European Journal of Operational Research 56 (1992) 357–363
North-Holland

Theory and Methodology

Multiobjective, preference-based search in acyclic OR-graphs *

Chelsea C. White, III
Department of Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI 48109, USA

Bradley S. Stewart
BELLCORE, Operations System Planning Center, Morristown, NJ 079, USA

Robert L. Carraway
Darden School of Business Administration, University of Virginia, Charlottesville, VA 22901, USA

Received May 1989; revised November 1989

Abstract: We consider the problem of determining a most preferred path from a start node to a goal node set in an acyclic OR-graph, given a multiattribute preference function, a multiobjective reward structure, and heuristic information about this reward structure. We present an algorithm which is shown to terminate with a most preferred path, given an admissible heuristic set. The algorithm illustrates how Artificial Intelligence techniques can be productively employed to solve multiobjective problems.

Keywords: Artificial intelligence, computation analysis, decision theory, heuristics, programming: multiple criteria

1. Introduction

Decision making often involves the consideration of multiple, conflicting, and noncommensurate objectives. Decision making techniques that have been developed to explicitly consider multiple objectives include the determination of the set of all *nondominated* decisions (Geoffrion, 1968), the use of a multiattribute *preference function* (Keeney and Raiffa, 1976), and various interactive techniques (e.g., Zionts and Wallenius, 1977). The interactive techniques are usually based on concepts associated with nondominated decisions and a, perhaps implicit, preference function.

It is possible to represent many real-world decisions problems – particularly those involving a sequence of interrelated decisions – as what is referred to in the Artificial Intelligence community as a *finite OR-graph* (Pearl, 1984). Here,

* This research has been supported by ARO Contract DAAG 29-85-K0089, NSF Grant ECS8708183, and JPL Contract 957721.

'OR' refers to the fact that if a path in the graph reaches a particular node, then extending that path beyond that node requires selecting precisely one of the arcs emanating from the node. (When a graph is such that a path can be defined as including more than one arc emanating from the same node, then it is called an AND-graph.) In a finite OR-graph, a nondominated path can be described as follows. Assume $r(P)$ is the vector of rewards accrued by path P, each scalar element of $r(P)$ corresponds to the attribute score of a particular objective, and 'more is better' for all attributes. We say path P is *nondominated* if there does not exist a path P' such that $r(P') \geq r(P)$ and $r(P') \neq r(P)$.

A preference function u provides a scalar-valued measure of preference based on rewards. We say reward vector r is at least as preferred as reward vector r' if and only if $u(r) \geq u(r')$. In order to be consistent with the 'more is better' assumption, u is assumed to be isotone (monotonically nondecreasing); i.e., $r \geq r'$ implies $u(r) \geq u(r')$. Thus, candidates for the most preferred reward vector are the nondominated reward vectors.

In this paper, we consider the problem of finding the most preferred path from the start node to a set of goal nodes in a finite OR-graph, where:

1. There is a vector of rewards associated with each arc in the graph.
2. The reward vector associated with a path is not necessarily the sum of the reward vectors of the arcs that comprise the path.
3. Path preference is based on a multiattribute preference function.

We present a so-called 'best-first' search algorithm, referred to as U*, that searches for a most preferred path using estimates of path reward vectors. These estimates are referred to in the Artificial Intelligence literature as *heuristics*. When these heuristics are used in an algorithm, such as U*, the result is a procedure that appears to be hybrid of branch-and-bound and dynamic programming algorithms commonly encountered in the Operations Research literature. In the following work, we represent the reward vector estimates by a set of vectors referred to as a *heuristic set*.

One solution approach is to determine the set of all nondominated paths (perhaps using GOZ*, Stewart, 1988) and then to rank these paths using the preference function. A disincentive to this approach is the (likely) possibility that for large problems the number of nondominated paths is large and their determination is computationally intensive.

Another approach is conventional dynamic programming. However, dynamic programming requires that the preference function satisfy the following monotonicity assumption in order to be guaranteed to find the most preferred path. Let s be the start node, Γ be the set of goal nodes, and n be an intermediate node. Let P_1 and P_2 be two different paths from s to n, and let P_3 be any path from n to Γ. Let $P_1 \cup P_3$ be the path from s to Γ through n following path P_1 and then path P_3, and define $P_2 \cup P_3$ analogously. The monotonicity assumption is: if $u[r(P_1)] \geq u[r(P_2)]$, then $u[r(P_1 \cup P_3)] \geq u[r(P_2 \cup P_3)]$. Carraway et al. (1990) present examples of preference functions that do not satisfy this assumption and as a result guide the standard dynamic programming algorithm to a suboptimal path.

