound algorithm

(R8) *If the following conditions are satisfied, then the limit of* $\{\boldsymbol{t}^n\}_{n>0}$ *is some* $\boldsymbol{t}^*\epsilon\boldsymbol{T}^*$ (a) $S\subset U(\boldsymbol{t}^1)$, ($b$) *the points of* $\{\boldsymbol{t}^n\}_{n>0}$ *are in a compact set*, (c) $\boldsymbol{B}$ *is continuous at all the points of* $\{\boldsymbol{t}^n\}_{n>0}$, *except possibly at points* $\boldsymbol{t}^*\epsilon\boldsymbol{T}^*$, *and* ($d$) *there exists a continuous function* Z $\boldsymbol{T}\to R$ *for which* $Z(\boldsymbol{t}^{n+1})<Z(\boldsymbol{t}^n)$ *for any* $\boldsymbol{t}^n\notin\boldsymbol{T}^*$

Possible choices for the function $Z(\boldsymbol{t})$ might be $\lambda(\boldsymbol{t})$ or $\omega(\boldsymbol{t})$, as defined above

The following final result can be useful in evaluating near-optimal solutions

(R9) *If* $x\epsilon S$ *and* $\{x\}$ $\epsilon\boldsymbol{t}^n$ *and* $S\subset U(\boldsymbol{t}^1)$, *then* $f(x)\leqq f^*\leqq\sup_{T_i\epsilon\boldsymbol{t}^n}B(T_i)$

ILLUSTRATION—DISCRETE PROGRAMMING

TO ILLUSTRATE the concepts developed in the preceding sections, their application to a broad class of optimization problems—discrete programming—will be presented Discrete programming includes integer programming and combinatorial optimization problems, among others Much of the impetus for the development of branch-and-bound methods has arisen from applications to specific discrete programming problems

Let $Z_1, \ldots, Z_N$ be a finite number of finite sets and let $T = X_{i=1}^{N} Z_i$ be their Cartesian product so that any $x \epsilon T$ is an N-tuple $x = (x_1, \ldots, x_N)$ with each $x_i \epsilon Z_i$ The discrete programming problem may be defined as follows Maximize $f(x)$ over all $x \epsilon S \subset T$

The set S is defined by stating suitable restrictions on the elements of T For example, in integer programming, each $Z_i = \{x_i | a_i \leqq x_i \leqq b_i$ and x_i an integer$\}$ and the set S is defined by $S = \{x | x \epsilon T$ and $H(x) \leqq c\}$ for a given vector-valued function $H\ T \rightarrow E^M$ In combinatorial optimization problems, on the other hand, $Z_1 = \cdots = Z_N = Z$ and $S = \{x | x \epsilon T$ and $x_i \neq x_j$ if $i \neq j\}$—i e , S consists of the set of all permutations of the elements of the set Z

The branching operations take place in a tree structure consisting of nodes connected by directed links Each node is identified with a 'partial solution' represented by an m-tuple $y = (y_1, \ldots, y_m)$ with $m \leqq N$ and each $y_i \epsilon Z_i$ Except when $m = N$, directed links from node y connect it to a set of successor nodes, each of which is represented by an $(m+1)$-tuple of the form $y' = (y_1, \ldots, y_m, y'_{m+1})$ for each $y'_{m+1} \epsilon Z_{m+1}$

Associated with each node $y = (y_1, \ldots, y_m)$ is a subset $T_y = \{x | x \epsilon T$ and $x_1 = y_1, \ldots, x_m = y_m\}$, only collections $\boldsymbol{t}$ consisting of subsets of the form T_y are considered Starting with $\boldsymbol{t}^1 = \{T\}$, for each $n > 0$ the collection $\beta(\boldsymbol{t}^n)$ is obtained by replacing a subset $T_y \epsilon \boldsymbol{t}^n$ by the proper subsets $T_{y'}$ associated with the successor nodes y' The selection of the branch node y is arbitrary, although several heuristics have been proposed and investigated empirically

The calculation of bounds in discrete programming depends on the particular form of the objective function $g\ T \rightarrow R$ However, some of the possibilities may be illustrated by two examples

In integer programming the upper bound for a subset T_y may be calculated as the solution to a standard programming problem without integer constraints Specifically, for given $y = (y_1, \ldots, y_m)$ with $0 < m \leqq N$, the upper bound $B(T_y) = g(x')$ where x' is a solution to the problem Maximize $g(x)$ over all $x \epsilon E^N$ satisfying $x_i = y_i$ for $i = 1, \ldots, m$, $a_i \leqq x_i \leqq b_i$ for $i = m+1, \ldots, N$, and $H(x) \leqq c$

In combinatorial optimization problems, condition (IVc) can frequently be used to obtain a useful initial lower bound on f^* To do so, the recursive process starts with $\boldsymbol{t}^1=\{T,\{x\}\}$, where $x\epsilon S$ is any 'reasonably good' permutation Possible sources of such a reasonably good x are (1) The best permutation in a collection of randomly generated permutations, (2) a conjectured optimal permutation, or (3) the historically best permutation, in the case of a realized process Such a permutation x provides an immediately available lower bound $b(\boldsymbol{t})=f(x)$ for use in deleting dominated subsets

