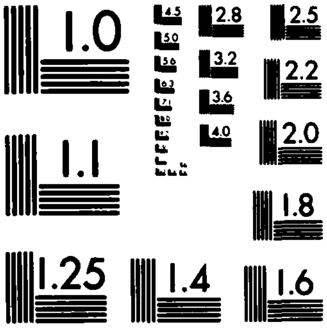
AD-A134 109 ON SOOE ISSUES CONCERNING OPTIMIZATION AND DECISION 1/1
TREES(U) ARIZONA STATE UNIV TEMPE GROUP FOR COOPUTER
STUDIES OF STRATE.. N V FINDLER ET AL. 1983 GCSS--R-14
UNCLASSIFIED AFOSR-TR-83-0780 AFOSR-82-0340 F/G 12/1 NL

END
FILMED


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

3

ASU Department of Computer Science

TEMPE, ARIZONA 85287

ON SOME ISSUES CONCERNING OPTIMIZATION AND DECISION TREES

Nicholas V. Findler, Michael S. Belofsky,
and Timothy W. Bickmore
Group for Computer Studies of Strategies
Computer Science Department
Arizona State University
Tempe, AZ 85287; USA

Group for Computer Studies of Strategies
Technical Report Number: 14

Departmental Technical Report Number: TR-83-003

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER
AFOSR-TR- 83-0780

2. GOVT ACCESSION NO.
AD A134109

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
"ON SOME ISSUES CONCERNING OPTIMIZATION AND DECISION TREES"

5. TYPE OF REPORT & PERIOD COVERED
INTERIM

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Nicholas V. Findler
Michael S. Belofsky
T.W. Bickmore

8. CONTRACT OR GRANT NUMBER(s)
AFOSR-82-0340

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Computer Science Department
Arizona State University
Tempe, AZ 85287

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
61102F 2304/A2

11. CONTROLLING OFFICE NAME AND ADDRESS
Air Force Office of Scientific Research/NM
Bolling› AFB, Washington, D.C. 20332

12. REPORT DATE
1983

13. NUMBER OF PAGES
15

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We describe the context and the constituent modules of a large-scale programing system, the Quasi-Optimizer. Its objectives are (a) to observe and measure adversaries' behavior in a competitive environment, to infer their strategies and to construct a computer model, a descriptive theory of each; (b) to identify strategy components, evaluate their effectiveness and to select the most satisfactory ones from a set of descriptive theories; (c) to combine these components in a quasi-optimum strategy that represents a normative theory in the statistical sense.

We also discuss certain properties of decision trees which are the primary representational structures of strategies in the computer. The verification of these properties, such as identity, equivalence and similarity between two decision subtrees, enable us to eliminate redudancies in the decision trees.

ON SOME ISSUES CONCERNING
OPTIMIZATION AND DECISION TREES

Nicholas V. Findler, Michael S. Belofsky,
and Timothy W. Bickmore
Group for Computer Studies of Strategies
Computer Science Department
Arizona State University
Tempe, AZ 85287; USA

ABSTRACT

The authors ~~We~~ describe the context and the constituent modules of a large-scale programming system, the Quasi-Optimizer. Its objectives are (a) to observe and measure adversaries' behavior in a competitive environment, to infer their strategies and to construct a computer model, a descriptive theory of each; (b) to identify strategy components, evaluate their effectiveness and to select the most satisfactory ones from a set of descriptive theories; (c) to combine these components in a quasi-optimum strategy that represents a normative theory in the statistical sense.

Also discussed are We also ~~discuss~~ certain properties of decision trees which are the primary representational structures of strategies in the computer. The verification of these properties, such as identity, equivalence and similarity between two decision subtrees, enable us to eliminate redundancies in the decision trees.

1. INTRODUCTION

First, we give a brief description of a long-term project, the Quasi-Optimizer (QO) system, in which decision trees (DTs) are used as the primary representational structure.