Carraway et al. (1990) present a concept – generalized dynamic programming – that avoids this pitfall, in the process further blurring the distinction between the traditional Operations Research concepts of branch-and-bound and dynamic programming.

In this paper, we provide an Artificial-Intelligence based perspective on the same problem studied in Carraway et al. (1990). Our U* algorithm – and in particular our use of heuristic information – provides a more general framework within which the positive characteristics of branch-and-bound and dynamic programming can be combined, and hence approaches like generalized dynamic programming can be implemented. Heuristic information is represented in the form of a set, rather than a function, which has been inspired by the problem considered by Lark and White (1988). Our interest in adopting a heuristic search algorithm to the problem considered in this paper is due to the fact that a well-studied heuristic search algorithm A* has been shown to be significantly more computationally efficient than dynamic programming for a variety of applications (see Pearl, 1984).

This paper is outlined as follows. The problem is formulated in Section 2 and the U* algorithm is presented in Section 3. In Section 4 we present a condition on the heuristic set which guarantees that upon termination, U* will have found a most preferred path. An example presented in Carraway et al. (1990) is examined in Section 5. Conclusions are presented in Section 6.

2. Problem formulation

Assume N is a finite set of *nodes*, $A \subseteq N \times N$ is a set of *directed arcs*, where $(n, n') \in A$ is the arc from node n to node n', $s \in N$ is the *start node*, and $\Gamma \subseteq N$ is the *goal node set*. Let SCS: $N \to 2^N$ be the successor set function, i.e., $\text{SCS}(n) = \{n' \in N : (n, n') \in A)$, and let $\text{SCS}^{-1}(n') = (n \in N : (n, n') \in A\}$. A *path* $P = (n_1, \ldots, n_K)$ is a sequence of two or more nodes such that $(n_k, n_{k+1}) \in A$ for all $k = 1, \ldots, K-1$. Let $\mathscr{P}$ be the set of all paths in the graph, and assume that the graph is *acyclic*, i.e., $n_1 \neq n_K$ for all $(n_1, \ldots, n_K) \in \mathscr{P}$. Let $\mathscr{P}(n, S) \subseteq \mathscr{P}$ be the set of all paths beginning at $n \in N$ and ending in $S \subseteq N$; if the set S is a singleton, then let $\mathscr{P}(n, S) = \mathscr{P}(n, n')$, where $S = \{n'\}$. For simplicity, we assume throughout that for all $n \notin \Gamma$, $\mathscr{P}(n, \Gamma) \neq \phi$, where ϕ is the null set. $\mathscr{P}(s, \Gamma)$ is the set of *solution paths*.

The graph has the following multicriteria reward structure. Let M be the number of criteria or objectives under consideration. There is a *reward function* r: $\mathscr{P} \to \mathrm{R}^M$ and a commutative binary operator '$\circ$' on R^M such that, for any path P_3 comprised of two subpaths P_1 and P_2, $r(P_3) = r(P_1) \circ r(P_2)$. Furthermore, the binary operator $\circ$ is *order-preserving*; i.e., $r_1 \leq r_2$ implies $r_1 \circ x \leq r_2 \circ x$, for $r_1, r_2 x \in \mathrm{R}^M$. Let e be the identity vector for $\circ$; i.e., e is such that $e \circ r = r$ for all r. Each scalar element of $r(P)$ corresponds to the attribute score of a particular objective, and 'more is better' for all attributes. We remark that since $A \subseteq \mathscr{P}$, each arc in the graph has associated with it a reward vector.

With respect to preference structure, there is a multiattribute *preference function* u: $\mathrm{R}^M \to \mathrm{R}$. We assume that u is isotone. If $P^* \in \mathscr{P}(n, S)$ is such that $u[r(P^*)] \geq u[r(P)]$ for all $P \in \mathscr{P}(n, S)$, then we say that P^* is a *most preferred* path in $\mathscr{P}(n, S)$. Let $\mathscr{P}^*(n, S)$ be the set of all most preferred paths in $\mathscr{P}(n, S)$. Our objective is to determine a most preferred path in $\mathscr{P}(s, \Gamma)$, i.e., a path in $\mathscr{P}^*(s, \Gamma)$.

