

Reference

NBS
Publi-
cations

NAT'L INST. OF STAND & TECH R.I.C.



A11105 047780

A11101 728821

NBSIR 79-1929

A Computerized Approach for Identifying Cost-Effective Fire Safety Retrofits in Health Care Facilities

Robert E. Chapman

Center for Building Technology

William G. Hall *11/22/79 SRS*

Center for Applied Mathematics

Phillip T. Chen *1/22/80 SRS*

Center for Building Technology
National Engineering Laboratory
National Bureau of Standards
U.S. Department of Commerce
Washington, D.C. 20234

December 1979

Final Report
Issued January 1980

Sponsored by the:
Center for Fire Research
National Engineering Laboratory
National Bureau of Standards
Washington, D.C. 20234

QC
100
.U56
79-1929
1980

SEP 19 1980

Be an - B

Q1100

1156

79-1959

1980

NBSIR 79-1929

**A COMPUTERIZED APPROACH FOR
IDENTIFYING COST-EFFECTIVE FIRE SAFETY
RETROFITS IN HEALTH CARE FACILITIES**

Robert E. Chapman

Center for Building Technology

William G. Hall

Center for Applied Mathematics

Phillip T. Chen

Center for Building Technology
National Engineering Laboratory
National Bureau of Standards
Washington, D.C. 20234

December 1979

Final Report
Issued January 1980

Sponsored by the:
Center for Fire Research
National Engineering Laboratory
National Bureau of Standards
Washington, D.C. 20234



U.S. DEPARTMENT OF COMMERCE, Philip M. Klutznick, *Secretary*

Luther H. Hodges, Jr., *Deputy Secretary*

Jordan J. Baruch, *Assistant Secretary for Science and Technology*

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

PREFACE

This research was conducted under the sponsorship of the Center for Fire Research by the Applied Economics Group, Center for Building Technology, and the Operations Research Division, Center for Applied Mathematics, National Engineering Laboratory, National Bureau of Standards.

This study develops a computerized procedure which helps identify the least-cost means of retrofitting a designated area of a health care facility to achieve compliance to the Life Safety Code. The procedure also identifies from 10 to 20 retrofit alternatives, some of which are quite close to the least-cost solution. The procedure is intended to be used as a management tool to facilitate the design selection process by providing both information on relative costs and a chance to match common retrofit packages across areas of the health care facility designated as deficient.

Appreciation is extended to Dr. Harold E. Marshall and Ms. Kimberly A. Barnes, Applied Economics Group, who reviewed the economic aspects of this paper. Appreciation is also extended to Ms. Patsy Saunders, Dr. Howard Hung and Mr. Ralph Schofer, Operations Research Division, and Messrs. Harold E. Nelson and A. J. Shibe, Center for Fire Research, who provided many useful suggestions for improving the treatment of certain topics in this paper. Special appreciation is extended to Dr. Alan J. Goldman, Operations Research Division, whose stimulating discussions provided guidance and encouragement throughout this effort.

ABSTRACT

This study focuses on how a computerized version of the Fire Safety Evaluation System developed by the Center for Fire Research at the National Bureau of Standards can be used to determine equivalence to the Life Safety Code in the least costly manner. This study presents a programmer-oriented discussion of the mathematical, economic and engineering considerations that went into the development of the linear programming algorithm for identifying cost-effective retrofits. Programmer-oriented topics treated in this report include: a discussion of user options; program documentation; format statements; flow charts; sample computer runs; and a complete listing of the computer program.

Keywords: Applied economics; building codes; economic analysis; fire safety; health care facilities; hospitals; life safety; mathematical programming; nursing homes; optimization; renovation.

CONTENTS

	<u>Page</u>
PREFACE	iii
ABSTRACT	iv
LIST OF FIGURES	vii
LIST OF TABLES	vii
LIST OF EXHIBITS	viii
SI CONVERSION UNITS	ix
1. INTRODUCTION	1
1.1 Background	1
1.2 Purpose	2
1.3 Scope and Approach	3
1.4 The Fire Safety Evaluation System	4
2. COMPUTERIZED PROCEDURE FOR IDENTIFYING COST-EFFECTIVE RETROFITS.	13
2.1 Method of Approach	14
2.2 Problem Formulation	15
2.3 Description of Computer Algorithm	19
2.4 Applying the Computerized Procedure	25
2.4.1 Data Requirements and User Options	26
2.4.2 The Output Format and Its Interpretation	38
3. SUMMARY	40
APPENDIX A Unit Pricing Information	42
APPENDIX B Documentation of Computer Algorithm/Program .	45
B.1 Program Functional Description	45
B.2 Variable Definition	49
B.3 Definitions of Data Inputs	53
B.4 Description of Reports Produced	55
APPENDIX C Sample Computer Runs	60
APPENDIX D Listing of the Computer Programs	83
D.1 Listing of MAIN Program	85
D.2 Listing of Subroutine PRSOL (Print SOLutions)	97

	<u>Page</u>
D.3 Listing of Subroutine FEASIB (FEASIBility Test)	98
D.4 Listing of Subroutine MOTEST (MODification TEST).	99
D.5 Listing of Subroutine RHTAB (Right Hand TABles).	99
D.6 Listing of Subroutine RVSM PX (ReVised SiMPleX).	100
D.7 Listing of Subroutine PRSMX (PRint SiMPleX)	112
REFERENCES	114

LIST OF FIGURES

	<u>Page</u>
1.1 Delphi Method	5
2.1 User Flowchart	14
2.2 Optimal Solution: Graphical Method	20
2.3 Alternative Solution: Graphical Method	23
D.1 Flowchart of Computer Program	84

LIST OF TABLES

	<u>Page</u>
1.1 Values of Safety Parameters Corresponding to Strict Compliance to the Life Safety Code for Non-Sprinklered Health Care Facilities	7
2.1 Abbreviations for Elements and Units to be Used in Reporting Data on Potential Retrofits	27
2.2 Listing of Data Requirements	28
2.3 User Options: Control and Modification Statements	34

LIST OF EXHIBITS

	<u>Page</u>
1.1 Occupancy Risk Factor Calculation Worksheet	8
1.2 Worksheet for Determining the Values of the Safety Parameters	9
1.3 Worksheet for Calculating Containment Safety, Extinguishment Safety, People Movement Safety, and General Safety	11
1.4 Worksheet for Evaluating Fire Zone Safety Equivalency	12
2.1 Initial Inputs	26
2.2 Sample Input Information on Potential Retrofits	30
2.3 Sample Control/Modification Card Sequence	35
2.4 Sample Output Format for Optimal and Prescriptive Solutions	39
C.1 Sample Input Run Stream	62
C.2 Sample Output Run Stream	63

SI CONVERSION UNITS

The conversion factors and units contained in this report are in accordance with the International System of Units (abbreviated SI for Systeme International d'Unites). The SI was defined and given official status by the 11th General Conference on Weights and Measures which met in Paris, France in October 1960. For assistance in converting U.S. customary units to SI units, see ASTM E 380, ASTM Standard Metric Practice Guide, available from the American Society for Testing and Materials, 1916 Race Street, Philadelphia, PA 19103. The conversion factors for the units found in this Standard are as follows:

Length

$$1 \text{ in} = 0.0254^* \text{ meter}$$

$$1 \text{ ft} = 0.3048^* \text{ meter}$$

$$1 \text{ mil} = 0.001^* \text{ in}$$

$$1 \text{ yd} = 0.9144^* \text{ meter}$$

Area

$$1 \text{ in}^2 = 6.4516^* \times 10^{-4} \text{ meter}^2$$

$$1 \text{ ft}^2 = 0.0929 \text{ meter}^2$$

$$1 \text{ yd}^2 = 0.836 \text{ meter}^2$$

Volume

$$1 \text{ in}^3 = 1.639 \times 10^{-5} \text{ meter}^3$$

$$1 \text{ liter} = 1.00^* \times 10^{-3} \text{ meter}^3$$

$$1 \text{ gallon} = 3.785 \text{ liters}$$

Temperature

$$^{\circ}\text{C} = 5/9 (\text{Temperature } ^{\circ}\text{F} - 32)$$

* Exactly

1. INTRODUCTION

1.1 BACKGROUND

The identification of cost-effective levels of fire safety in health care facilities is a major concern to hospital administrators, fire safety engineers and public policy makers. Rising construction and operating costs coupled with more stringent building codes and continuing advances in medical and building technology have complicated the issue, forcing health care facility administrators to carefully assess the alternative means through which they can design, construct or update their facilities. Although multiple death fires in health care facilities are rare occurrences, the potential for major losses of life and property does exist and should be recognized in the selection of fire safety measures. This potential and the resulting need for a national commitment to fire safety in health care facilities has been emphasized in numerous congressional hearings.¹

The National Fire Protection Association (NFPA) has long been recognized as a vehicle through which such a national commitment could be achieved. For over 60 years the NFPA has been a leader in the development of voluntary codes which establish acceptable fire safety levels. The Life Safety Code (NFPA 101-1973) is currently the most widely used guide for identifying the minimum level of fire safety in buildings. Although the code may be thought of as prescriptive since it prescribes fixed solutions for life safety in designated occupancies, performance concepts can be explicitly introduced through a provision which allows for equivalent solutions. In particular, Provision 1-3118 states:

Nothing in this code is intended to prevent the use of new methods or new devices, providing sufficient technical data is submitted to the authority having jurisdiction to demonstrate that the new method or device is equivalent in quality, strength, fire resistance, effectiveness, durability, and safety to that prescribed by this code.²

In light of this provision, the National Bureau of Standards' Center for Fire Research, through support from the Department of Health, Education and Welfare, has developed a system for determining how combinations of several widely accepted fire safety systems could be used to provide a

¹ Tragedy of Multiple Death Nursing Home Fires: The Need for a National Commitment to Safety, Select Committee on Aging, Subcommittee on Long-Term Care, September 1976.

² Code for Safety to Life from Fire in Buildings and Structures, National Fire Protection Association, NFPA 101-1973, p. 101-2.

level of safety equivalent to (equal to) that required in the 1973 Life Safety Code.¹ The equivalency methodology which emerged from this effort is particularly attractive since it lends itself to computer optimization techniques. Such an optimization technique should result in improved fire safety in health care facilities because it will resolve many of the differences of opinion surrounding the cost impacts of fire safety in health care facilities in general and the Life Safety Code in particular.

1.2 PURPOSE

The purpose of this report is to present a programmer-oriented description of the mathematical, economic, and engineering considerations that went into the development of a linear programming algorithm which permits the least cost means of achieving compliance to the Life Safety Code to be identified.

The computer program discussed in this report is particularly useful because it is based on the equivalency methodology developed by the Center for Fire Research. Since the NFPA is in the process of adopting this equivalency methodology into the Life Safety Code, any solutions from the computer program will satisfy Provision 1-3118 of NFPA 101-1973. Furthermore, since each of the parameters used in the equivalency methodology has a unique value which corresponds to strict compliance, it is possible to quantify the cost savings attributable to a performance based approach or equivalency methodology over that of strict compliance. Although the procedure is valid for both new and existing facilities, it is anticipated that its primary use will be in identifying alternative courses of action open to administrators faced with retrofitting existing facilities.

The computer program uses as its primary input information collected as an integral part of a thorough fire safety evaluation. This information permits the current state of the health care facility to be unambiguously identified. The least-cost or optimal combination of retrofits is identified using the following information:

- (1) the current state of the health care facility;
- (2) the minimum passing "score" needed to achieve compliance; and
- (3) the anticipated costs of each retrofit measure.

The computer program then generates and analyzes a class of alternative retrofit combinations. The optimal combination of retrofits and any alternatives which the program produces, usually between 10 and 20, are then summarized in tabular form and ranked from least costly to most costly. By using this approach, health care facility decision makers

¹ H. E. Nelson and A. J. Shibe, A System for Fire Safety Evaluation of Health Care Facilities, National Bureau of Standards, NBSIR 78-1555, November 1978.

have greater flexibility in choosing among retrofit combinations. In particular, by providing alternatives, the decision maker has the opportunity to assess very effectively the impact that nonconstruction costs would have on the choice of the optimal retrofit combination.

1.3 SCOPE AND APPROACH

The bulk of this report is composed of a set of technical appendices. These appendices are designed to be self-contained and hence may be read independently. The major thrust of each appendix is to lay out in detail the technical and theoretical considerations upon which the computerized version of the Fire Safety Evaluation System are based. In that sense the technical appendices are supporting evidence for results presented in the companion report, Economic Aspects of Fire Safety in Health Care Facilities: Guidelines for Cost-Effective Retrofits.¹

The final section of Chapter 1 outlines the Fire Safety Evaluation System and establishes a framework for the development of the linear programming package discussed in the remainder of the report.

The computer program and its method of solution are presented in Chapter 2. A mathematical discussion of the linear programming package used to identify the least-cost combination of retrofits is then given. The way in which the computer program generates alternative solutions using the optimal solution as a base is then presented. Explicit directions for using the computer program and interpreting its output are given in the final section of Chapter 2.

Appendix A presents the retrofit cost data used in the computerized procedure. The purpose of this appendix is to serve as a ready reference for potential users who may wish to adjust one or more of the retrofit costs to reflect local construction market conditions.

Appendix B gives a programmer-oriented description of the linear programming package. The appendix includes algorithmic and functional descriptions as well as user options and general requirements.

Appendix C consists of a series of sample computer runs. These runs are designed to highlight the input and output format and the user options which have been built into the procedure.

The report concludes with a listing of the complete computer software package. All programs presented in Appendix D are written in FORTRAN.

¹ Robert E. Chapman, Philip T. Chen, and William G. Hall, Economic Aspects of Fire Safety in Health Care Facilities: Guidelines for Cost-Effective Retrofits, National Bureau of Standards, NBSIR 79-1902, November 1979.

1.4 THE FIRE SAFETY EVALUATION SYSTEM

The Fire Safety Evaluation System is a quantitative evaluation system for grading fire safety in health care facilities. The primary use of the system is to determine how combinations of widely accepted fire safety equipment and building construction features may provide a level of safety equivalent to that required by the 1973 Life Safety Code. Three major concepts are basic to the Fire Safety Evaluation System. They are:¹

- 1) Occupancy Risk: the number of people affected by a given fire, the level of fire they are likely to encounter, and their ability to protect themselves.
- 2) Building Safety Features: the ability of the building and its fire protection systems to provide measures of safety commensurate with the risk.
- 3) Safety Redundancy: in-depth protection, through the simultaneous use of alternative safety methodologies such as Containment, Extinguishment, and People Movement. The design of the complete fire safety system is intended to ensure that the failure of a single protection device or method will not result in a major failure of the entire system.