The QO has three major objectives (Findler and van Leeuwen, 1979; Findler, 1983):

(a) to observe and measure adversaries' behavior in a competitive environment, to infer their strategies and to construct a computer model, a descriptive theory, of each;

(b) to identify strategy components, evaluate their effectiveness and to select the most satisfactory ones from a set of descriptive theories;

(c) to combine these components in a quasi-optimum strategy that represents a normative theory in the statistical sense.

Let us define some terminology. A strategy is a decision-making mechanism that observes and evaluates its environment, and prescribes in response to it an action. This action, at the simplest level, does not change for the same environment over time, is a single and one-step response.

We have extended this concept in several directions. Learning strategies no longer are static. They improve the technique of evaluating the environment as well as the selection of the action, on the basis of experience. The single (that is, one-dimensional) action can be replaced by a set of (that is, multi-dimensional) actions. Instead of a one-step (momentary) action, we may have a sequence of actions that are unordered, weakly or strongly ordered over time. Finally, the decision variables defining the environment may also include descriptors that characterize relevant aspects of the history of the environment.

All these extensions make our studies more realistic, taking into account learning strategies, which can issue also multi-dimensional responses to complex environments. The actions may be the results of long-range planning processes and are based on both short-term and long-term considerations (tactical and strategic objectives, respectively).

As described later, we represent static strategies prescribing simple actions in terms of DTs. We note here only one important representational extension concerning learning strategies. We have developed a program that "freezes" the learning component of such a strategy and takes a "snapshot" of it in the form of a DT (Findler and Martins, 1981). Another module (Findler, Mazur and McCall, 1983) receives such a sequence of snapshots and, if it is statistically justified, computes the asymptotic form to which the sequence converges. We also note that the automatic generation of the computer model, the snapshot, can be done by the system either in being a passive observer or "under laboratory conditions," according to some experimental design. The experiments in the latter case are specified in one of three different ways:

(i) in an exhaustive manner when every level of a decision variable is combined with every level of the other decision variables;

(ii) by a binary chopping technique while relying on the assumption of a monotonically changing response surface;

(iii) according to a dynamically evolving design in which the levels selected for the decision variables, and the length of the whole experimentation, depend on the experimental results obtained up to that point (Findler, 1982; Findler and Cromp, 1983). This module minimizes the total number of experiments for a given level of precision.

2. ON COMPETITIVE ENVIRONMENTS AND THE QUASI-OPTIMIZER MODULES

Let us consider an environment in which several organizations compete to achieve some identical goal. (We may assume, for the sake of generality, that a goal vector is specified whose components need not be orthogonal in real life situations. In business management, for example, the relative share of the market and the volume of sales may be non-orthogonal goal dimensions.) Each organization perceives the environment by observing and measuring certain variables (numeric or symbolic) it considers relevant. Part of the strategy of the organizations aims at interpreting the measurements, determining a course of action leading to goal achievement and preventing the adversaries from achieving it. At any moment, the "rules" of competition, and the past and current actions of the competitors determine the next state of the environment.

The picture of the environment as perceived by an adversary is unclear because some information may be unavailable, missing (risky or uncertain -- according to whether or not the relevant a priori probability distributions are known, respectively) or obscured by noise. Noise may be caused by latent environmental factors or deliberate obfuscation by the competitors. There may also be conflicts and biases within an organization (e.g., rivalry between different divisions or personalities), which can perturb its measurements and distort its image of the environment. If a competitor's decisions based on such incomplete or faulty information are less sound than those of the others, resources will be wasted and goal attainment will be further removed.

If a new organization wants to enter such a confrontation, it must develop a strategy for itself. Assume that this strategy is to incorporate the best components of the extant adversaries' strategies. The process must start with a period of passive or active observation, i.e., before or after having entered the confrontation. In this phase, the new organization, therefore, has to construct first a model (a descriptive theory) of every other participant. To select the most satisfactory components of the (model) strategies, it would assign to each component some measure of quality, i.e., an outcome-dependent credit assignment must be made (Findler and McCall, 1983). (This assumes that the models are of uniform structure such as decision trees or production systems. Furthermore, credit must be assigned not on the basis of immediate outcome but often in relying on long-term considerations in view of planning strategies.)

