

ABSTRACT

Development of a Programming Language
for Multistage Optimization (August 1969)

James H. Nolen, B. A., Baylor University
M. S., Texas A&M University

Directed by: Dr. Wilbur L. Meier, Jr.

The primary objective of this research is to provide a user-oriented computer language for multistage optimization. While multistage optimization problems may assume many diversified forms in many areas of application, a basic underlying structure has been identified. Operating on a basis of this structure and the computational similarities of multistage optimization solutions, the language MOST (Multistage Optimization Syntax Translator) has been developed complete with formal description and user's guide. This report on the research is logically divided into two parts, Development of the Language and Utilization of the Language.

In the development of the language, a basic structure has been identified which unifies the varied forms of multistage optimization problems. Each problem formulation is comprised of certain basic parts and these parts consist of certain basic components. These parts, their components, and how they combine with the computational similarities is discussed in Chapter II.

A formal description of MOST is provided in Chapter III. The description includes the syntax of the language in Backus Normal Form. The semantics are presented in text form along with several

examples of each of the syntactical structures.

The MOST compiler has been developed in modular form with the modules linked closely to the basic parts of a problem formulation and the language syntax. The compiler is written in the PL/I language and produces PL/I as its target language. A justification for the use of PL/I as well as a discussion of the compiler modules and special procedures are presented in Chapter IV.

As an aid to the utilization of MOST, Chapter V provides a user-oriented guide. A step-by-step procedure is presented by which the user may formulate his problem, code it in the MOST language, submit it to the MOST compiler, and receive, as output from the system, the problem solution in a form which he has specified or chosen. Examples of the various language elements are included throughout the chapter.

The complete solution to two example problems are included in Chapter VI. The examples, one with analytical functions and the other with tabular entries, are discussed with relationship to the procedure of Chapter V. Included in the discussion is a brief description of the action of the compiler as related to the problems.