The concept of safety redundancy is of central importance to the Fire Safety Evaluation System since Provision 2-1111 of the Life Safety Code requires:

The design of exits and other safeguards shall be such that reliance for safety to life in case of fire or other emergency will not depend solely on any single safeguard; additional safeguards shall be provided for life safety in case any single safeguard is ineffective due to human or mechanical failure.²

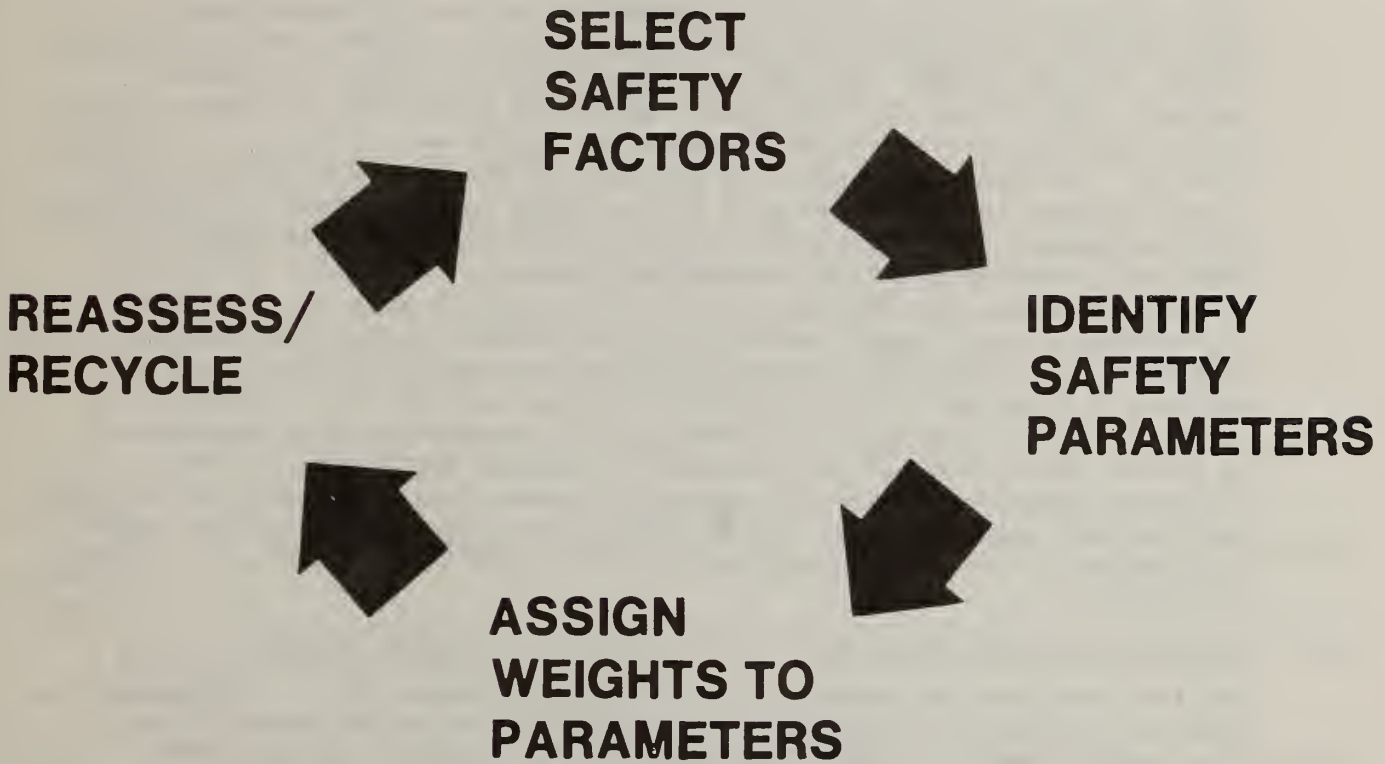
The task of ensuring that the Fire Safety Evaluation System satisfied Provision 2-1111 of the Life Safety Code was the responsibility of a panel of fire safety experts. The goal of the panel was to reach consensus on all relevant fire safety issues. To facilitate this process, a management tool known as the Delphi Method was used. The Delphi Method, as used in developing the Fire Safety Evaluation System, consisted of four steps. These steps are illustrated in Figure 1.1. Notice that the steps illustrated in the figure form a closed loop. This is because a certain amount of recycling of ideas was needed in order to achieve consensus.

¹ Definitions are taken from the report by H. E. Nelson and A. J. Shibe, A System for Fire Safety Evaluation of Health Care Facilities, National Bureau of Standards, NBSIR 78-1555, November 1978.

² Code for Safety to Life from Fire in Buildings and Structures, National Fire Protection Association, NFPA 101-1973, p. 101-4.

FIGURE 1.1 DELPHI METHOD

DELPHI METHOD



The first step in the process was to select a set of key safety factors. These factors were related to the concepts of Occupancy Risk and Building Safety Features mentioned earlier. (For example, Occupancy Risk includes as a factor Patient Mobility and Building Safety Features includes as a factor Interior Finish in the Corridor and Exits.) The second step focused upon the identification of a set of parameter associated with each factor. (For example, Patient Mobility includes as parameters Mobile, Limited Mobility, Not Mobile, Not Movable. Interior Finishes in the Corridor and Exits include Class A, Class B, and Class C flame spread ratings as parameters.) The third and most critical step was to assign a weight to each parameter which best reflects either the relative risk posed by that parameter or the net contribution to life safety provided by that parameter. More specifically, negative values reflected greater risks whereas positive values contributed toward a higher level of safety within the fire zone.¹ The system treats a value of zero as "safety" neutral. (The values for the three parameters associated with the safety factor Interior Finish in the Corridor and Exits are: Class C, -5 points; Class B, 0 points; and Class A, 3 points.) In the fourth step, the value of each safety parameter assigned by the panel is reassessed for adequacy and consistency. In the event that the system which emerged from the panel was shown to be inadequate or inconsistent, the entire four-stage process was repeated. The end result of the panel's work was a series of worksheets which permitted the relative merits of each fire safety parameter to be carefully assessed. These worksheets are presented as Exhibits 1.1 through 1.4. In addition, since each of the 13 Building Safety Features (see Table 1.1) has a unique parameter which corresponds to strict compliance, it was possible to compute the score, or level of safety, provided by the Life Safety Code for Extinguishment, Containment and People Movement Safety. These values were then used as a base which any alternative to strict compliance to the Life Safety Code had to match or exceed.

The first worksheet, Exhibit 1.1, consists of a brief description of the fire zone and the means for calculating Occupancy Risk. Occupancy Risk is calculated by selecting the appropriate value for each of the five factors shown in Table 1 of Exhibit 1.1. These values are then entered into the coded spaces in Table 2 (Exhibit 1.1) and multiplied together to get an unadjusted occupancy risk factor. The resultant is then entered either in Table 3A (Exhibit 1.1) if the building is new or in Table 3B if the building is existing. The occupancy risk factor for the fire zone is then calculated by taking the product of the weighting factor and the unadjusted occupancy risk factor. The Occupancy Risk factor is then used to establish the minimum level of General Safety which must be provided by the 13 Building Safety Features in order to be deemed in compliance to the Life Safety Code.

¹ A fire zone is defined as a space separated from all other spaces by floors, horizontal exits, or smoke barriers.

Table 1.1 VALUES OF SAFETY PARAMETERS CORRESPONDING TO STRICT COMPLIANCE TO THE LIFE SAFETY CODE FOR NON-SPRINKLERED HEALTH CARE FACILITIES

SAFETY FACTOR	BUILDING TYPE			
	NEW 1 STORY	NEW-MORE THAN 1 STORY	EXISTING 1 STORY	EXISTING MORE THAN 1 STORY
	PROT. NONCOMB. ^a	FIRE RESISTIVE	PROT. NONCOMB.	FIRE RESISTIVE
1. CONSTRUCTION	2	4	2	4
2. INTERIOR FINISH [Corr. & Exit]	CLASS A	CLASS A	CLASS B	CLASS B
	3	3	0	0
3. INTERIOR FINISH [Rooms]	CLASS B	CLASS B	CLASS B	CLASS B
	1	1	1	1
4. CORRIDOR PARTITIONS/WALLS	≥ 1.0 HR	≥ 1.0 HR	$< 1/3$ HR	$< 1/3$ HR
	2	2	0	0
5. DOORS TO CORRIDOR	> 20 Min	≥ 20 Min.	≥ 20 Min.	≥ 20 Min.
	1	1	1	1
6. ZONE DIMENSIONS ^b	100'-150'	100'-150'	100'-150'	100'-150'
	0	0	0	0
7. VERTICAL OPENINGS	Non-App.	2 HR	Non-App.	≥ 1 - < 2 HR
	0	3	0	2
8. HAZARDOUS AREAS	No Deficiencies	No Deficiencies	No Deficiencies	No Deficiencies
	0	0	0	0
9. SMOKE CONTROL	Smoke Partition	Smoke Partition	Smoke Partition	Smoke Partition
	0	0	0	0
10. EMERGENCY MOVEMENT ROUTES	Multiple Routes	Multiple Routes	Multiple Routes	Multiple Routes
	0	0	0	0
11. MANUAL FIRE ALARM	W/FD Conn.	W/FD Conn.	W/O/FD Conn.	W/O/FD Conn.
	2	2	1	1
12. SMOKE DETECTION & ALARM	Corridor Only	Corridor Only	None	None
	2	2	0	0
13. AUTOMATIC SPRINKLERS	None	None	None	None
	0	0	0	0
TOTAL VALUE	13	18	5	9

^a Protected Noncombustible.

^b No dead ends greater than 30 feet and corridor length is as recorded.

Table 1. OCCUPANCY RISK PARAMETER FACTORS

RISK PARAMETERS	RISK FACTOR VALUES					
1. PATIENT MOBILITY (M)	MOBILITY STATUS	MOBILE	LIMITED MOBILITY	NOT MOBILE	NOT MOVABLE	
	RISK FACTOR	1.0	1.6	3.2	4.5	
2. PATIENT DENSITY (D)	PATIENT	1-5	6-10	11-30	>30	
	RISK FACTOR	1.0	1.2	1.5	2.0	
3. ZONE LOCATION (L)	FLOOR	1ST	2ND OR 3RD	4TH TO 6TH	7TH AND ABOVE	BASEMENTS
	RISK FACTOR	1.1	1.2	1.4	1.6	1.6
4. RATIO OF PATIENTS TO ATTENDANTS (T)	PATIENTS ATTENDANT	<u>1-2</u>	<u>3-5</u>	<u>6-10</u>	<u>>11</u>	ONE OR MORE NONE
	RISK FACTOR	1.0	1.1	1.2	1.5	4.0
5. PATIENT AVERAGE AGE (A)	AGE	UNDER 65 YEARS AND OVER 1 YEAR		65 YEARS & OVER 1 YEAR & YOUNGER		
	RISK FACTOR	1.0		1.2		

* RISK FACTOR OF 4.0 IS CHARGED TO ANY ZONE THAT HOUSES PATIENTS WITHOUT ANY STAFF IN IMMEDIATE ATTENDANCE

Table 2. OCCUPANCY RISK FACTOR CALCULATION

	M	D	L	T	A	F
OCCUPANCY RISK	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Table 3A. (NEW BUILDINGS)

	F	R
	<input type="text"/>	<input type="text"/>

1.0^x

Table 3B. (EXISTING BUILDINGS)

	F	R
	<input type="text"/>	<input type="text"/>

0.5^x

Table 4. SAFETY PARAMETERS VALUES

PARAMETERS	PARAMETERS VALUES						
	COMBUSTIBLE				NON-COMBUSTIBLE		
1. CONSTRUCTION	WOOD FRAME		ORDINARY		UNPROTECTED	PROTECTED	FIRE RESIST.
	UNPROTECTED	PROTECTED	UNPROTECTED	PROTECTED			
FLOOR OF ZONE	UNPROTECTED	PROTECTED	UNPROTECTED	PROTECTED	UNPROTECTED	PROTECTED	FIRE RESIST.
FIRST	-2	0	-2	0	0	2	2
SECOND	-7	-2	-4	-2	-2	2	4
THIRD	-9	-2	-4	-7	-2	2	4
4TH & ABOVE	-13	-7	-13	-7	-9	-7	4
2. INTERIOR FINISH (Corr. & Exit)	CLASS C	CLASS B	CLASS A				
	-5	0	3				
3. INTERIOR FINISH (Rooms)	CLASS C	CLASS B	CLASS A				
	-3	3	3				
4. CORRIDOR PARTITIONS/WALLS	NONE OR INCOMPLETE	<1/3 HR	>1/3 <1.0 HR	≥1.0 HR			
	-10 (0)*	0	1 (0)*	2 (0)*			
5. DOORS TO CORRIDOR	NO DOOR	<20 MIN. FR	>20 MIN. FR	>20 MIN. FR & AUTO CLOS.			
	-10	0	1 (0)***	2 (0)***			
6. ZONE DIMENSIONS	DEAD END MORE THAN 100'	DEAD END 30'-100'	NO DEAD ENDS >30' & ZONE LENGTH IS:				
			>150'	100'-150'	<100'		
	-6 (0)**	-4 (0)**	-2	0	1		
7. VERTICAL OPENINGS	OPEN 4 OR MORE FLOORS	OPEN 2 OR 3 FLOORS	ENCLOSED WITH INDICATED FIRE RESIST.				
			<1 HR.	≥1HR <2 HR.	≥2 HR.		
	-14	-10	0	2 (0)*	3 (0)*		
8. HAZARDOUS AREAS	DOUBLE DEFICIENCY		SINGLE DEFICIENCY		NO DEFICIENCIES		
	IN ZONE	OUTSIDE ZONE	IN ZONE	IN ADJACENT ZONE			
	-11	-5	-6	-2			0
9. SMOKE CONTROL	NO CONTROL	SMOKE PARTITION	MECH. ASSISTED SYSTEMS				
			BY ZONE	BY CORRIDOR			
	-2 (0)***	0	0	0			
10. EMERGENCY MOVEMENT ROUTES	<2 ROUTES	MULTIPLE ROUTES					
		DEFICIENT CAPACITY	W/O HORIZONTAL EXIT(S)	HORIZONTAL EXIT(S)	DIRECT EXIT(S)		
	-8	-2	0	3	5		
11. MANUAL FIRE ALARM	NO MANUAL FIRE ALARM		MANUAL FIRE ALARM				
			W/O F.D. CONN.	W/F.D. CONN.			
	-4		1	2			
12. SMOKE DETECTION & ALARM	NONE	CORRIDOR ONLY	ROOMS ONLY	CORRIDOR & HABIT SPACE	TOTAL SPACE		
	0	2	3	4	5		
13. AUTOMATIC SPRINKLERS	NONE	CORRIDOR	CORRIDOR & HABIT SPACE	TOTAL SPACE			
	0	2 (0)**	8	10			

NOTE: * Use (0) when item 5 is -10.
 ** Use (0) when item 10 is -8.
 *** Use (0) in zone with less than 31 patients in existing buildings.

