Program Abstracts/Algorithms

A program of a dynamic programming algorithm for finding pure admissible decision functions in statistical decisions

HERBERT MOSKOWITZ and RAYMOND JONES

*Krannert Graduate School of Industrial Administration
Purdue University, West Lafayette, Indiana 47907*

In certain decision situations it may be either difficult or objectionable on philosophical grounds to assess a decision maker's subjective probability distribution over the set of payoff relevant events. In such circumstances, the general decision problem may be examined using the normal form of analysis (see e.g., Raiffa, 1970). This method proceeds to find the set of ordinally admissible pure decision functions (strategies) by exhaustively evaluating all decision functions. Unfortunately, finding the ordinally admissible pure decision functions by the above procedure is computationally onerous and, in fact, becomes intractable for problems of even relatively small size [e.g., for a 10-act, 2-event, 9-message problem, with all acts common to each message, there exists 10^9 (= 10 billion) decision functions]. The program described below uses a dynamic programming algorithm developed by Moskowitz (1973), which finds the set of ordinally admissible pure decision functions through efficient recursion. The computational savings of the algorithm increases with problem size, and can be as much as 99% more efficient than exhaustive enumeration in terms of number of strategies evaluated.

Description. The program can analyze any type of decision tree problem. It begins with a basic input vector matrix of conditional utilities of partial decision functions (VMCUPDF), where a partial decision function is defined as an act taken in response to a specific message. The VMCUPDF is an Act by Message matrix, where each component of the vector of utilities associated with a given act-message pair (viz, partial strategy) corresponds to each payoff-relevant event.

The programmed algorithm operates on the principle that since different decision functions are composed of common partial strategies, only the vector of conditional utilities of the unique partial strategies need to be evaluated to ascertain dominance. In the algorithm, each message is regarded as a stage in the recursion. The program proceeds recursively by checking all partial decision functions for dominance, discarding those dominated and then adding the nondominated vectors to all vectors associated with another message, again

checking for dominance, etc. The algorithm thus proceeds sequentially on a message by message basis, eliminating those strategies, whose partial strategies are dominated at each stage. The final stage defines the set of ordinally admissible strategies. The order in which each message is considered does not affect the solution but can significantly affect core memory utilization and central processing unit (CPU) time.

Input. The input consists of three cards. Card 1 specifies: (1) the problem size in terms of number of actions, events, and messages, (2) whether the objective is to maximize or minimize the criterion function, (3) the sequence in which the messages are to be processed, i.e., random or systematic, (4) and the type and amount of output. Card 2 contains the utilities in the VMCUPDF, and Card 3 prespecifies the sequence in which messages are to be processed. Any number of combinations of message sequences can be processed in a given run. This permits observation of order effects on core memory and CPU time.

Output. The complete printed output includes: (1) a listing of parameter values, (2) problem input matrix (VMCUPDF), (3) sequence in which messages are processed, (4) the number of dominated and nondominated strategies identified at the end of each stage of analysis, and (5) specification of each nondominated strategy at each stage. Several options exist for eliminating or reducing the amount of output in (4) and (5) at each stage.

Computer and Language. The program is written in FORTRAN IV for the CDC 6500 computer. Conversion to other computer systems should be quite simple.

Restrictions. Limitations on problem size are primarily a function of core memory capacity and CPU time requirements. As an illustration, CPU time for a problem with 390×10^6 decision functions was about 35 min.

Availability. A detailed description of the algorithm and computer program, and complete program listing may be obtained free of charge by writing Herbert Moskowitz, Krannert Graduate School of Industrial Administration, Purdue University, West Lafayette, Indiana 47907.

REFERENCES

- Moskowitz, H. A dynamic programming algorithm for finding pure admissible decision functions in statistical decisions. Krannert Institute Paper No. 395, February 1973.
Raiffa, H. *Decision analysis, introductory lectures on choices under uncertainty*. Reading, Massachusetts: Addison-Wesley, 1970.