Both short-term and long-term objectives can be discerned in the behavior of the adversaries. Short-term objectives comprise local and momentary goals, such as to mislead temporarily the others or to eliminate one of their resources, but short-term objectives naturally contribute to the long-term ones. The long-term objectives are achieved through the overall strategy which is an aggregate of tactics directed toward some short-term objective. A strategy is also more than that. It includes the means of evaluating the adversaries' situation and actions, scheduling of ones own tactics, and making use of feedback from the environment in modifying the rules of tactics both in terms of their contents and their inter-relations. In short, strategy gives tactics its mission and seeks to reap it results.

The strategy obtainable from the best components of the model strategies is a normative theory which is potentially the best of all available ones, on the basis of the information accessible by the new organization. This normative strategy is in fact only quasi-optimum for four reasons. First, the resulting strategy is optimum only against the original set of strategies considered. Another set may well employ controllers and indicators for decision-making that are superior to any of the "training" set. Second, the strategy is normative only in the statistical sense. Fluctuations in the adversary strategies, whether accidental or deliberate, impair the performance of the quasi-optimum strategy. Third, the adversary strategies may change over time and some aspects of their dynamic behavior may necessitate a change in the quasi-optimum strategy. Finally, the generation of both the descriptive theories (models) and of the normative theory (the quasi-optimum strategy) is based on approximate and fallible measurements.

This is the general context and the underlying motivation for the QQ system. The following is a brief description of the different modules it comprises:

(i) The QQ-1 assumes a monotonic strategy response surface and uses either exhaustive search or binary chopping to contruct a descriptive theory of static (non-learning) strategies. The program can make an inductive discovery in identifying correlations, if any, between the stochastic components of the strategy response and the subranges of the decision variables. The program can also be rendered a passive observer of the conflict situations -- in addition to operating under "laboratory conditions" under which it specifies the environment the strategy is to respond to. It can then experimentally discover the probability distribution of occurrence of the different regions of the domain of competition.

(ii) The QQ-2 extrapolates a finite sequence of decision trees, each representing the same learning strategy at different stages of development, and computes their asymptotic form. The latter is then used in constructing the normative theory.

(iii) The QQ-3 minimizes the total number of experiments QQ-1 has to perform. It no longer assumes that the strategy response surface is monotonic and also deals with multi-dimensional responses. QO-3 starts with a balanced incomplete block design for experiments and computes dynamically the specifications for each subsequent experiment. In other words, the levels of the decision variables in any single experiment and the length of the sequence of experiments depend on the responses obtained in previous experiments.

(iv) The QO-4 performs the credit assignment. That is, it identifies the components of a strategy and assigns to each a quality measure of the 'outcomes'. An outcome need not be only the immediate result of a sequence of actions prescribed by the strategy but can also invoke long-range consequences of planned actions. An important extension of this subproject enables a meta-strategy to channel the domain of confrontation to such regions in which a given strategy is most proficient.

(v) The QO-5 constructs a 'Super Strategy' by combining strategy components associated with outcomes of a quality above a threshold value.

(vi) The QO-6 generates a Quasi-Optimum strategy from the Super Strategy by eliminating inconsistencies and redundancies from the latter. It also tests and verifies the QO strategy for completeness.

3. ON DECISION TREES AND CERTAIN PROPERTIES OF THEIRS

A recent survey (Moret, 1982) has described in detail a particular type of DTs which are suitable for problems in switching theory, taxonomy and pattern recognition. Our investigations have used a different structure, as shown in the example of Fig. 1. (See last page.)

Each level of the DT is associated with one of the decision variables, $\underline{x}_1$, $\underline{x}_2$, ..., $\underline{x}_n$. The values of the latter may be numerically-oriented, rank numbers, symbolic (attributes, ordered or unordered categories) or structured data (hierarchies, relationships or priorities). The total range of each variable is mapped onto a normalized scale of (0, 128). The out-degree of every node equals the number of distinct subranges of the variable associated with the level at hand. The leaves attached to the branches at the last level, $\underline{a}_1$, $\underline{a}_2$, ..., $\underline{a}_m$, represent actions. Thus a particular combination of values of every decision variable characterizes the environment -- as perceived by the strategy the DT represents -- and defines a pathway from the root down to an action.