* Use (0) when item 1 is based on first floor zone or on an unprotected type of construction.
 ** Use (0) when item 1 is based on an unprotected type of construction.
 *** Use (0) when item 4 is -10.

The second worksheet, Exhibit 1.2, provides the foundation for the computerized procedure discussed in Chapter 2 and Appendices B, C, and D. Table 4, Exhibit 1.2, gives the scores associated with the parameters for each of the 13 Building Safety Features. To evaluate each fire zone, it is necessary to identify the appropriate value associated with each of the 13 Building Safety Features. The existing state of the fire zone is then defined by recording (circling, marking, or checking) all of these values on Table 4 (Exhibit 1.2). In the evaluation, it is important to point out that the level of each Building Safety Feature is determined by the worst case condition within the fire zone. Referring once again to Exhibit 1.2 it can be seen, for each Building Safety Feature, that those parameters which have a higher score represent potential retrofits. For example, if in the current state the flame spread rating on Interior Finishes in the Corridor and Exits was Class C, then both Class B and Class A flame spread ratings would be potential retrofits. More importantly, by combining the improvement in score with the anticipated retrofit costs, it is possible to establish a means for upgrading the level of fire safety within the fire zone in the most cost-effective manner.

Table 5, the third worksheet, is shown in Exhibit 1.3. Table 5 Exhibit 1.3 Provides a means for calculating the score associated with the four safety redundancy requirements. (The four safety redundancy requirements are: 1) Containment Safety; 2) Extinguishment Safety; 3) People Movement Safety; and 4) General Safety.)

In order to calculate the score for each of the safety redundancy requirements, it is necessary to enter the value of the parameter identified in Table 4 (Exhibit 1.2) as corresponding to the existing state of the Building Safety Feature into the appropriate spaces in the coded rows of Table 5 (Exhibit 1.3). (No values are entered in the shaded spaces of Table 5, Exhibit 1.3.) Each of the four columns is then summed to get an overall score. These scores are labeled S_1 , S_2 , S_3 and S_G in Table 5 of Exhibit 1.3.

The fourth worksheet provides the means for determining whether or not the fire zone generates a level of fire safety equivalent to that of the Life Safety Code. Basically, this is done by taking the four scores calculated in Table 5 (see Exhibit 1.3) and entering them in the boxes labeled S_1 , S_2 , S_3 and S_G . The user then selects the values from Table 6 (see Exhibit 1.4) for Containment Safety, Extinguishment Safety, and People Movement Safety for the appropriate building type and fire zone location. These values are entered in the boxes labeled S_a , S_b , S_c in Table 7 of Exhibit 1.4. The Occupancy Risk factor calculated on the first worksheet is then entered in the box labeled R. Based on these two sets of numbers, it is possible to test if the fire zone provides a level of safety equivalent to the Life Safety Code. This test is performed by determining if the difference between the first set of numbers, S_1 , S_2 , S_3 , and S_G , and the second set of numbers, S_a , S_b , S_c , and R, in Table 7 (see Exhibit 1.4) is greater than or equal to zero.

Table 5. **INDIVIDUAL SAFETY EVALUATIONS**

SAFETY PARAMETERS	CONTAINMENT SAFETY (S ₁)	EXTINGUISHMENT SAFETY (S ₂)	PEOPLE MOVEMENT SAFETY (S ₃)	GENERAL SAFETY (S _G)
1. CONSTRUCTION				
2. INTERIOR FINISH (Corr. & Exit)				
3. INTERIOR FINISH (Rooms)				
4. CORRIDOR PARTITIONS/WALLS				
5. DOORS TO CORRIDOR				
6. ZONE DIMENSIONS				
7. VERTICAL OPENINGS				
8. HAZARDOUS AREAS				
9. SMOKE CONTROL				
10. EMERGENCY MOVEMENT ROUTES				
11. MANUAL FIRE ALARM				
12. SMOKE DETECTION & ALARM				
13. AUTOMATIC SPRINKLERS			÷ 2 =	
TOTAL VALUE	S ₁ =	S ₂ =	S ₃ =	S _G =

Table 6. MANDATORY SAFETY REQUIREMENTS						
ZONE LOCATION	CONTAINMENT S _a		EXTINGUISHMENT S _b		PEOPLE MOVEMENT S _c	
	New	Exist.	New	Exist.	New	Exist.
FIRST FLOOR	9.0	4.0	6.0	3.0	6.0	1.0
ABOVE FIRST FLOOR	14.0	8.0	8.0	5.0	9.0	3.0

Table 7. ZONE SAFETY EQUIVALENCY EVALUATION						YES	NO
CONTAINMENT SAFETY (S ₁)	less	MANDATORY CONTAINMENT (S _a)	≥ 0	S ₁ - S _a = C	<input type="text"/>	<input type="text"/>	<input type="text"/>
EXTINGUISHMENT SAFETY (S ₂)	less	MANDATORY EXTINGUISHMENT (S _b)	≥ 0	S ₂ - S _b = E	<input type="text"/>	<input type="text"/>	<input type="text"/>
PEOPLE MOVEMENT SAFETY (S ₃)	less	MANDATORY PEOPLE MOVEMENT (S _c)	≥ 0	S ₃ - S _c = P	<input type="text"/>	<input type="text"/>	<input type="text"/>
GENERAL SAFETY (S _G)	less	OCCUPANCY RISK (R)	≥ 0	S _G - R = G	<input type="text"/>	<input type="text"/>	<input type="text"/>

In the event that the fire zone fails to pass the equivalency test, it will be necessary to select a retrofit strategy which will ensure that the Building Safety factors produce scores which match or exceed each of the four redundancy requirements. A systematic means for doing this which explicitly introduces relative costs into the retrofit decision is the subject of the remainder of this report.

2. COMPUTERIZED PROCEDURE FOR IDENTIFYING COST-EFFECTIVE RETROFITS

The computerized procedure presented in this section is based on a mathematical technique known as linear programming. In its usual context linear programming deals with the problem of allocating limited resources among competing activities in an optimal way. At the foundation of any linear programming problem is a mathematical model which describes the problem of concern. In this case the mathematical model is the Fire Safety Evaluation System. The term "linear" refers to the requirement that all mathematical functions in the model are linear.¹ The term "program" is used in the general sense in that it refers to a plan rather than a computer program per se. The basic reason why all mathematical functions involved in the problem are linear may be explained through reference to Table 4 and Table 5 of the Fire Zone Safety Evaluation Worksheet (see Exhibits 1.2 and 1.3). In Table 4 there is one and only one level of each Building Safety Feature possible at any one time.² This is due to the requirement that the most hazardous level associated with each feature determines the score for that feature. In Table 5 (Exhibit 1.3) the score for each of the four safety redundancy requirements (containment safety, extinguishment safety, people movement safety, and general safety) is the sum of the values of the appropriate parameter identified in Table 4 (Exhibit 1.2) as either the existing state or a potential retrofit.

¹ A linear function is defined as a function of the form

$$f(X) = a_0 + a_1x_1 + \dots + a_jx_j + \dots + a_nx_n$$

where a_j are coefficients not all zero and the x_j are variables. The geometrical representation of a linear function is a straight line, a plane, or a hyperplane. For example, $f(x) = a + bx$, a straight line, is a linear function where $g(x) = c + dx^2$, a parabola, is not.

² The above condition requires that all variables corresponding to factor levels must be either zero or one. The solution to a linear programming problem yields continuous variables; however, it can be shown that, because of the constraint structure of the model, the continuous solution is always "nearly" integer.

2.1 METHOD OF APPROACH

The method of approach taken in developing and using the linear programming procedure has the Fire Safety Evaluation System as its foundation. The steps which must be taken in order to go from the basics of the Fire Safety Evaluation System to the linear programming procedure are the subject of this section. To illustrate these steps a schematic diagram known as a flowchart is useful.

FIGURE 2.1 USER FLOWCHART

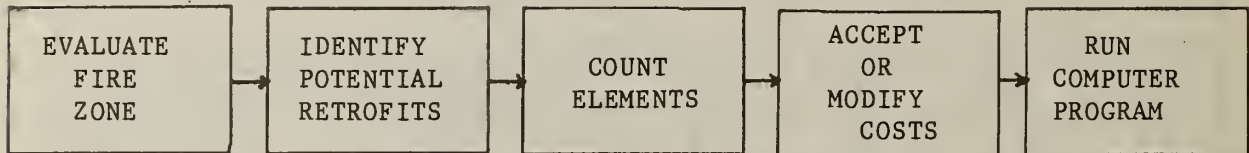


Figure 2.1 displays the sequence of steps between the Fire Safety Evaluation System and the computerized procedure. As a first step the Fire Zone Safety Evaluation Worksheet must be used to identify the existing state of each of the 13 Building Safety Features. Plugging this information into Table 5 of the worksheet (see Exhibit 1.3) permits the existing state of each feature and each individual fire zone to be determined. Using the information on the existing state for each feature as a starting point, engineering judgment may then be used to identify a set of potential retrofits.¹ It is important to point out that based on engineering judgment some "theoretically" possible retrofits may be excluded.² One important opportunity which should not be overlooked is the collection of information on potential retrofits during the safety inspection. As indicated in the third block, once a set of potential retrofits has been identified it is then necessary to count the number of elements which must be treated in order to move to a higher state. The fourth step in the sequence of events relates to the unit cost or cost per element. These costs are the ones actually used in performing the economic analyses. They are an integral part of the procedure and are stored in an "element cost matrix." This cost matrix interacts with the information on the number of elements in the following way. Associated with each potential retrofit is a set of information on the one or more elements which must be treated to move to a higher state. This "format" is followed when information

¹ Engineering input is important at all stages of the procedure but is particularly important in identifying potential retrofits.

² This opportunity represents the first of three user options which will be addressed in Section 2.4. (A theoretically possible retrofit is any parameter which has a higher score than the parameter corresponding to the existing state.) By the same token the user has the option to require that a particular retrofit be undertaken, or not undertaken.

on the number of elements is input. All of this information is then stored in an "element count matrix." The product of the element count and element cost matrices yields the total cost associated with each potential retrofit. It is important to point out that the user has the option to accept or modify any information contained in the element cost matrix. The considerations underlying the per unit or per element costs are presented in Appendix A. Thus, should the user feel that the information in the element cost matrix does not adequately reflect the costs associated with the local construction market or is inappropriate on some other grounds, it can be modified either through the use of time-location factors or changing one or more of the engineering assumptions.

The information identified in the sequence of steps outlined above is the only information required to run the computerized procedure. Although the computerized procedure is only a "working prototype" a maximum amount of flexibility has been built into it. These efforts at simplifying the procedure are the subject of Sections 2.4 and 2.5. The subject of the next two sections, however, is the formulation of the linear programming problem and a discussion of how the computer algorithm selects the optimal solution and generates a set of alternatives. Certain portions of the discussion which follows are more technical than previous sections. Therefore the reader who is primarily interested in obtaining a conceptual understanding of linear programming and how it may be applied to the problem of retrofitting a fire zone is directed to Section 2.3.

2.2 PROBLEM FORMULATION

To formulate the mathematical model (linear program) for this problem, we shall refer to information in Table 4 of the Fire Zone Safety Evaluation Worksheet. Table 4 may be thought of as a matrix with 13 rows,¹ one

¹ Building Safety Feature 1, Construction, is a special case. In Table 4 of the Worksheet there are four separate rows of scores corresponding to different floors of the building. Since a retrofit involving a change of floor need not be considered, only one of these rows (that for the appropriate floor) is included in the computer stored version of Table 4. Each row is divided into three segments: (1) Combustible-Wood Frame; (2) Combustible-Ordinary; and (3) Non-combustible. The first two segments contain two cells: (1) Unprotected; and (2) Protected. The third segment contains three cells: (1) Unprotected; (2) Protected; and (3) Fire Resistive. The procedure assumes that it is impossible to move from one segment to another. This assumption was made because of the difficulty in identifying cost elements and standard procedures for such retrofits. For computation purposes a dummy element is added to segments one and two. This permits an auxiliary 4 by 3 by 3 array to be used to store all values of the first Building Safety Feature and simplifies the construction of the computer stored version of Table 4. This approach may be changed when actual production runs begin

for each of the 13 Building Safety Features and 6 columns.¹ There are up to 6 parameters (states) associated with any given safety feature. The variable X_{ij} is used to identify the state of each factor. By definition, X_{ij} is equal to one if the i^{th} factor is in the j^{th} state and is zero otherwise. (Recall that the score for each factor is determined by the most hazardous condition within the fire zone.) V_{ij} is the score associated with the i^{th} feature and j^{th} state (entry in Table 4). These values are taken directly from Table 4. For example, if the i^{th} factor is in the j^{th} state the value, or score, associated with the i^{th} row is $V_{ij}X_{ij}$ since X_{ij} is equal to zero for j' unequal to j . Score improvements are thus possible as a result of retrofits. If the i^{th} factor is currently in the j_c^{th} state, X_{ij_c} would be the preretrofit or current state, and X_{ij} would be the postretrofit or new state. It is of course required that $j > j_c$ otherwise no improvement has been made. An examination of the scores associated with X_{ij} and X_{ij_c} shows that $V_{ij} > V_{ij_c}$.² It is now time to show how costs may be introduced into the mathematical model. The variable C_{ij} is used to record the cost of upgrading the i^{th} Building Safety Feature from the j_c^{th} state to the j^{th} state. For each possible retrofit being considered, C_{ij} is a function of the number of elements and the unit cost or cost per element. In mathematical terms C_{ij} is the dot product of the i^{th} vector of element quantities and the i^{th} vector of element costs.