We assume heuristic information is available to help guide the search for a path in $\mathscr{P}^*(s, \Gamma)$. Let $H \subseteq \mathrm{R}^L$, where $L = M \times \#N$, $\#N$ being the cardinality of the set N. We call H the *heuristic set*. We think of each element $h \in H$ as being a collection $h = \{h(n) : n \in N\}$ of reward vector estimates, where $h(n) \in \mathrm{R}^M$ is an estimate of $r(P)$ for some $P \in \mathscr{P}(n, \Gamma)$.

3. The U* algorithm

We now present a best-first algorithm for determining a path in $\mathscr{P}^*(s, \Gamma)$. Let LABEL(n', n) be the set of nondominated accrued costs of paths starting at s and having arc (n, n') as the last arc on the path. Let $G(s) = \{e\}$, and for all $n' \neq s$, let $G(n')$ be the set of all nondominated accrued costs from s to n' at a given stage in the search. That is, G(n') is the nondominated subset of $\{\bigcup \text{LABEL}(n', n) : n \in \text{SCS}^{-1}(n')\}$.

Step 0. *Initialization.*
Set OPEN $= \{s\}$, $G(s) = \{e\}$, and LABEL$(n', n) = \phi$ for all $(n, n') \in A$, so that $G(n) = \phi$ for all $n \in N \sim \{s\}$, where $\sim$ is the set difference operator.

Step 1. If OPEN $= \phi$, then terminate with failure.

Step 2. *Node selection for expansion.*
Select $n' \in$ OPEN for expansion such the IE$(n') \geq$ IE(n) for all $n \in$ OPEN, where

$$\text{IE}(n) = \sup\{u[g \circ h(n)] : g \in G(n), h \in H\}.$$

If $n' \in \Gamma$, then perform a final culling step on $G(n')$, use the LABEL sets to trace back from n' to s in order to determine a path in $\mathscr{P}(s, n')$, and terminate successfully. If $n' \notin \Gamma$, then remove n' from OPEN and add all successors of n' to OPEN; i.e.,

$$\text{OPEN} = \{\text{OPEN} \sim \{n'\}\} \cup \text{SCS}(n').$$

For all $n'' \in \text{SCS}(n')$, let

$$G(n'') = G(n'') \cup \{r(n', n'') \circ g' : g' \in G(n')\}.$$

Step 3. *Reward vector culling.*
Remove g' from all label sets LABEL$(n', \cdot)$,

$n' \in$ OPEN, if there exists a $g'' \in G(n'')$, $n'' \in$ OPEN, such that

$$u[g' \circ h(n')] < u[g'' \circ h(n'')]$$

for all $h \in H$. If, during the culling process, $G(n')$ becomes null for any $n' \in$ OPEN, then remove n from OPEN. Go to Step 1.

Assume that the performance function u is such that $g' \le g''$ and $g' \ne g''$ implies $u(g') < u(g'')$. Then assuming that $g', g'' \in G(n')$, $n' \in$ OPEN, $g' \le g''$ and $g' \ne g''$, it is easy to show that g' will be culled from all sets LABEL $(n', \cdot)$, and therefore from $G(n')$. Thus, $G(n')$ will contain only nondominated elements after completion of the culling process.

4. Admissibility

We now present a condition, the admissibility of the heuristic set, that guarantees the admissibility of U*. U* is *admissible* if it terminates having identified a path in $\mathscr{P}^*(s, \Gamma)$. Thus, when we demonstrate that U* is admissible, we show that it produces optimal solutions in spite of the heuristic nature of the information it uses. The heuristic set H is *admissible* (with respect to U*) if there is a $h^* \in H$, $h^* = \{h^*(n): n \in N\}$, that satisfies the following conditions:

1. For all $n \in N$, there exists a path $P \in \mathscr{P}^*(n, \Gamma)$ such that $h^*(n) \le r(P)$.

2. There is a path such that equality holds in Condition 1; i.e., there is a path $(n_1^*, \ldots, n_K^*) \in \mathscr{P}^*(s, \Gamma)$ such that $h^*(n_k^*) = r[(n_k^*, \ldots, n_K^*)]$, for all $k = 1, \ldots, K$.