One can easily see that the representation of strategies by DTs is reasonably complete (with the extensions of the concept described earlier), including the uncertainties inherent in the identification of the environment and in the relation between given environments and given actions prescribed by the strategy.

Next, we discuss certain relations between two DTs or decision subtrees (DSTs): identity, equivalence and similarity. Algorithms to verify or disprove these properties are needed, for example, in the QO-6 module, mentioned before, that eliminates redundancies in DTs. There are four dimensions along which testing must be done:

(i) The ordered set of decision variables that appear in two DTs or DSTs;

(ii) The out-degrees of the corresponding nodes;

(iii) The boundary points of the corresponding subranges of decision variable values;

(iv) The corresponding actions prescribed by the strategy.

We call two DSTs _identical_ if the entities are the same with each corresponding member in the above four categories.

Two DSTs are _equivalent_ if there is a permutation on the sequence of decision variables of the _first_ DST that transforms it to another DST identical with the _second_ DST. (Actually, the permutation is performed in our program only if the DSTs are _likely_ to be equivalent -- as suggested by some inexpensive heuristic calculations.)

We note that one could argue that two DSTs are equivalent also in the case in which one or more functional mappings of certain decision variables of the first DST can transform their subranges to those of the decision variables at corresponding levels of the second DST. We contend, however, that any non-linear transformation changes the 'sensitivity' of the affected decision variables. In other words, the minimum discernible difference between adjacent values would change. This means that, in certain borderline cases, the strategy represented would no longer be the same.

Finally, we must provide a parametrizable metric to assess the degree of similarity between two DSTs. Let it suffice to say here that the user specifies for the program relative levels of dissimilarity tolerated in each of the four categories noted before. The aim is to reject the assumption of similarity, if such is the case, with as little calculation as possible. Therefore, the tests are carried out in an order of increasing complexity. Also, heuristic rules can be employed that recommend for execution the most likely test to fail.

4. FINAL COMMENTS AND CONCLUSIONS

We have described a large-scale programming system, the QQ, that has several theoretical and practical aspects of interest. We are in the process of integrating its different modules in order to use the whole system for several different applications.

We have also discussed certain properties of decision trees, the primary representational structures of competitive strategies in the computer.

5. ACKNOWLEDGMENTS

We are indebted to many past and current members of our Group for Computer Studies of Strategies, in particular to Neal Mazur, Robert Cromp, Bede McCall, George Sicherman, and Joan Martins. Our work has been supported by AFOSR Grants 81-0220 and 82-0340.

6. REFERENCES

Findler, N. V. and J. van Leeuwen (1979). The complexity of decision trees, the Quasi-Optimizer and the power of heuristic rules. *Information and Control, 40*, 1-19.

Findler, N. V. and J. P. Martins (1981). On automating computer model construction -- The second step towards the Quasi-Optimizer system. *J. of Information and Optimization Sciences, 2*, 119-136.

Findler, N. V. (1982). On a system that dynamically generates its own experimental design. In *Proc. of COMSTAT82, Fifth Internat. Symp. on Computational Statistics*, Toulouse, France. Physica Verlag: Würzburg, West Germany.

Findler, N. V. (1983). An overview of the Quasi-Optimizer System. Submitted for publication.

Findler, N. V., N. Mazur, and B. B. McCall (1983). A note on computing the asymptotic form of a limited sequence of decision trees. To appear in *Information Sciences*.

Findler, N. V. and R. F. Cromp (1983). An artificial intelligence technique to generate self-optimizing experimental designs. Submitted for publication.

Findler, N. V. and B. B. McCall (1983). A conceptual framework and a heuristic program for the credit assignment problem. Submitted for publication.

Moret, B. M. E. (1982). Decision trees and diagrams. *ACM Computing Surveys*, *14*, 593-623.