Keeping in mind that the objective is to match or exceed the scores for each of the four categories of safety (containment, extinguishment, people movement, and general) in the least costly manner, it is now possible to formulate the problem mathematically. The problem is to choose X_{ij} , $j_c^1 \leq j \leq j_{\text{max}}^1$, where j_c^1 is the current state of the i^{th} factor and j_{max}^1 is the maximal state of the i^{th} factor, so as to minimize total retrofit costs

$$\sum_{i=1}^{13} \sum_{j=j_c^i}^{j_{\text{max}}^i} C_{ij} X_{ij} \quad X_{ij} = 0 \text{ or } 1$$

¹ There are no more than five real states for any of the safety factors, but the conditions *, **, and ... footnoted in Table 4 are most easily accommodated by the introduction of contrived variables and constraints. Therefore the extra column is required. The contrivance is not observable by the user; pre-optimization and post-optimization processing transform the variables from and to those defined by Table 4.

² Several of the V_{ij} 's have alternative values due to the level of some other Building Safety Feature (other i). Whenever such a situation occurs it is called a row interdependence. Three distinct row interdependencies are possible. These interdependencies, which involve rows 4, 5 and 6, are defined in the footnotes at the bottom of Table 4. These interdependencies may result in up to 3 additional constraints, called interdependency constraints, and up to 3 additional variables, called dummy states.

subject to

- (1) parameter value constraints for the cases where row interdependencies are not possible are given as:

$$\sum_{j=j_c^i}^{j_{\max}^i} X_{ij} = 1 \quad \text{for } i \neq 4, 5, 6$$

parameter value constraints for the cases where a row interdependence is possible are given as:

$$\sum_{j=j_c^i}^{j_{\max}^i+1} X_{ij} = 1 \quad \text{for } i = 4, 5, 6$$

where $j_{\max}^i + 1$ denotes a dummy state

- (2) parameter value interdependency constraints¹

$$X_{45} - X_{51} = 0$$

$$X_{41} - X_{53} - X_{54} - X_{55} = 0$$

$$X_{65} + X_{66} - X_{10,1} = 0$$

- (3) total value requirements

$$\sum_{i \in S_1} \sum_{j=j_c^i}^{j_{\max}^i} V_{ij} X_{ij} - Y_1 = S_a$$

$$\sum_{i \in S_2} \sum_{j=j_c^i}^{j_{\max}^i} V_{ij} X_{ij} - Y_2 = S_b$$

$$\sum_{i \in S_3} \sum_{j=j_c^i}^{j_{\max}^i} V_{ij} X_{ij} + \sum_{j=13_c}^{j=13_{\max}} (V_{13,j})/2 - Y_3 = S_c$$

¹ The dummy variables X_{45} , X_{55} and X_{66} represent dummy states should a row interdependence be in effect.

$$\sum_{i=1}^{13} \sum_{j=j_c^i}^{j_{\max}^i} V_{ij} X_{ij} - Y_4 = R$$

where

i = the factor index (row of Table 4),

j = the state index (column of Table 4),

V_{ij} = the score associated with the i^{th} feature and j^{th} state (entry in Table 4),

C_{ij} = the total retrofit cost of going from the j_c^{th} state to the j^{th} state,¹ for the i^{th} feature

¹ If $j^i < j_c^i$, corresponding to a regression in score, then

C_{ij} = 20 times the maximum cost, that is, arbitrarily large.

If $j^i = j_c^i$ then

$C_{ij} = 0$ indicating that it is possible to remain in the current state at no cost.

If $j_{\max}^i \geq j^i > j_c^i$ and a potential retrofit is possible.

$$C_{ij} = e_i^! \cdot c_i$$

where

$e_i^!$ = the transpose of the vector of element counts, and

c_i = the vector of element costs

If $j_{\max}^i \geq j^i > j_c^i$ and

j^i precluded by the user then

C_{ij} is made arbitrarily large.

If $j^i > j_{\max}^i$, a dummy state due to a row interdependence, then

$$C_{ij} = 0 \quad \text{and} \quad V_{ij} = 0$$

$$s_1 = (1, 2, 3, 4, 5, 7, 8, 13),^1$$

$$s_2 = (1, 8, 11, 12, 13),^1$$

$$s_3 = (2, 5, 6, 7, 9, 10, 12),^1$$

s_A, s_B, s_C = mandatory safety requirements from Table 6,

R = calculated risk factor from Tables 1, 2, and 3, and,

Y_1, Y_2, Y_3, Y_4 = nonnegative surplus variables representing a requirement excess

2.3 DESCRIPTION OF COMPUTER ALGORITHM

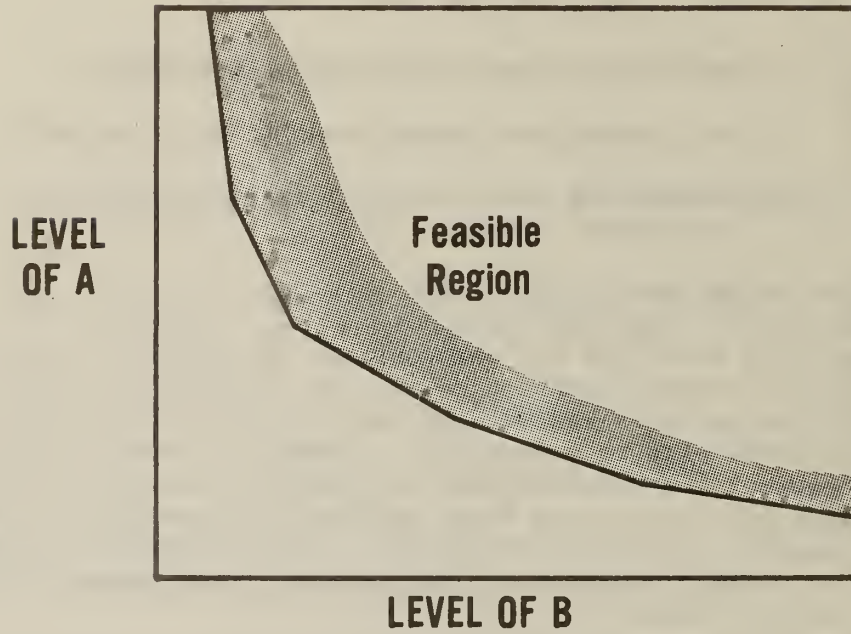
This section will draw upon the mathematical formulation of the problem presented in the previous section to show how an optimal combination of retrofit options can be selected. The way in which alternative solutions are generated will also be presented. In order to focus on the main concepts, a geometrical, rather than an algebraic, approach will be presented. Algebraic discussions which parallel the geometric illustrations will be presented in footnotes as needed. Those readers interested in a programmer oriented discussion of the computer algorithm are referred to Appendix B of this report.

In order to illustrate graphically how the computerized procedure operates, it will be necessary to use an example in which it is assumed that the level of containment safety is a function of the fire rating in minutes of two techniques, "A" and "B." A graphical solution to the least-cost means of achieving a given level of containment safety is shown in Figure 2.2. In the figure, any movement up along the vertical axis indicates higher fire ratings for technique "A." Similarly any movement out along the horizontal axis indicates higher fire ratings for technique "B." Consequently, if one were to construct a straight line which passed through the origin, any movement outward along that line (i.e., a move in the north easterly direction) would indicate a movement to higher levels of containment safety.

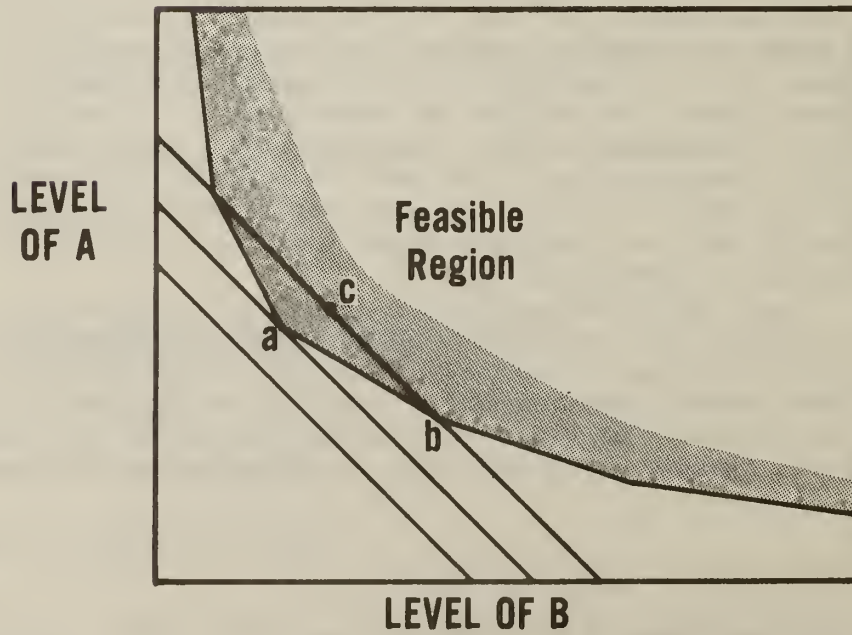
In Part A of Figure 2.2, the lightly shaded region represents all the combinations of fire ratings on techniques "A" and "B" which match or exceed the containment safety requirements of the Life Safety Code. The lightly shaded region in Part A of Figure 2.2 is referred to as the feasible region since all points which lie along its boundary or within are

¹ Numbers in parentheses denote rows of Table 5 of the Fire Zone Safety Evaluation Worksheet.

Part A



Part B



technically feasible.¹ The boundary is all those combinations of the two techniques which exactly satisfy the containment safety requirements of the Life Safety Code.

In Part B of Figure 2.2 a series of equal cost lines are shown. (Recall that an equal cost line shows all the combinations of technique "A" and technique "B" which cost the same. It is drawn based on the assumption that the unit prices for technique "A" and technique "B" are constant.) Higher equal cost lines imply greater costs are being incurred. By referring to Part B of Figure 2.2, it can be seen that the first equal cost line does not touch the shaded region, implying that not enough funds are being allocated to attain the level of containment safety required by the Life Safety Code. The second equal line curve just touches a vertex of the shaded feasible region. Since no other equal cost line which is lower also touches the feasible region, it can be asserted that the point of contact is the least-cost combination of the two factors generating containment safety. Figure 2.2 also illustrates that if more money were spent it would still be possible to achieve the same level of safety, as shown by the vertex of the feasible region labeled "b". However, from the graph, it can be seen that it would be more cost effective to use those dollars to achieve a higher level

¹ Algebraically the feasible region for this case may be expressed as

$$\sum_{i=1}^2 \sum_{j=j_c^i}^{j_{\max}^i} V_{ij} X_{ij} \geq S_{\text{containment}}$$

where $i = 1$ refers to technique "A"
 $i = 2$ refers to technique "B"; and
 $X_{ij} =$ either 0 or 1 and

$$\sum_{j=j_c^i}^{j_{\max}^i} X_{ij} = 1$$

If a nonnegative surplus variable, Y , is included, the inequality can be turned into the following equality

$$\sum_{i=1}^2 \sum_{j=j_c^i}^{j_{\max}^i} V_{ij} X_{ij} - Y = S_{\text{containment}}$$

of containment safety than required by the code. Such a strategy would result in a point lying entirely within the feasible region such as "c."¹

One way in which alternative solutions are generated by the computerized procedure is shown graphically in Figure 2.3.

The procedure actually used in the computer program generates two classes of alternative solutions. Only the first class is illustrated in Figure 2.3. The optimal solution derived earlier is denoted by "a" in Figure 2.3.

For example, suppose X_{ij} appears in the optimal solution as a retrofit (i.e., $X_{ij} = 1$), then the cost, C_{ij} , of going from X_{ijc} , the preretrofit

state, to X_{ij} , the postretrofit state is made arbitrarily high.² This step guarantees that X_{ij} can no longer be in the optimal solution. Graphically this would correspond to a rotation of the equal cost line. The new solution would thus be the point at which the lowest of the new class of equal cost curves just touches the feasible region. Such a point is designated a b in Figure 2.3. Exactly analogous is the case where X_{ij} did not appear in the optimal solution (i.e., $X_{ij} = 0$). In this case, the cost of going from X_{ijc} to X_{ij} , C_{ij} is held fixed and all C_{ik} , $k \neq j$,

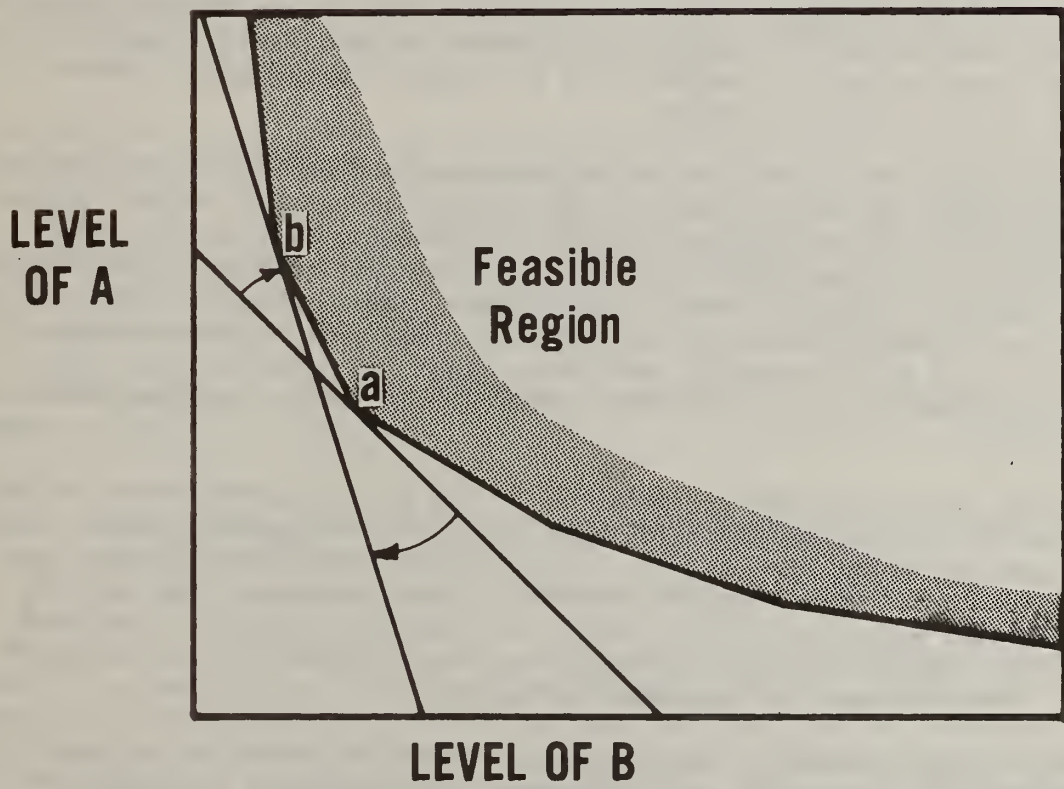
¹ The problem is solved algebraically through the use of the simplex algorithm. As a first step, a simplex tableau is established similar to the one for the simplified problem shown below