For example, let $H(n) \subseteq \mathrm{R}^M$ contain the set of all nondominated elements in $\{r(P): P \in \mathscr{P}(n, \Gamma)\}$, and define $H = \times_{n \in N} H(n)$. Then it is straightforward to show that H is admissible.

We now present the main result of this section.

Proposition. *Assume H is admissible. Then* U* *is admissible.*

Proof. Let path $P = (n_1^*, \ldots, n_K^*) \in \mathscr{P}^*(s, \Gamma)$ be a path that satisfies the second condition in the definition of the admissibility of H. Note that $\mathrm{IE}(\gamma) \le u^{\mathrm{OPT}}$, for all $\gamma \in \Gamma$, where $u^{\mathrm{OPT}} = u[r(P)]$ for any $P \in \mathscr{P}^*(s, \Gamma)$. We claim that at any time before U* terminates, there exists a node $n_k^* \in P \cap$ OPEN such that $r[(n_1^*, \ldots, n_k^*)] \in G(n_k^*)$ and $\mathrm{IE}(n_k^*) \ge u^{\mathrm{OPT}}$ and that termination always identifies a path in $\mathscr{P}^*(s, \Gamma)$. Proof of this claim is based on an induction argument. Initially, s is expanded, n_2^* is placed in OPEN, and $r[(n_1^*, n_2^*)]$ becomes a member of $G(n_2^*)$. The culling process does not remove $r[(n_1^*, n_2^*)]$ from $G(n_2^*)$ since there exists an $h \in H$, namely h^*, such that

$$u[r(n_1^*, n_2^*) \circ h^*(n_2^*)] \ge u[g(s, n) \circ h^*(n)]$$

for all $n \in \mathrm{SCS}(s)$. Note that the LHS of the above inequality equals u^{OPT}; thus, $\mathrm{IE}(n_2^*) \ge u^{\mathrm{OPT}}$.

Assume there exists a node $n_{k-1}^* \in P \cap$ OPEN such that $r[(n_1^*, \ldots, n_{k-1}^*)] \in \mathrm{G}(n_{k-1}^*)$ and $\mathrm{IE}(n_{k-1}^*) \ge u^{\mathrm{OPT}}$. If n_{k-1}^* is not chosen for expansion and U* terminates, then since $\mathrm{IE}(n_{k-1}^*) \ge u^{\mathrm{OPT}}$, some path in $\mathscr{P}^*(s, \Gamma)$ other than P has been found, and the result holds. If n_{k-1}^* is chosen for expansion and is not a member of Γ, then n_k^* becomes or remains a member of OPEN and $r[(n_1^*, \ldots, n_k^*)]$ becomes or remains a member of $\mathrm{G}(n_k^*)$. The culling process will not remove $r[(n_1^*, \ldots, n_k^*)]$ from $G(n_k^*)$ since there is an $h \in H$, namely h^*, such that

$$u[r(n_1^*, \ldots, n_k^*) \circ h^*(n_k^*)] \ge u[g' \circ h^*(n)]$$

for any $g' \in G(n)$, for any $n \in$ OPEN. Again, the LHS of the above inequality equals u^{OPT}; therefore, $\mathrm{IE}(n_k^*) \ge u^{\mathrm{OPT}}$. If $n_{k-1}^* \in \Gamma$, i.e., if $K = k - 1$, then U* terminates having successfully found $P \in \mathscr{P}^*(s, \Gamma)$. Thus, termination is always successful, and hence U* is admissible. □

5. Example

We now illustrate application of U* by considering Example 1 in Carraway et al. (1990). This example is a two-criteria, best-path problem, where on each arc the first criterion represents the length of the arc and the second criterion represents the probability that the arc can be successfully traversed. All probabilities are assumed independent across the arcs. The objective is to find the nondominated paths from node 1 to node 6, where we wish to minimize distance and maximize the probability of successfully reaching node 6. Specifically, let

$N = \{1, \ldots, 6\}$,

$s = 1$,

$\Gamma = \{6\}$,

$A = \{(1, 2), (1, 3), (2, 3), (2, 4), (3, 4), (3, 5), (4, 5), (4, 6), (5, 6)\}$,

$r[(1, 2)] = (-3, 0.9)$,

$r[(1, 3)] = (-2, 0.75)$,

$r[(2, 3)] = (-1, 1.0)$,

$r[(2, 4)] = (-4, 0.75)$,

$r[(3, 4)] = (-4, 0.8)$,

$r[(3, 5)] = (-2, 0.6)$,

$r[(4, 5)] = (-2, 0.8)$,

$r[(4, 6)] = (-1, 0.8)$,

$r[(5, 6)] = (-1, 0.95)$,

$\circ = (+, \cdot)$,

$u[(r_1, r_2)] = r_1 + 20r_2$.