Other types of discrete programming problems may present other special structures that can be exploited to yield good (and readily calculated) upper and lower bounds

The result (R4) ensures that in discrete programming problems the set of all optimal solutions will be enumerated in a finite number of steps

ILLUSTRATION—SEQUENTIAL UNIMODAL SEARCH

As a final exercise, sequential unimodal search methods[26] will be shown to be an interesting class of branch-and-bound procedures—a fact apparently not previously recognized Neither the procedure nor the sets involved is finite (in fact, the sets are uncountable), and, in addition, this is perhaps the only currently known application of branch-and-bound methods employing a branching rule that can be demonstrated to be optimal

In sequential unimodal search, $S=\{x|x\epsilon R \text{ and } a_1<x<b_1\}$ is a bounded real interval and $f\ S\rightarrow R$ is a bounded unimodal function in the open interval (a_1,b_1)—i e , if $x<y<z$ are three points in S, then $f(y)>\min\{f(x), f(z)\}$ In this problem, $T=S$ and so $\boldsymbol{t}_0=\phi$ The optimal solution x^* exists and is unique, so that $\boldsymbol{T}^*$ contains only one element, $\boldsymbol{t}^*=\{\{x^*\}\}$

The following property of unimodal functions will be of importance in subsequent developments If $y<z$ are two points in an open interval (a,b) in S, then y and z define three open subintervals—a center interval $I_c=(y,z)$, an outer interval adjacent to y, $I(y)=(a,y)$, and an outer interval adjacent to z, $I(z)=(z,b)$ Then, by the unimodal property of f, the following relations hold if $f(y)<f(z)$ then $f(y)>f(x)$ for all $x\epsilon I(y)$, and if $f(y)>f(z)$ then $f(z)>f(x)$ for all $x\epsilon I(z)$ Thus, this property provides an upper bound over either $I(y)$ or $I(z)$, depending on the relative values of $f(y)$ and $f(z)$

The branch-and-bound procedure recursively generates the sequence $\{\boldsymbol{t}^n\}_{n>1}$ from a sequence $\{x^n\}_{n\geq 0}$ of points in S, starting from $x_0=a_1$ and $x_1=b_1$ and a value of x_2 specified (arbitrarily) by the procedure Throughout the following, take $n>1$

Given the set $X_n=\{x_0, \quad , x_n\}$, define the element $u_n\epsilon X_n$ by $f(u_n)=\max_{1<j\leq n} f(x_j)$, and, further, let (a_n,b_n) be the shortest open interval such that $a_n<u_n<b_n$ with $a_n\epsilon X_n$ and $b_n\epsilon X_n$ For given X_n and the current col-

lection $t^n = \{(a_n, u_n), \{u_n\}, (u_n, b_n)\}$, the branching rule prescribes a point $x_{n+1} \epsilon (a_n, b_n)$ with $x_{n+1} \neq u_n$

Then the two points u_n and x_{n+1} in (a_n, b_n) define three open subintervals—the center interval I_c between u_n and x_{n+1}, the outer interval $I(u_n)$ adjacent to u_n, and the outer interval $I(x_{n+1})$ adjacent to x_{n+1} The branching rule then yields the following new collection $\beta(t^n) = \{I(u_n), \{u_n\}, I_c, \{x_{n+1}\}, I(x_{n+1})\}$ The lower bound is $b[\beta(t^n)] = \max\{f(u_n), f(x_{n+1})\}$, and the previously stated property of unimodal functions may be used to calculate upper bounds The bounds show that $I(u_n)$ and $\{u_n\}$ are dominated if $f(u_n) < f(x_{n+1})$, and that $I(x_{n+1})$ and $\{x_{n+1}\}$ are dominated otherwise Thus, $\boldsymbol{B}(t^n) = \{I(x_{n+1}), \{x_{n+1}\}, I_c\}$ if $f(u_n) < f(x_{n+1})$, or $\boldsymbol{B}(t^n) = \{I(u_n), \{u_n\}, I_c\}$ if $f(u_n > f(x_{n+1})$, if $f(u_n) = f(x_{n+1})$, either alternative is permissible

Since all the intervals considered are open, the above procedure will not converge to S^* in a finite number of steps However, under very mild conditions, the sequence $\{t^n\}_{n>0}$ does converge to $t^* = \{S^*\}$ in the limit For example, in (R6) the measure of the open interval (a, b) may be taken to be $\mu(a, b) = b - a$ Since S^* contains only a single point, $\mu(S^*) = 0$ and so $\omega(t^n) = b_n - a_n$ Thus condition (b) can be met easily—for example if u_n and x_{n+1} are located so as to divide the interval (a_n, b_n) into fixed proportions

Fibonacci search[26] provides branching rules for the current problem that can be shown to be optimal in the sense that, for fixed $N > 0$, they minimize the maximum possible length of the interval (a_N, b_N)