X_{11}	X_{12}	X_{13}	X_{14}	X_{21}	X_{22}	X_{23}	X_{24}	$\sum_{i=1}^2 \sum_{j=1}^4 X_{ij}$	
1	1	1	1	0	0	0	0		= 1
0	0	0	0	1	1	1	1		= 1
V_{11}	V_{12}	V_{13}	V_{14}	V_{21}	V_{22}	V_{23}	V_{24}	Y	= $S_{\text{containment}}$
C_{11}	C_{12}	C_{13}	C_{14}	C_{21}	C_{22}	C_{23}	C_{24}		= OBJECTIVE

The method then proceeds by first identifying a basic feasible solution, some vertex of the feasible region. Such a solution could be the highest state for each of the two techniques. The elements, X_{14} and X_{24} , of the basic feasible solution are referred to as a basis. The next step is a basis changing process known as pivoting. Each pivot or basis change is the selection of an adjacent vertex. Elements are removed and included to produce another basis as long as the pivoting process results in reductions of the objective function (retrofit costs). The resulting solution is the minimum value of the objective function.

² An arbitrarily high retrofit cost is defined as one which is 20 times the highest individual retrofit cost.

FIGURE 2.3 ALTERNATIVE SOLUTION: GRAPHICAL METHOD



are made arbitrarily high. This step guarantees that X_{ij} will appear in the optimal solution as a retrofit. The number of solutions generated in this class is equal to the number of variables in the original linear programming problem minus the number of retrofit states which cannot be attained (those within a row having a lesser value than the input and those precluded by an arbitrarily high cost). In the simple case illustrated graphically, up to six alternative solutions could be generated. The second class of alternatives is determined by the rows of Table 4. For example, suppose X_{ij} appears in the optimal solution as a retrofit, then the cost of going from X_{ij_c} to $X_{ij'}$ for all j' , $j_c < j' \leq j_{max}$

where j_{max} is the maximal state in the i^{th} row is made arbitrarily high. This step guarantees that no potential retrofits associated with that factor, say automatic sprinkler systems, can appear in the optimal solution. The number of solutions generated is equal to the number of factors (rows in Table 4) which moved to a higher state (were retrofitted) in the optimal solution.

The two classes of alternative solutions described above are very interesting because they are "close" to the optimal solution. "Close" is put within quotation marks because through reference to Figure 2.3 it can be seen that the alternative solutions are adjacent vertices of the boundary of the feasible region.¹ Thus "close" means a small change in the basis rather than a small change in the objective function. Although these solutions will be more costly than the optimal solution, some of them will be extremely close (in dollar terms) to the least-cost combination of retrofits.

All of the discussion has proceeded as if the continuous solution to the optimization were actually the integer solution. In general, this is not true; however, there are several strong arguments for the approach used. The first is the "near integer" property of the solutions; the second, the judicious selection of the family of alternate solutions; and the third, the computational cost, efficiency, and reliability of the algorithm. Each of these is discussed in turn.

The near integer property occurs because of the structure of the constraint matrix. Ignoring, for the moment, the contrived constraints² a solution will have exactly 17 variables. The 13 equations corresponding to the Building Safety Features in Table 4 have no variables in common and each has a right hand side of 1. Therefore, there are 13 mutually

¹ In n dimensional space the alternative solutions correspond to the vertices of the polyhedron (feasible region) in the neighborhood of the optimal basis (optimal combination of retrofits). Therefore, they should map into a neighborhood about the optimal value of the objective function (least-cost combination of retrofits).

² Experience has shown that the contrived constraints help in reducing the number of non-integers.

exclusive subsets of the 17 variables which must sum to 1. Since it requires at least two fractions to sum to 1, there must be at least nine variables which are exactly one. Nine or more integers among the 13 variables of interest is defined as near integer. Nine integers is the worst case; some of the variables may be zero and/or some of the surplus variables (the Y's in Section 2.2) may be non-zero. Should either condition occur, there are fewer than 17 variables to be included in the 13 subsets and more than ten of the variables may be integer.

Regardless of the integer characteristics of the optimal solution, it represents a lower bound on the total retrofit cost. It is obvious that the introduction of an additional constraint cannot improve the cost. In production work, although not in the development work to date, a heuristic integerization procedure would be applied to each solution generated. This would produce an integer solution by making a minimal change in the basic continuous solution.

One of the criteria used in the design of the alternate solution family was that any variable appearing as a fraction in the optimal solution must appear in at least one of the alternates as one and be suppressed in at least one of the alternates. Other criteria were that the alternates should be amenable to generation via a systematic but comprehensive specification, that they meet (at least our conception of) user needs, and that the number of alternates be adequate but not burdensome.

Thus the solutions presented, while not guaranteed to include the optimum integer solution, do contain:

- (1) a bounding value on the retrofit cost;
- (2) costs corresponding to the alternates most likely to be considered by the user; and
- (3) one or more costs close to the integer optimum.

To solve the optimization problem as an integer model is actually possible. The reasons for selecting a continuous model are completely pragmatic. Computation time for an integer solution can be expected to be an order of magnitude higher. Off-the-shelf software existed for the continuous but not for the integer solution; this is quite important since its impact is not merely on development costs but on reliability and system maintenance costs while operating in the production mode. For production, some satellite software (again off-the-shelf, proven, and reliable) is available which can reduce the costs of alternate solutions by about a factor of three. No analogous method is known for the integer problem.

2.4 APPLYING THE COMPUTERIZED PROCEDURE

Thus far the focus has been on how the computerized procedure operates. The purpose of this section is to build on that base and illustrate how the procedure can be applied in an actual case. Two subsections have been prepared to achieve this objective. The first subsection is con-

cerned with the input portions of the computer program (data requirements and user options). The second subsection is concerned with the output of the computer program. For those readers wishing a more detailed treatment of the computerized procedure, including formatting, documentation and sample computer runs, two technical appendices have also been prepared.

2.4.1 Data Requirements and User Options

Before the computerized procedure can be exercised, all potential retrofits must first be identified. Once this information is available, it is possible to identify and count the elements which require treatment in order to move to a higher state. These data requirements are listed in Table 2.2. Table 2.1 defines the abbreviations for the elements which appear in Table 2.2. Under the assumption that all elements have been counted, it becomes possible to examine the computer input in detail. A sample computer run will be used in order to illustrate the input data requirements.

The first three lines of input specify (1) the initial state of the fire zone, (2) the occupancy risk factor, and (3) information on the building and zone location. All information required for the first line of the input may be taken directly from Table 4 of the Fire Zone Safety Evaluation Worksheet. In particular, each Building Safety Feature has its parameters numbered from 1 to n, where n could be 3, 4 or 5, depending on the feature.¹ The number corresponding to the existing state of each feature is then identified for use as an input. All information required for the second line of input may be taken directly from Table 1 of the Fire Zone Safety Evaluation Worksheet. As in the previous case, each parameter is numbered from 1 to m, where m could be 2, 4 or 5. The number corresponding to the existing state is then identified for use as an input. The first three lines of the sample input are shown below as Exhibit 2.1.

Exhibit 2.1 Initial Inputs

Line	Sample Input
1	3, 1, 1, 2, 1, 2, 3, 1, 1, 3, 2, 1, 1,
2	3, 4, 2, 3, 1
3	1, 2, 3, 3, 32

¹ Recall that "Row 1" of Table 4 is actually a 4 (Floor of Zone) by 3 (Combustible-Wood Frame, Combustible-Ordinary, Noncombustible) by 3 (Unprotected, Protected, Fire Resistant) array. Consequently, "Row 1" as used here refers to the vector whose elements correspond to the Unprotected, Protected, Fire Resistant and Combustible-Wood Frame, Combustible-Ordinary, Noncombustible condition specified in line 3 of the input.

Table 2.1 Abbreviation for Elements and Units to be Used
in Reporting Data on Potential Retrofits

Abbreviation	Definition
Ft ² (C)	Square feet of construction to be protected (e.g., ceilings, walls, columns, or floors)
Ft ² (F)	Square feet of floor area
Ft ² (W)	Square feet of wall area
Ft(P)	Linear feet of partition extended to the structural slab
D	Number of doors
D + F	Number of door and frame systems
O	Number of openings
C	Number of closers
FD	Number of fire doors
FD + F	Number of fire door and frame systems
CPFD + F	Number of corridor partitionings with fire door and frame systems
S	Number of stairwells
#	Number of any element/unit

Table 2.2 Listing of Data Requirements

Factors	Parameters Values			
	Combustible		Noncombustible	
1. CONSTRUCTION	Wood Frame	Ordinary		
FLOOR #	Protected $Ft^2(C)$	Protected $Ft^2(C)$	Protected $Ft^2(C)$	Fire Resist $Ft^2(C)$
2. INTERIOR FINISH [Corr. & Exit]	Class B $Ft^2(F), Ft^2(W)$	Class A $Ft^2(F), Ft^2(W)$		
3. INTERIOR FINISH [Rooms]	Class B $Ft^2(F), Ft^2(W)$	Class A $Ft^2(F), Ft^2(W)$		
4. CORRIDOR PARTITION WALLS	<1/3 Hr $Ft^2(W), 0$	1/3-1.0 Hr $Ft^2(W), 0$	>1.0 Hr $Ft^2(W), 0$	
5. DOORS TO CORRIDOR	<20 min FR D, D+F, 0	>20 min FR D, D+F, 0	>20 min FR D, D+F, 0, C	
6. CORRIDOR LENGTHS	Dead End 30'-100' S	No Dead Ends >30' >150' S	& Corridor Length 100'-150' S	Length is <100' S
7. VERTICAL OPENINGS	Enclosed with Indicated Fire Resist			
	<1 HR $Ft^2(W), FD, FD+F$	≥ 1 HR <2 HR $Ft^2(W), FD, FD+F$	2 HR $Ft^2(W), FD, FD+F$	
8. HAZARDOUS AREAS	Single Deficiency $Ft^2(F), Ft^2(W), FD, FD+F$		No Deficiencies $Ft^2(F), Ft^2(W), FD, FD+F$	
9. SMOKE CONTROL	Smoke Partition $Ft^2(W), CPF+D$	Mechanically Assisted Systems Fire Zone Corridor # #		
10. EMERGENCY MOVEMENT ROUTES	w/o Horizontal Exit[s] #	Multiple Routes Horizontal Exit[s] $Ft^2(W), Ft(P), CPF+D$		Direct Exit[s] #
11. MANUAL FIRE ALARM	Manual Fire Alarm			
	w/o F.D. Conn #	w/F.D. Conn #		
12. AUTOMATIC DETECTION & ALARM	Corridor Only $Ft^2(F)$	Rooms Only $Ft^2(F)$	Corridor & Habit. Space $Ft^2(F)$	Total Space $Ft^2(F)$
13. SPRINKLERS	Corridor $Ft^2(F)$	Corridor & Habit. Space $Ft^2(F)$		Total Space $Ft^2(F)$

The first line of input, shown in Exhibit 2.1, gives the following information: the first feature, Construction, is in the third state; the second and third features, Interior Finishes, are in the first state; the fourth feature, Corridor Partition Walls, is in the second state; the fifth feature, Doors to the Corridor, is in the first state; the sixth feature, Zone Dimensions, is in the second state; the seventh feature, Vertical Openings, is in the third state; the eighth and ninth features, Hazardous Areas and Smoke Control, are in the first state; the tenth feature, Emergency Movement Routes, is in the third state; the eleventh feature, Manual Fire Alarm, is in the second state; and the twelfth and thirteenth features, Automatic Smoke Detection and Alarm and Sprinklers, are in the first state.¹

The second line of input shows the following information on occupancy risk: Patient Mobility is in the third state (not mobile); Patient Density is in the fourth state (>30); Zone Location is in the second state (2nd or 3rd floor); Ratio of Patients to Attendants is in the third state (6-10); and Patient Average Age is in the first state.²

The third line of input shows the following information: the problem number is 1; the building age is 2 (existing); the construction code is 3 (noncombustible); the fire zone is located on the third floor; and the number of patients is 32. The first entry, problem number, may be used in case the floor has more than one zone or if the user wished to change slightly the formulation of the problem to see how the optimal combination of retrofits might change. The latter could be accomplished through the use of one of the three user options.

The second entry gives the building age, 1 = New, 2 = Existing. The third entry gives the construction type, 1 = Wood Frame, 2 = Ordinary, 3 = Noncombustible. The fourth entry, floor of zone, is self-explanatory. The last entry, number of patients, is included because the charge for no smoke control changes if more than 30 patients are in the zone.

¹ The admissible states for each factor are: (1) Construction, 1-3; (2) Interior Finish (Corridor and Exit), 1-3; (3) Interior Finish (Rooms), 1-3; (4) Corridor Partition Walls, 1-4; (5) Doors to the Corridor, 1-4; (6) Zone Dimensions, 1-5; (7) Vertical Openings, 1-5; (8) Hazardous Areas, 1-5; (9) Smoke Control, 1-4; (10) Emergency Movement Routes, 1-5; (11) Manual Fire Alarm, 1-4; (12) Automatic Detection and Alarm, 1-5; (13) Sprinklers, 1-4.