Therefore, if $r = (r_1, r_2)$ and $x = (x_1, x_2)$, then $u(r \circ x) = u[(r_1 + x_1, r_2 \cdot x_2)] = (r_1 + x_1) + 20(r_2 \cdot x_2)$. Observe that $u[r(1, 2, 3)] = 14 > 13 = u[r(1, 3)]$ but that $u[r(1, 2, 3, 5, 6)] = 2.60 < 3.55 = u[r(1, 3, 5, 6)]$, thus indicating that u violates the monotonicity assumption required for dynamic programming to find a most preferred path. In fact, Carraway et al. (1990) show that a conventional dynamic programming algorithm selects path (1, 2, 3, 5, 6) over (1, 3, 5, 6).

Furthermore, let $H = \times_{n \in N} H(n)$, where:

$H(2) = \{(-5, 0.6), (-6, 0.64), (-4, 0.57)\}$,

$H(3) = \{(-5, 0.64), (-3, 0.57)\}$,

$H(4) = \{(-1, 0.8)\}$,

$H(5) = \{(-1, 0.95)\}$,

$H(6) = \{e\}$.

The choice of $H(1)$ is of no consequence and can be made arbitrarily. We remark that $H(n)$, $n = 2, \ldots, 5$, represents the set of all nondominated reward vectors associated with paths from node n to Γ, which should be particularly effective information on which to base our search. Application of U* proceeds as follows.

Initialization
OPEN = $\{s\} = \{1\}$, $G(1) = (0, 1)$, LABEL$(n', n) = \varnothing$ $\forall (n, n') \in$ A.

Iteration 1.
Node 1 chosen for expansion.
OPEN = (2, 3).
LABEL(2, 1) = $\{(-3, 0.9)\}$, LABEL(3, 1) = $\{(-2, 0.75)\}$.
The culling step eliminates no reward vectors.
IE(2) = 3.20, IE(3) = 3.55.

Iteration 2.
Node 3 chosen for expansion.
OPEN = {2, 4, 5}.
LABEL(2, 1) = $\{(-3, 0.9)\}$, LABEL(4, 3) = $\{(-6, 0.6)\}$,
LABEL(5, 3) = $\{(-4, 0.45)\}$.
The culling step eliminates all reward vectors in $G(2)$ and $G(4)$. Therefore, the OPEN set is modified as follows:
OPEN = {5}.

Iteration 3.
Node 5 chosen for expansion.
OPEN = {6}.
LABEL(6, 5) = $\{(-5, 0.4275)\}$.

Iteration 4.
Node 6 chosen for expansion. Thus, U* terminates with the path(1, 3, 5, 6) having preference function value 3.55.

Consider U* now operating on the basis of the following less ambitious (admissible) heuristic set $H = \times_{n \in N} H(n)$:
$H(2) = \{r : r \le (-1, 1.0)\}$.
$H(3) = \{r : r \le (-4, 0.8)\} \cup \{r : r \le (-2, 0.6)\}$.
$H(4) = \{r : r \le (-1, 0.8)\}$.
$H(5) = \{r : r \le (-1, 0.95)\}$.
$H(6) = \{e\}$.

We note that $H(n)$, $n = 2, \ldots, 5$, has been induced by the reward vectors associated with arcs $(n, n') \in A$ for all $n' \in$ SCS(n). Application of U* proceeds as follows.

Initialization
OPEN = {1}.

Iteration 1.
Node 1 chosen for expansion.
OPEN = {2, 3}.
LABEL(2, 1) = $\{(-3, 0.9)\}$, LABEL(3, 1) = $\{(-2, 0.75)\}$.
The culling step eliminates no reward vectors.
IE(2) = 14, IE(3) = 6.

Iteration 2.
Node 2 chosen for expansion.
OPEN{3, 4}.
LABEL(3, 1) = {(−2, 0.75)},
LABEL(3, 2) = {(−4, 0.9)},
LABEL(4, 2) = {(−7, 0.675)}.
The culling step eliminates no reward vectors.
IE(3) = 6.4, IE(4) = 2.8.