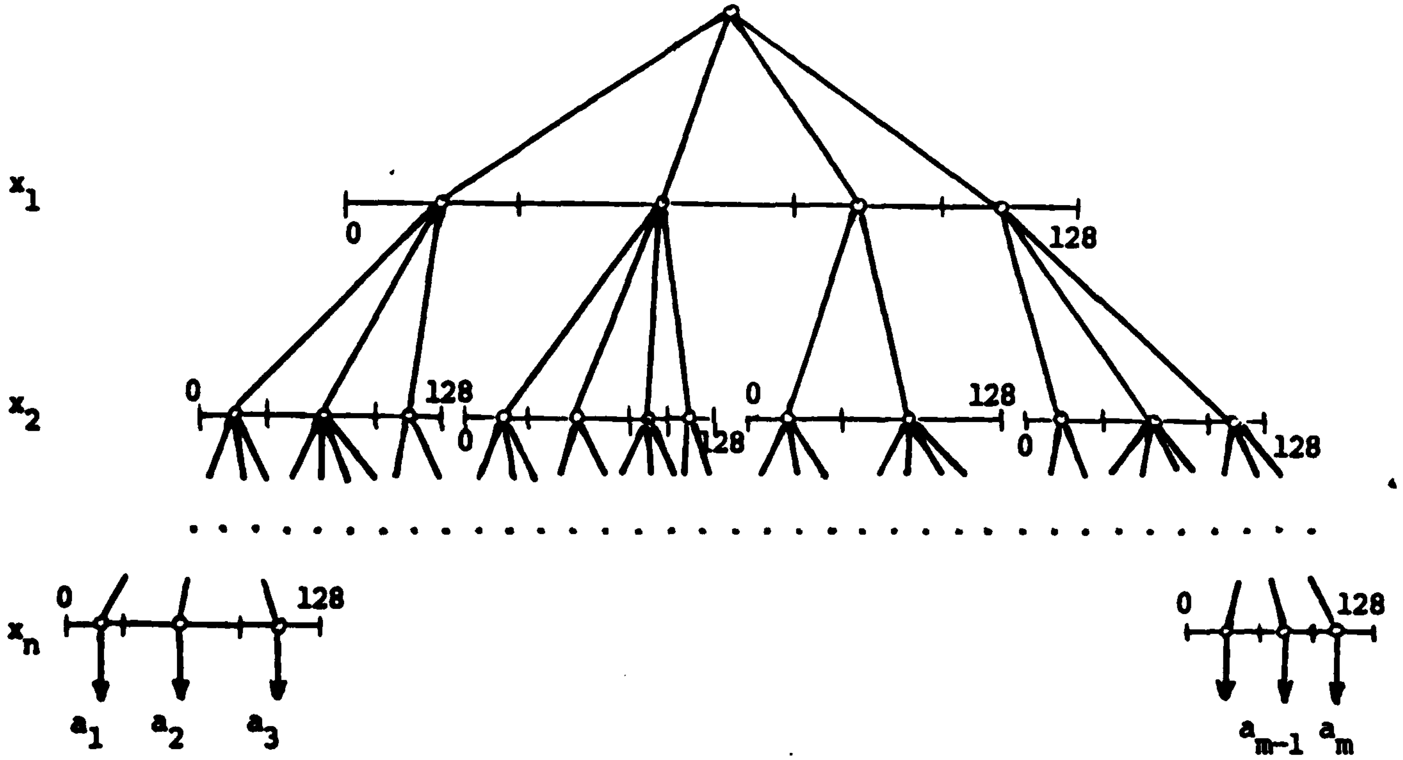


Figure 1

An Exemplary Decision Tree

Each level of the decision tree is associated with a decision variable, $\underline{x}_1$, $\underline{x}_2$, ..., $\underline{x}_n$. The total range of each is mapped onto a normalized scale (0, 128). The out-degree of every node equals the number of distinct subranges of the variable associated with the level at hand. The leaves attached to the branches at the last level, $\underline{a}_1$, $\underline{a}_2$, ..., $\underline{a}_m$, represent actions.

See the text concerning extending the scope of the representation for learning strategies, producing multi-dimensional, and sequence-of-action responses.