² The admissible states for each (occupancy risk) factor are: (1) Patient Mobility, 1-4; (2) Patient Density, 1-4; (3) Zone Location, 1-5; (4) Ratio of Patients to Attendants, 1-5; (5) Patient Average Age, 1-2.

Lines 4 through 16 of the computer input are patterned after the information contained in Table 4 of the Worksheet. They identify the number of elements which must be treated to move to a higher state. The entries use the format established in Table 2.1. A sample input is shown in Exhibit 2.2. In order to explain the input in some detail, a line-by-line description of Exhibit 2.2 will be given.¹

Note carefully that the only information which must be input on lines 4 through 16 is the factor ID and the element counts to go to a higher state for that factor. When inputting the factor ID and element count information each factor is input on a separate card. (For the eighth factor, Hazardous Areas, two cards, both numbered 8, may be used.) Consequently either 13 or 14 cards are required to input all of the necessary information on the potential retrofits. Thus, if the factor is already in the highest state, the only information required on that card is the factor ID number. It is important to point out that the first line of input (i.e., the first card) contains the current state for each factor. Since the current state dictates the set of potential retrofits for each factor, care must be exercised to ensure that the information recorded on cards 4 through 16 is consistent with the information on the first card.

Exhibit 2.2 Sample Input Information on Potential Retrofits

Line	Factor	Factor ID	Element Counts
4	Construction	1	
5	Interior Finish (C&E)	2	2000, 0, 2000, 0
6	Interior Finish (Rooms)	3	250, 1200, 250, 1200
7	Corridor Partition Walls	4	0, 3, -1, -1
8	Doors to the Corridor	5	0, 0, 34, 0, 34, 0, 0, 0, 34, 0, 0, 34
9	Zone Dimensions	6	-1, 1, -1
10	Vertical Openings	7	0, 0, 3, 0, 0, 3
11	Hazardous Areas	8	-1, -1, -1, -1, 350, 0, 0, 0, -1, -1, -1, -1, 700, 0, 0, 7
12	Smoke Control	9	0, 1, -1, -1, -1, -1, -1
13	Emergency Movement	10	0, 8, 1, -1, -1, -1
14	Manual Fire Alarm	11	1
15	Auto Detection & Alarm	12	2000, 6000, 8000, 8700
16	Sprinklers	13	2000, 8000, 8700

¹ An element count entry of "-1" indicates an impossible retrofit.

Line 4: Construction; Building Safety Feature ID: 1

The first column of the fourth card contains the single entry 1 indicating that this is the first feature (Construction).¹ The only element count entry for this feature is a blank space indicating that this feature is already in the highest state and can stay there at zero cost.

Line 5: Interior Finish (Corridor and Exit); Building Safety Feature ID: 2

The first and second element count entries show that to move to the next highest state, Class B, 2000 square feet of floor area and 0 square feet of wall area must be treated. The third and fourth element count entries show that to move to the highest state, Class A, 2000 square feet of floor area and 0 square feet of wall area must be treated.

Line 6: Interior Finish (Rooms); Building Safety Feature ID: 3

The first and second element count entries show that to move to the next highest state, Class B, 250 square feet of floor area and 1200 square feet of wall area must be treated. The third and fourth element count entries show that to move to the highest state, Class A, 250 square feet of floor area and 1200 square feet of wall area must be treated.

Line 7: Corridor Partition Walls; Building Safety Feature ID: 4

The first and second element count entries show that 0 square feet of corridor partition wall must be installed and 3 openings in the corridor partition walls require treatment to upgrade to a 1/3 to 1 hour fire rating (the third state in Table 4 of the Worksheet). The third and fourth element count entries indicate that going to a fire rating of more than 1 hour is impossible (the highest state in Table 4 of the Worksheet).

Line 8: Doors to Corridor; Building Safety Feature ID: 5

The first, second, third and fourth element count entries indicate that going from the "No Door" charge to a fire rating of less than 20 minutes would require the replacement of no doors and no complete door and frame systems, but would require that the openings (e.g., ordinary glass lites) in 34 doors be treated. The fourth element count is a place holder for automatic closers used only in the highest retrofit state. The fifth, sixth, seventh and eighth element count entries show that to go to a fire rating of more than 20 minutes would require 34 doors to be replaced, no complete door and frame systems to be replaced and no opening to be

¹ In every case, for cards 4 through 16 the first entry will be the factor ID.

treated.¹ Again a 0 is entered for the eighth element count since it is used as a place holder only. The ninth, tenth, eleventh and twelfth element count entries show that to go to a fire rating of more than 20 minutes with automatic closers would require 34 doors to be replaced, no complete door and frame systems to be replaced, no openings to be treated and 34 closers to be installed.

Line 9: Zone Dimensions; Building Safety Feature ID: 6

The first element count entry indicates that going to a situation with no dead ends greater than 30 feet and a corridor length greater than 150 feet is an impossible retrofit. The second element count entry indicates a potential retrofit, the installation of a stairwell to remove the dead end (the corridor length remains the same). The third element count entry indicates an impossible retrofit (no dead ends greater than 30 feet and a corridor length of less than 100 feet).

Line 10: Vertical Openings; Building Safety Feature ID: 7

The first, second and third element count entries show that 0 square feet of fire proofing, no fire doors, and 3 complete fire door and frame systems (Class B) are required to upgrade to a fire rating of between 1 and 2 hours. The fourth, fifth and sixth element count entries show that 0 square feet of fire proofing, no fire doors, and 3 complete fire door and frame systems (Class A) are required to upgrade to a fire rating of more than 2 hours.

Line 11: Hazardous Areas; Building Safety Feature ID: 8

The first, second, third and fourth element count entries indicate an impossible retrofit.² The fifth, sixth, seventh and eighth element count

¹ Note that an entry for doors and frames is included. In this case all doors can be fitted onto the existing frames. If the present frames were incompatible with the replacement doors then the first eight element count entries would be: 0, 0, 34, 0, 0, 34, 0, 0.

If 20 doors could be installed on existing frames, but the remaining 14 would require new frames, then the first eight element count entries would be: 0, 0, 34, 0, 20, 14, 0, 0.

² It is important to point out that, unless some a priori retrofit scenario is hypothesized which makes explicit the cost dependence among hazardous areas in other fire zones, the costs of treating deficiencies in other fire zones will be assumed to be independent of the costs of treating deficiencies within the fire zone. Consequently no explicit cost charge for the treatment of hazardous areas in other fire zones is entered into the computer program. This approach implies that hazardous area retrofits in other fire zones are treated as impossible retrofits for the fire zone under consideration.

entries indicate that to move to a single deficiency 350 square feet of floor area must be sprinklered, 0 square feet of wall area must be fire proofed, no fire doors are required, and no complete fire door and frame systems are required. The ninth, tenth, eleventh and twelfth element count entries show that to remove all deficiencies in hazardous areas 700 square feet of floor area must be sprinklered, 0 square feet of wall area must be fire proofed, no fire doors are required, and 7 complete fire door and frame systems are required.

Line 12: Smoke Control; Building Safety Feature ID: 9

The first and second element count entries indicate that to install a smoke partition requires 0 square feet of wall area to be fire proofed and 1 (8 foot wide) section of corridor partitioning with fire doors to be installed. The third, fourth, fifth and sixth element count entries indicate that both mechanically assisted system retrofits are impossible.

Line 13: Emergency Movement Routes; Building Safety Feature ID: 10

The first, second and third element count entries indicate that the installation of a horizontal exit requires 0 square feet of wall to be fire proofed, 8 linear feet of corridor partition to be extended to the structural slab, and 1 (8 foot wide) section of corridor partitioning with fire doors to be installed. The fourth, fifth and sixth element count entries indicate that the installation of direct exits is an impossible retrofit.

Line 14: Manual Fire Alarms; Building Safety Feature ID: 11

The single element count entry indicates that to move to the highest state requires the installation of a fire department connection.

Line 15: Automatic Detection and Alarm; Building Safety Feature ID: 12

Protection of corridors, the first element count entry, requires 2000 square feet of floor area to be treated. Protection of rooms, the second element count entry, requires 6000 square feet of floor area to be treated. Protection of corridor and habitable spaces, the third element count entry, requires 8000 square feet of floor area to be treated. Total protection, the fourth element count entry, requires 8700 square feet of floor area to be treated.

Line 16: Sprinklers; Building Safety Feature ID: 13

The first element count entry, sprinklering the corridor, requires 2000 square feet of floor area to be sprinklered. The second element count entry, sprinklering the corridor and habitable spaces, requires 8000 square feet of floor area to be sprinklered. The third element count entry, total sprinklering, requires 8700 square feet of floor area to be sprinklered.

The options available to the user to control or modify the data input begin with line 17. The admissible control/modification statements are defined in Table 2.3. Each statement will now be examined.

Table 2.3 User Options: Control and Modification Statements

Control Statement	Function
SOLVE	Indicates that no further modifications to the fire zone are desired and to begin calculation of the family of solutions.
NEXT	Read in "lines" 1 through 16 for the next fire zone (see Exhibits 2.1 and 2.2).
END	Terminates run; no more fire zones to be analyzed.
Modification Statement	Function
CHANGE α, β, γ	α = Building Safety Feature ID number β = State number γ = Option code The costs associated with feature α , state β are modified $\gamma < 0$ + EXCLUDE $\gamma = 0$ + INCLUDE $\gamma > 0$ + COST
REQUIR δ, ϵ	The value of mandatory safety requirement δ is increased by ϵ percent. $\delta = 1$ for S_a $\delta = 2$ for S_b $\delta = 3$ for S_c $\delta = 4$ for R

In order to provide the user with the maximum flexibility, it was necessary to require the user to exercise control over the program. Thus all of the Control/Modification cards (C/M cards) are of the same format. The user is required to supply a SOLVE card for every fire zone (or variation thereof) for which a solution is desired. The user is also required to supply a NEXT card when all of the required variants for a particular fire zone have been solved and solutions for another fire zone are desired. The END card terminates the run.

CHANGE and REQUIR cards are user options only. There is no requirement that they appear in the run sequence. If used, they must appear (in any order or combination) prior to an END card. That is to say SOLVE's and NEXT's may be legitimately interspersed throughout the run sequence. It is important to point out that the CHANGE and REQUIR cards are defined in such a way that their effect on the run sequence is cumulative. Thus the order in which the options appear controls the definition of the fire zone problem solved. Consequently, the user is encouraged to use the SOLVE card as the first C/M card in the run sequence. This will guarantee a solution to the original problem.

A sample C/M card sequence is given in Exhibit 2.3 as an illustration. As a means of reinforcing the functions of the C/M cards defined earlier, a line-by-line description of the solution generated by the C/M card sequence given in Exhibit 2.3 will be presented.

Exhibit 2.3 Sample Control/Modification Card Sequence

Control/Modification Card or Descriptor	Control/Modification Card Number	Probable Run Sequence Card Number
INPUT ON FIRST FIRE ZONE	-	1-16
SOLVE	1	17
CHANGE 1 3 -1	2	18
CHANGE 4 2 0	3	19
REQUIR 1 10	4	20
CHANGE 6 2 17500	5	21
SOLVE	6	22
CHANGE 9 3 -1	7	23
SOLVE	8	24
REQUIR 1 10	9	25
NEXT	10	26
INPUT ON SECOND FIRE ZONE	-	27-42
CHANGE 3 1 -1	11	43
SOLVE	12	44
SOLVE	13	45
END	14	46

C/M Card 1: SOLVE

This card causes the problem defined by run sequence card numbers 1 through 16 to be solved. The solution includes: (1) the least-cost means of achieving compliance to the Life Safety Code; (2) the cost of strict compliance to the Life Safety Code; and (3) a set of alternative means of achieving compliance to the Life Safety Code.

C/M Card 2: CHANGE 1 3 -1

This card causes state 3 of the first feature (Construction) to be precluded in the solution. Specifically, it implies that C_{13} (the cost of feature 1, state 3) is made arbitrarily large.

C/M Card 3: CHANGE 4 2 0

This card causes state 2 of the fourth feature (Corridor Partition Walls) to be included in the retrofit package. This is accomplished by making C_{41} , C_{43} and C_{44} arbitrarily large. Thus the second state is forced into the section.

C/M Card 4: REQUIR 1 10

This card causes the minimum passing score associated with the first safety redundancy requirement, S_a (Containment), to be increased by 10 percent. This option is useful in those cases where the decision maker wishes to purchase extra units of safety in a particular category. For example, due to the configuration of the fire zone or for some other a priori reason, the decision maker wishes to purchase higher levels of Containment Safety than provided by the Life Safety Code. The linear programming package ensures that this is done in the most least-costly manner.

C/M Card 5: CHANGE 6 2 17500

This card changes the cost of upgrading from the existing state (specified on run sequence card 3) to state 2 of the sixth feature (Zone Dimensions) to \$17,500.

C/M Card 6: SOLVE

This card causes the problem defined by run sequence card numbers 1 through 21 to be solved (see Exhibit 2.3).

C/M Card 7: CHANGE 9 3 -1

This card causes state 3 of the ninth feature (Smoke Control) to be precluded.

C/M Card 8: SOLVE

This card causes the problem defined by run sequence card numbers 1 through 21 and modified by run sequence card number 23 to be solved.

C/M Card 9: REQUIR 1 10

This card causes the minimum passing score associated with the first safety redundancy requirement, S_a (Containment), to be increased by 10 percent. Since this C/M card was already used (C/M Card 4) any solution would exceed the Containment Safety requirement by 21 percent. Note, however, that no SOLVE C/M card follows so that no solutions will be generated.

C/M Card 10: NEXT

This card clears all options from the program and initiates the reading in of input for the next fire zone.

C/M Card 11: CHANGE 3 1 -1

This card causes state 1 of the third feature (Interior Finish: Rooms) to be precluded.

C/M Card 12: SOLVE

This card causes the problem defined by run sequence card numbers 27 through 42 and modified by run sequence card number 43 to be solved. Note that the original problem was not solved since state 1 of feature 3 was precluded by C/M card 11.

C/M Card 13: SOLVE

This card generates the same output as C/M card 12. It illustrates the fact that care must be exercised by the user since legitimate commands if combined in the wrong way may generate meaningless output. Note that if the first SOLVE card had preceded the CHANGE 3 1 -1 card, C/M card 13's output would differ from the output generated by the first SOLVE card.