To identify sequential unimodal search methods with branch-and-bound procedures is doubly illuminating The existence of an optimal branching rule for this problem suggests some interesting avenues of investigation in other areas of application (where branching rules have typically been heuristic) On the other hand, attempts to extend sequential search methods to the broader domain $R^n (n > 1)$ have been notably unsuccessful to date Identification of current methods (with domain R) with branch-and-bound procedures provides a new perspective in which to view possibilities for such an extension

REFERENCES

1 N AGIN "Optimum Seeking with Branch-and-Bound," *Management Sci* **13,** B176–B185 (1966)

2 E BALAS, "Programare liniară cu variable bivalent" ("Linear Programming with Zero-One Variables"), *Proc Third Scientific Session on Statistics*, Bucharest, December 5–7, 1963

3 ———, "An Additive Algorithm for Solving Linear Programs with Zero-One Variables," *Opns Res* **13,** 517–526 (1965)

4 ———, "Finding a Minimaximal Path in a Disjunction PERT Network," Actes des journees *Thèorie des graphs et ses applications, ICC*, Rome (1966)

5 ———, "A Note on the Branch-and-Bound Principle," *Opns Res* **16**, 442–445 (1968), (Errata p 886)

6 P BERTIER AND B ROY, "Procédure de résolution pour une classe de problèmes pouvant avoir un charactère combinatoire," *Cahiers du Centre d'Études de Recherche Operationelle* **6**, 202–208 (1964) *Also* Note de Travail No 30 bis, Direction Scientific, SEMA, Paris, Dec 1964 Translated by Wm S Jewell *Operations Research Center Report 67-34* (University of California, Berkeley, California)

7 G H BROOKS AND C P WHITE, "An Algorithm for Finding Optimal and Near Optimal Solutions to the Production Scheduling Problem," *J Indus Eng* **16**, 34–40 (1965)

8 R J DAKIN, "A Tree-search Algorithm for Mixed Integer Programming Problems," *The Computer Journal* **8**, 250–255 (1965)

9 N J DRIEBEEK, "An Algorithm for the Solution of Mixed Integer Programming Problems," *Management Sci* **12**, 576–587 (1966)

10 M A EFROYMSON AND T L RAY, "A Branch-and-Bound Algorithm for Plant Location," *Opns Res* **14**, 361–368 (1966)

11 J W FAVETT AND N PLYTER, "The Optimal Assignment of Facilities to Locations by Branch and Bound," *Opns Res* **14**, 210–232 (1966)

12 A M GEOFRION, "Integer Programming by Implicit Enumeration and Balas' Method," *SIAM Rev* **9**, 178–190 (1967)

13 P C GILMORE, "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem," *J Soc Indust and Appl Math* **10**, 305–313 (1962)

14 E IGNALL AND L SCHRAGE, "Application of the Branch-and-Bound Techniques to Some Flow Shop Scheduling Problems," *Opns Res* **13**, 400–412 (1965)

15 P J KOLESAR, "A Branch-and-Bound Algorithm for the Knapsack Problem," *Management Sci* **13**, 723–735 (1967)

16 A H LAND AND A DOIG, "An Automatic Method for Solving Discrete Programming Problems," *Econometrica* **28**, 497–520 (1960)

17 E L LAWLER, "The Quadratic Assignment Problem," *Management Sci* **9**, 586–599 (1963)

18 ——— AND D E WOOD, "Branch-and-Bound Methods A Survey," *Opns Res* **14**, 669–719 (1966)

19 J D C LITTLE, K G MURTY, D W SWEENEY, AND C KAREL, "An Algorithm for the Traveling Salesman Problem," *Opns Res* **11**, 972–989 (1963)

20 Z A LOMNICKI, "A Branch-and-Bound Algorithm for the Exact Solution of the Three-Machine Scheduling Problem," *Opnal Res Quart* **16**, 89–100 (1965)

21 W L PRICE, "Increasing the Capacity of a Network Where the Costs Are Nonlinear A Branch-and-Bound Algorithm," *CORS Journal* **5**, 100–114 (1967)

22 F RADÓ, "Programare liniară cu condiții logice" ("Linear Programming with Logical Conditions"), *Communicarile Academiei R P R* **13**, 1039–1041 (1963)

23 L SCHRAGE AND S WOILER, *A General Structure for Implicit Enumeration*, Report, Department of Industrial Engineering, Stanford University, Stanford, California

24 J W STROUP, "Allocation of Launch Vehicles to Space Missions A Fixed-Cost Transportation Problem," *Opns Res* **15,** 1157–1163 (1967)

25 R S WALKER, "An Enumerative Technique for a Class of Combinatorial Problems," *American Mathematics Society Symposium on Applied Mathematics Proceedings* **10,** pp 91–94, 1960

26 D J WILD, *Optimum Seeking Methods*, Prentice Hall, Englewood Cliffs, N J, 1964

27 W I ZANGWILL, *Non-Linear Programming A Unified Approach*, Prentice Hall, Englewood Cliffs, N J, 1969