Iteration 3.
Node 3 chosen for expansion.
OPEN = {4, 5}.
LABEL(4, 3) = {(−6, 0.6), (−8, 0.72)},
LABEL(4, 2) = {−7, 0.675)},
LABEL(5, 3) = {(−4, 0.45), (−6, 0.54)}.
The culling step eliminates no reward vectors.
IE(4) = 2.8, IE(5) = 3.55.

Iteration 4.
Node 5 chosen for expansion.
OPEN = {4, 6}.
LABEL(4, 3) = {(−6, 0.6), (−8, 0.72)},
LABEL(4, 2) = {(−7, 0.675)},
LABEL(6, 5) = {(−5, 0.4275), (−7, 0.513)}.
The culling step eliminates all reward vectors in $G(4)$. Therefore, the OPEN set is modified as follows:
OPEN = {6}.

Iteration 5.
Node 6 chosen for expansion. The final culling step eliminates the value (−7, 0.513) from LABEL(6, 5) and, therefore, from $G(6)$. U* then terminates with the most preferred path: (1, 3, 5, 6), having preference function value 3.55.

We remark that for the first heuristic set, only one solution path was fully considered, (1, 3, 5, 6), and only two other partial solution paths were considered, (1, 2) and (1, 3, 4). The second heuristic set required the evaluation of two solution paths, (1, 3, 5, 6) and (1, 2, 3, 5, 6), and three partial solution paths, (1, 3, 4), (1, 2, 4), and (1, 2, 3, 4). In contrast, exhaustive enumeration would require evaluation of all eight solution paths. Therefore, we have demonstrated the existence of problems for which U* outperforms exhaustive enumeration procedures, in terms of the number of partial solution paths examined.

A possibly more important point of comparison for the U* algorithm is provided by the Generalized Dynamic Programming (GDP) algorithm of Carraway et al. (1990) mentioned earlier. On this same example problem, without using the heuristic information we have assumed here, GDP required the examination of two solution paths and three partial solution paths, just as did U* for the second heuristic set. However, the GDP solution provided by Carraway et al. required the construction of bounds that could be reasonably compared to our provision of heuristic values. Thus, the computational comparison between U* and GDP remains in question based on this simple example.

We might note that the description of GDP provided in Carraway et al. (1990) does no provide sufficient detail as to the implementation of the solution procedure for the dynamic programming equations to enable us to compare operation counts or other complexity measures. We also note that, in general, the functional equations specified for GDP must be solved for all nodes in the graph, while our procedure could allow some nodes to be avoided entirely. In large graphs such node avoidance and potential subpath pruning could possibly result in substantial computational savings for U* compared to GDP. Finally, we note that much of the culling computation is unproductive, as can be seen from the small proportion of iterations in which reward vectors are actually eliminated with this procedure. A detailed comparison of this algorithm with various other approaches is a topic for further research.

6. Conclusions

In this paper, we considered the problem of determining a most preferred path from start node to goal node set in an acyclic OR-graph, given a multiattribute preference function, a multiobjective reward structure, and heuristic information about this reward structure. We presented an algorithm U* which is guaranteed to terminate with a most preferred path if given an admissible heuristic set. This result was applied to a simple example that fails to satisfy a sufficient condition for dynamic programming, as conventionally applied, to terminate with a most preferred path. A comparison of the performance of U* given different heuristics and the concept of consistency (see Pearl, 1984) will be explored in future research.

References

Carraway, R.L., Morin, T.L., and Moskowitz, H. (1990), "Generalized dynamic programming for multicriteria optimization", *European Journal of Operational Research* 44, 95–104.

Geoffrion, A.M. (1968), "Proper efficiency and the theory of vector optimization", *Journal of Mathematical Analysis and Applications* 22, 618–630.

Keeney, R.M., and Raiffa, H. (1976), *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, Wiley, New York.

Lark, J.W., and White, C.C. (1988), "A^G: A generalization of A^*", Department of Systems Engineering, University of Virginia, Charlottesville, VA.

Pearl, J. (1984), *Heuristics*, Addision-Wesley, Reading, MA.

Stewart, B.S. (1988), "Heuristic search with a general order relation", Ph.D. Dissertation, Department of Systems Engineering, University of Virginia, Charlottesville, VA.

Zionts, S., and Wallenius, J. (1977), "An interactive programming method for solving the multiple criteria problem", *Annals of Discrete Mathematics* 1, 551–562.