C/M Card 14: END

This card causes the run sequence to terminate.

It is important to reiterate that modifications are cumulative between fire zone "reads" (NEXT statements). Any number of fire zones is permissible. That is, there is no a priori limit on the number of NEXT cards. (Computer system limits such as page counts or CPU time, however, will limit the number of problems which can be solved.) Furthermore, there is no limit on the number of modifications to a fire zone as long

as the problem remains feasible. If a particular modification is not feasible, nothing catastrophic occurs; some computer time, but not a great deal, is used. As pointed out in the sample run sequence, the problem defined by the fire zone input (run sequence lines 1 through 16 and 27 through 42 in Exhibit 2.3) does not have to be solved. Unless the user has a good reason, the basic problem should be solved prior to any modification of the data input.

The set of solutions for a modified problem is the same as that for an unmodified problem except that those states which have been "forced in" (CHANGE i j 0; denoted as an "F" type change on the computer output) are not allowed to leave the basis.

2.4.2 The Output Format and Its Interpretation

The output format for each problem solved begins with a printback of Table 1 and Table 4. This printback is intended as a check so that the user can verify that the problem actually solved corresponds to the one intended. After the printback, information on the least cost and prescriptive solutions is given. This information identifies each of the 13 factors, the state number, the retrofit cost, the score, and one or more solution ID's. Five solution ID's are possible:

- (1) S indicating that the state is in the optimal solution;
- (2) I indicating that the state is the one input;
- (3) P indicating that the state is in the prescriptive solution;
- (4) F indicating that the user modification which forces a state into the solution has been performed; and
- (5) C indicating that a cost change modification has been performed.

The information shown in Exhibits 2.1 and 2.2 and C/M card 1 in Exhibit 2.3 will generate the output given in Exhibit 2.4.¹ Notice that under feature 9, smoke control, an entry is marked by an asterisk. This is to indicate that the installation of a horizontal exit under feature 10 allows a smoke partition to be claimed at no cost under feature 9.²

¹ As was mentioned earlier, the continuous solution to the optimization problem may contain noninteger variables. However, given the "near integer" property of the solutions, this problem is easily handled. Consequently, the output shown in Exhibit 2.4 is not the output as it comes from the computer directly. It represents the result of a post-optimization integerization procedure which is presently done manually. As indicated in Section 2.3, in production work the user will receive an output like that presented in Exhibit 2.4.

² The logic required to treat this interdependency is currently not in the computer program. Should a horizontal exit be installed, however, in the least-cost retrofit package, the user can claim two additional points in People Movement Safety and General Safety.

Exhibit 2.4 Sample Output Format for Optimal and Prescriptive Solutions

FACTOR	TABLE 4	COST	SCORE		TOTAL COST 20400
1. CONSTRUCTION	3	0	4	SIP	SA = 13 SURPLUS = 5
2. INTERIOR FINISH CORRIDOR & EXIT	1	0	-5	I	SA = 5 SURPLUS = 0
	2	800	0	P	SC = 5 SURPLUS = 2
	3	800	3	S	SG = 13 SURPLUS = 6
3. INTERIOR FINISH ROOMS	1	0	-3	I	PRESCRIPTIVE 38000
	2	600	1	P	
	3	600	3	S	
4. CORRIDOR PARTITION WALLS	1	0	0	SIP	SA = 8 SURPLUS = 0
	2	0	0	SIP	SB = 5 SURPLUS = 0
	3	0	0	SIP	SC = 3 SURPLUS = 0
5. DOORS TO CORRIDOR	1	0	-10	I	SG = 9 SURPLUS = 2
	2	3000	0	S	
	3	9000	1	P	
6. ZONE DIMENSIONS	2	0	-4	SI	
	4	15000	0	P	
7. VERTICAL OPENINGS	3	0	0	I	
	4	3750	2	P	
	5	3900	3	S	
8. HAZARDOUS AREAS	1	0	-11	I	
	5	7450	0	SP	
9. SMOKE CONTROL	1	0	-2	I	
	2	1400	0	P	
	2*	0	0	S	
10. EMERGENCY MOVEMENT ROUTES	3	0	0	IP	
	4	4650	3	S	
11. MANUAL FIRE ALARM	2	0	1	SIP	
12. SMOKE DETECTION AND ALARM	1	0	0	SIP	
13. SPRINKLERS	1	0	0	SIP	

The output format is designed to parallel Table 4 of the Worksheet. All entries marked with a 0 correspond to the initial state of that feature. These entries are marked with an I in Exhibit 2.4. All entries marked with an s and a number other than 0, except for the smoke partition marked as 0*, are the retrofits included in the optimal solution. All entries marked with a p and a number other than 0 are the retrofits included in the prescriptive solution. Counting from left to right in Table 4 (Exhibit 1.2) gives the state number in each case. This number is entered under the column labeled TABLE 4 in Exhibit 2.4. The cost figures associated with each possibility ($I = 0$ and $S, P \geq 0$) are then recorded in the column labeled COST in Exhibit 2.4. The cost figure which appears beside the state number is the total retrofit cost. It is based on the information stored in the element count and element cost matrices. The figure entered under the column SCORE in Exhibit 2.4 is taken directly from Table 4 of the Worksheet (see Exhibit 1.2). Note also that in the upper right hand corner of the output is printed out a cost summary statement. In this case the cost of equivalence is \$20400 whereas the cost of strict compliance is \$38000. The use of the equivalency methodology thus results in a cost reduction of almost 50 percent for this fire zone. Immediately below the cost summary are the scores for each of the four safety categories. Surplus scores are given to the right of each category.

The output format and its interpretation for each of the alternative solutions is exactly the same. The parameter or factor being forced into or out of the solution set is marked with an asterisk, *, to the left of the solution ID's (S, I, P, F, or C).

For modified fire zones (those for which at least one CHANGE or REQUIR modification has been performed) the solution ID will contain a C (in addition to S, I, P and *) if the cost has been changed by the user to reflect local market conditions (i.e., a better cost estimate) or the option has been forced out (i.e., $\gamma < 0$ in Table 2.3) by an arbitrarily high cost. If the state has been forced in (i.e., $\gamma = 0$ in Table 2.3) an F will appear in the printout. The alternative solutions, whether for "input" or modified fire zones, are the same as the original solutions except that a P no longer appears in the solution ID and the sub tableau, labeled PRESCRIPTIVE in Exhibit 2.4, which shows the prescriptive cost and requirement surpluses (if any) will not appear.

3. SUMMARY

The identification of cost-effective levels of fire safety in health care facilities is a major concern to hospital administrators, fire safety engineers and public policy makers. Rising construction and operating costs coupled with more stringent building codes and continuing advances in medical and building technology have complicated the issue, forcing health care facility administrators to carefully assess the alternative means through which they can design, construct or update their facilities. The Life Safety Code developed by the National Fire Protection Association (NFPA) is currently the most widely used guide for identifying the minimum level of fire safety in health care facilities. Although the

Life Safety Code may be thought of as a prescriptive code since it prescribes fixed solutions for life safety, performance concepts can be explicitly introduced through a provision which allows for equivalent solutions. In light of this provision, the National Bureau of Standards' Center for Fire Research through support from the Department of Health, Education and Welfare has developed a system, the Fire Safety Evaluation System, for determining how combinations of several widely accepted fire safety systems could be used to provide a level of safety equivalent to that required in the 1973 Life Safety Code.

This report has provided a programmer-oriented description of the mathematical, economic, and engineering considerations that went into the development of a computerized version of the Fire Safety Evaluation System which permits the least cost means of achieving compliance to the Life Safety Code to be identified. It was shown that each of the parameters used in the Fire Safety Evaluation System had a unique value which corresponded to strict compliance to the Life Safety Code. Consequently, it was possible to quantify the cost savings attributable to a performance-based Fire Safety Evaluation System approach over that of strict compliance.

In the discussion of data requirements, it was also shown that the computer program uses as its primary input information collected as an integral part of a thorough fire safety evaluation. This information permits the current state of the health care facility to be unambiguously identified. Based on (1) the current state of the health care facility, (2) the minimum passing "score" needed to achieve compliance, and (3) the anticipated costs of each retrofit measure, the least cost or optimal combination of retrofits is identified. The computer program then analyzes other retrofit combinations to see if alternatives might exist which are close to the one identified as optimal. The optimal combination of retrofits and any alternative which the program produces, usually between 10 and 20, are then summarized in tabular form and ranked from least costly to most costly. By using this approach, health care facility decision makers have greater flexibility in choosing among retrofit combinations. In particular, by providing alternatives, the decision maker has the opportunity to assess the impact that non-construction costs would have on the choice of the optimal retrofit combination.

APPENDIX A

Unit Pricing Information

The unit pricing information presented in this section is based on anticipated costs in the Washington, D.C. area for the last Quarter of 1978. These figures are used as a resident element cost matrix for the computerized procedure discussed earlier in the report. They can be used to identify the least cost combination of the retrofits for a single zone as well as the costs of strict compliance and any alternative retrofit strategies. It is important to point out that the unit prices presented in this section are adequate only for the purpose of budget planning prior to the request for construction design. The actual cost estimates for a particular retrofit strategy will have to be made following the completion of the retrofit specification, design and material selection, and the establishment of the retrofit date. Most of the unit pricing information presented in this section was either obtained from the Central Office of the Veteran's Administration or the Building Construction Cost Data 1978 booklet published by the Robert Snow Means Company, Inc. The unit prices are IN-PLACE PRICES which include: demolition, waste removal, building materials, labor, installation and finishing, refinishing of adjacent areas and a markup for contractor overhead and profit. Similar unit pricing information can also be developed by local estimators for various retrofit strategies at specific locations. This information can be used to redefine the element cost matrix. The unit pricing information used in the computer program is given as follows:

Coat with Fire-Retardant Paint	\$0.40 Per Square Foot; (PSF)
Fireproof with Intumescent Mastic	
1/8" thick, 3/4 hour	\$1.30 PSF
3/16" thick, 1 hour	\$2.70 PSF
7/16" thick, 2 hour	\$3.90 PSF
Fireproof Existing Wood Panel	\$0.50 PSF
Replace Carpeting	\$1.60 PSF
Install Drywall Including Painting	
1/2" thick	\$0.75 PSF
5/8" thick	\$0.80 PSF
Replace Door to Achieve 20 Minute or More Fire Resistance	
4'x7' door and hardware	\$ 300 each
4'x7' door, frame and hardware	\$ 700 each
7'x7' door, frame and hardware	\$1100 each
Add automatic closing device	\$ 60 each

Replace Door to Achieve Class B & C
Fire Rating

4'x7' door and hardware	\$ 450 each
4'x7' door, frame and hardware	\$1,250 each
7'x7' door and hardware	\$ 800 each
7'x7' door, frame and hardware	\$1,650 each

Replace Door to Achieve Class A
Fire Rating

4'x7' door and hardware	\$ 500 each
4'x7' door, frame and hardware	\$1,300 each
7'x7' door and hardware	\$ 900 each
7'x7' door, frame and hardware	\$1,750 each

Replace Ordinary Glass Light With
Wire Glass Light \$ 40 each

Replace Defective Latch \$ 60 each

Replace Window Frames in Corridor
Partition Walls (1000 square
inches) \$ 400 each

Add Corridor Partitioning (8' wide)
With Fire Door

Class B and C Rating	\$3,800 each
Class A Rating	\$3,900 each

Provide Partition Extension
(average 3' high) From Existing
Partition to Structural Slab
Above \$ 30 Per Linear Foot; (PLF)

Install Coded Manual Fire Alarm Station \$ 140 each

Connect Existing Manual Fire Alarm
Stations to Fire Department \$1,000 Per Fire Zone

Install Detectors, Photoelectric
and Ionization \$ 2.25 PSF

Install Magnetic Door Release Each Leaf \$ 425 each

Install Sprinklers in Nonhazardous
Areas \$ 2.00 PSF

Install Sprinklers in Hazardous Areas \$ 2.25 PSF

Add a Fire Alarm to the Sprinkler	\$ 225 each
Install Halon Head for Mechanical and Electrical Areas	\$ 8,000 each
Add External Stairwell for Exit	\$15,000 Per Floor

APPENDIX B

Documentation of Computer Algorithm/Program

This Appendix gives a programmer-oriented description of the computer software package. Algorithmic and functional descriptions are contained in the body of this appendix as are the user requirements and options. The appendix consists of four sections:

- 1) Functional descriptions of each subroutine and major functional block of the main program;
- 2) Definitions of each of the variables;
- 3) Definitions of input; and
- 4) Descriptions of each report produced.

B.1 Program Functional Description

A brief description of the function of each of the subroutines is given in this section. In addition, the calling sequence, the reports produced, and the program from which it is called are also given. Since many of the scalars in the calling sequence are array dimensions, a statement containing each multi-dimensional array is given as well. For the main program, a similar description is given by main functional blocks. The relevant variables are defined and the subroutines called within each block are identified. All variables except the Hollerith ones conform to the FORTRAN real-integer name convention.

Subroutine MOTEST: MODification TEST:

Calling Sequence: (HFORCE, I1, I2, I4)

I1 is the simplex variable corresponding to the first state (Table 4) of row I, where I is implied by the value of I1.

I2 is the simplex variable corresponding to the last state (Table 4) of row I.

I4 is subroutine output.

MOTEST is called from MAIN in each of the Alternate Solution Blocks. If the variation of the cost row dictated by the alternate solution logic is not compatible with user modifications, I4 is 0; otherwise I4 is 1.

Subroutine PRSOL: PRint SOLution:

Calling Sequence: (IPROB, IALT, C, Z, HFORCE, IST4, N1, N2, IT4X, X, IT4S, N3, HPAR, N4, N5, N6)

DIMENSION: IST4 (N1, N2), IT4S (N3, N2), HPAR (N4, N5), A (N4, N6)

PRSOL is called from MAIN, once from the Bounding Solution Block and once from each of the Alternate Solution Blocks. It, with repeated calls to RHTAB, prints a Solution Report if the solution is valid. If the solution is not valid (i.e., non-feasible, contains an illegal state, etc.), it prints a self-explaining error report and sets IALT to its negative.

Subroutine RHTAB: Right Hand TABLEs: Calling Sequence: (K, X)

RHTAB is called only from PRSOL. K is a variable set to 0 upon entry to PRSOL. It prints the cost-requirement surplus tableaux of the solution report. K is a line-space count advanced in the subroutine.

Subroutine PRSMX: Print SimpleX

Calling Sequence: (A, IA, IB, HPAR, IC, IST4, ID, IE, IALT)

DIMENSION: A (IA, IB), HPAR (IA, IC), IST\$ (ID, IE)

PRSMX prints the Simplex Matrix Report. It is called from MAIN just prior to the call to RVSMXP which produces the bounding solution. If IALT is zero or one, the multi-page report is printed. For any other value of IALT, the one page version is printed.

Subroutine FEASIB: FEASIBILITY Test

Calling Sequence: (A, I1, I2, X, L, X2)

DIMENSION: A(I2, I1)

FEASIB is called from MAIN immediately after each RVSMXP call. RVSMXP returns in L(3) an indicator of an abnormal solution. However, since RVSMXP cannot recognize an illegal variable (i.e., one which has an arbitrarily high cost), FEASIB checks the solution for any such variable and returns L(3) = 2 if one is encountered.

Subroutine RVSMXP: Revised SimpleX

Calling Sequence: (A, MA, B, MB, MT, NT, L, X, TOLP, INV, K9P)

DIMENSION: A(MA, NT), B(MB, MB)

The description of the RVSMXP routine given in this section is application oriented. For those readers wishing a complete documentation of the RVSMXP routine, the report by Hall, et al., is highly recommended.¹ Several

¹ W. G. Hall, R. H. F. Jackson, P. B. Saunders, The National Bureau of Standards Linear and Quadratic Programming Subroutines, National Bureau of Standards Report 10695, February 1972.

of the variables in the calling sequence are fixed in this problem. The vector L is used internally for several purposes but for this discussion L(1) through L(14) are the only relevant entries. Upon input, L(1) is the number of less than or equal to constraints of A (Zero), L(2) is the number of greater than or equal to constraints of A (Zero), and L(3) through L(14) are print switches. For this application, L(5) and L(14) are equal to one; the remaining elements are zero. This combination of switch settings suppresses all internal prints except warning and error messages.

Upon output L(1) through L(3) only will be changed. L(1) is of no interest; L(2) is the iteration count; and L(3) indicates the cause of termination:

- 0 → optimal solution obtained
- .1 → optimal solution obtained but numerical difficulties encountered
- 2 → non-feasible problem
- 3 → unbounded problem
- 4 → system error.

X(81) upon exit contains the negative of the minimum cost solution.

INV is the last of the variables of interest to the user. If INV is zero, the computation will start from scratch; otherwise, the vector X is used as a starting solution. In the program, INV is zero for the bounding solution call and one for all other calls.

MAIN - Initialization Block

This block performs system initialization; it includes the first executable statement through statement number 70. It produces the System Status Report and constructs that portion of the A matrix which is Fire Zone independent. The Fire Zone independent portion of A consists of the 13 state constraints corresponding to the building safety factors of Table 4 and, except for the right hand side, the mandatory safety requirement constraints corresponding to Table 5.

MAIN - Fire Zone Input Block

This block performs the Fire Zone initialization. It contains statements 100 through 170. Statement 100 is the reentry point when a NEXT is encountered. The functions comprising the initialization are:

Fire Zone Input

Modification of A to accommodate row interdependencies

Modification of A to accommodate alternate fire safety level values
Incorporation of mandatory safety requirements into A
State change cost calculations and their incorporation into A
Calculation of prescriptive cost
Calculation of "arbitrarily high" costs
Printing of Fire Zone Report

The Fire Zone Input and Printing functions are dispersed throughout the block; the sequence is essentially either read-print or read-calculate-print for each line of the report. Other functions performed are isolated within the block.

The default row column interdependencies are incorporated into A in the section from statement 902 through statement 903. It is assumed at this time that the interdependencies defined by Table 4 footnotes *, **, and . . . will not occur; thus constraints 14, 15, and 16 are each set to a dummy variable which equals 1.

The Table 1 indices are read (and defaulted to highest risk state if necessary), R is calculated, a prototype solution ID is constructed, the problem-age input is read and the report print initiated in the next section. Statements run from 900 to 973.

The next section, up to statement 152, modifies the fire safety level values (rows 17-20) of the A matrix according to the Table 4 footnotes ***, ., and .. . Then to statement 154 is a section which incorporates into A the interdependency defined by *, if required by the input. The section from 154 to 972 computes F according to the building age, selects the appropriate entries for both row 1 of Table 4 and the mandatory safety requirements and incorporates them into A.

The section from 972 through 110 modifies the solution ID's (HFORCE) by appending "Input" and "Prescriptive" characters; then proceeds, by building safety factor, to read the element cost vectors, calculate the state transition costs, and print the Fire Zone Input Report.

The section through 170 then calculates X1 and X2, the two "arbitrarily high" costs used. X1 is twenty times and X2 is nineteen times the largest legitimate state transition cost. X1 goes into C where required, C is incorporated into A, and L and Z are constructed as subsequently required by subroutines.

MAIN - Control and Modification Block

This block performs the control card reads. For CHANGE and REQUIR cards, there are internal loops for updating C, A, and HFORCE as required.

There are multiple exits from this block depending upon the control card which causes the exit. Exit may be to termination (END), to the Fire Zone Input Block (NEXT), or to the Bounding Solution Block (SOLVE). The statement numbers are 300 to 370.

MAIN - Bounding Solution Block

This block calls PRSMX, RVSMPX, FEASIB, and PRSOL for the bounding solution. It also transfers the bounding solution into Y so that the alternative solutions can be calculated from an existing basis. The normal exit is to the First Alternate Class Block, but there is an error exit (non-feasible, unbounded, etc.) to the Fire Zone Input Block since a no solution case for the bounding problem implies a no solution case for all alternates. This block begins at statement 370 and includes all statements through the fourth (unnumbered) statement past 178.

MAIN - First Alternate Class Block

This block follows the Bounding Solution Block and extends through statement 230. It generates the class of solutions obtained by precluding, in turn, any retrofit for each of the Table 4 building safety factors. It calls MOTEST, RVSMPX, FEASIB, and PRSOL. It always exits to the Second Alternate Class block.

MAIN - Second Alternate Class Block

This block follows statement 230 and extends through 270. It generates the second class of alternate solutions by switching the variables, in turn, of the bounding solution. That is, if a variable is in/out the bounding solution an alternate is produced with the variable forced out/in. It calls MOTEST, RVSMPX, FEASIB, and PRSOL. It always exits to the Restore Solution ID Block.

MAIN - Restore Solution ID Block

This block begins at 280 and continues to the end of the program. Its function is to transfer Y into X and restore the solution ID for re-entry. It always exits to the Control-Modification Read Block.

B.2 Variable Definition

Scalars

- I_{PROB}: Problem number (input)
- N_{AGE}: Building age code (input)
- N_{CON}: Construction type code (input)
- I_{FL}: Floor of fire zone (input)
- I_{PA}: Number of patients in zone (input)
- X₁: The arbitrarily high cost used to preclude state changes. It is calculated in the Fire Zone Input Block as 20 times the largest legitimate state transition cost.
- X₂: .95 x X₁. X₂ is used as a comparand in checking that no arbitrarily high costs appear in the objective function.
- I_{ALT}: The alternate problem identification

Arrays

- A(21,61): The Simplex matrix-dimensions as required by RVSMPX. Rows 1-13 contain the Building Safety Feature constraints; rows 14-16, the interdependency constraints; rows 17-20, the mandatory requirements, and row 21, the objective function. The matrix is initially constructed from G via the structure of Table 4 in the System Initialization Block. Costs (row 21), alternate values (rows 17-20), interdependency constraints (rows 14-16), and the proper row 1 are inserted in the Fire Zone Input Block. Costs are modified by CHANGE inputs; the right hand side of the mandatory requirement constraints are modified by REQUIR inputs.
- B(23,23): This matrix is required by RVSMPX. Its content is not relevant for the user.
- X(150): This vector is required by RVSMPX. It contains the basic solution; upon exit from RVSMPX, the first sixty elements hold the variables. A positive element implies the variable is basic; a negative, that it is non-basic; X(81) contains the negative of the minimum cost. Upon input to RVSMPX, if the advance start option is used, X contains a feasible basis. Elements of X with an index higher than 81 are used to store the prescriptive cost and surpluses for PRSOL and RHTAB.
- Y(150): This vector is used to store the bounding value solution. It is used to restore X after each alternate solution so that the advance start option may be used.
- L(153): This vector is required by RVSMPX. Only the first fourteen elements are of interest to the user. Upon input to RVSMPX, these are all zero except L(5) and L(14), which are each 1; they are used to control warning and error prints internal in RVSMPX. Upon exit, L(1) through L(3) may be non-zero, but only L(3) is important. It contains an indicator as to whether RVSMPX exited with an optimum solution, an unbounded solution, a non-feasible solution, a numerically poor solution or because of a system error.
- T1(5,5): This matrix is comprised of the Risk Factor Values of Table 1. It is generated via a DATA statement.

- T6(3,2,2): This array contains the Mandatory Safety Requirements of Table 6. The first subscript identifies S_a , S_b , and S_c ; the second, age NEW or EXIST; the third, the floor. It is generated via a DATA statement.
- IT4X(13): This vector contains the current state indices for each PARAMETER of Table 4. It is read in the Fire Zone Input Block; it may be modified via CHANGE cards.
- HFORCE(53): This is a Hollerith vector; it contains the solution ID, as it appears in the solution prints, for each of the state variables. It is assembled character by character in various blocks of the program as follows:
- All blanks initially in System Initialization
- I&P appended in Fire Zone Input Block
- C&F appended in Control Modification Block
- * appended in the Alternate Solution Blocks
- S appended in PRSOL
- Blanks are partially restored within each of the Alternate Solution blocks and in the Restore Block
- It is not used in any calculations but for printing only.
- G(13,5): This is the Safety Value matrix of Table 4. It is created via a DATA statement and never changed. Values for row one and those values for which there are alternates dependent upon fire zone are irrelevant. Necessary modification is done on A instead.
- HPAR(21,5): This is a Hollerith matrix created via a DATA statement and never changed. Each row contains a five word (30 character) name for the corresponding row of A. It is printed in totality in the Simplex Matrix Report and in part in the System Initialization Report and the Solution Reports.
- HOUT(6): This is a Hollerith vector which is the problem name of six words (36 characters). It is read in the Fire Zone Input Block and used only for printing. It appears in the Fire Zone Report and all Solution Reports.

- C(53): This vector contains the state transition costs for each of the fifty-three Simplex variables. It is constructed in the Fire Zone input block, modified in the Control-Modification block, and printed in PRSOL.
- IT4S(13,2): This is the Table 4 to A linkage matrix. It is created via a DATA statement and used throughout the program. The rows correspond to the rows of Table 4; column 1 contains the index, within the A matrix variables corresponding to the first state within the row. Column 2 contains the A index for the last state (excluding the contrived ones) within the row.
- IST4 (60,2): This is the A matrix to Table 4 linkage matrix. It is created via a DATA statement. The rows correspond to the variable indices of A; columns 1 and 2 contain the row and column, respectively of Table 4 corresponding to the variable.
- JPR(13): This vector contains the indices of the prescriptive states for each of the rows of Table 4. It is created via a DATA statement.
- T4A(4,3,3): This is a safety value array corresponding to row 1 of Table 4. It is created via a data statement. The first dimension corresponds to FLOOR; the second, to construction type (WOOD FRAME, ORDINARY, or NON-COMBUSTIBLE); the third, to states.
- IT1X(5): This is the current state vector for Table 1. It is read in the Fire Zone input block. An illegal state index (i.e., less than 1 or greater than the maximum permissible) is defaulted to the index with the highest risk. It is used to calculate R and is printed in the Fire Zone Report.
- IT1M(5): This vector contains the maximum length for each of the rows of Table 1. It is created via a DATA statement.
- HT1(5,5): This matrix is Hollerith; it contains the Table 1 line headings for the System Status Report. It is created via a DATA statement.
- IQ(6,6): This is a temporary matrix into which the element cost vectors are read in the Fire Zone Input block. It is not used in any other part of the program.

- LV(13): This vector contains the length of the element cost vector for each row of Table 4. It is created via a DATA statement.
- AGE(2): This is a Hollerith vector containing NEW and EXIST. It is constructed via a DATA statement and is used in printing the Fire Zone Input report.
- CON(3): This is a Hollerith vector corresponding to the three construction types of row 1 of Table 4. It contains UNPROT, PROT, and FRST. It is created via a DATA statement.
- Z(150): This vector holds the safety values of Table 4 corresponding to the A matrix indices of the corresponding states.
- CS(13,5,5): This is the unit cost array. The first two dimensions correspond to the rows and columns of Table 4. The third dimension corresponds to the indices of the element cost vector.

B.3 Definitions of Data Inputs

There are two classes of input required: the first is that which defines the fire zone status; the second, that which allows user control of the problem variants to be solved. The description of the input is FORTRAN oriented. The input medium is assumed to be an 80 column card. Each card is given a name suggestive of the data contained therein and is defined by the FORTRAN format and the pertinent portion of the READ statement.

Fire Zone Input

Card 1: Building Safety Factor Indices Card

(13I3, 6A6), [(IT4X(I), I=1, 13), (HOUT(I), I=1, 6)]

The first thirteen fields are the indices of the current states of the Building Safety Factor values of Table 4. For factor 1, CONSTRUCTION, this is the index within FLOOR and COMBUSTIBLE-WOOD FRAME, COMBUSTIBLE-ORDINARY, and NON-COMBUSTIBLE. The next six fields may be considered as thirty-six characters of free form text. They are used only as an identification printed in several of the reports.

Card 2: Occupancy Risk Parameter Factors Card

(5I13), [(IT1X(I), I=1, 5)]

These fields are the indices of the current states of the Occupancy Risk Parameter Factors of Table 1.

Card 3: Problem - Floor Card

(513), [IPROB, NAGE, NCON, IFL, IPA]

IPROB is problem number, an identification field

NAGE is the building age:

1 → New

2 → Existing

NCON is the construction type:

1 → Combustible-Wood Frame

2 → Combustible - Ordinary

3 → Non-Combustible

IFL is the floor

IPA is the number of patients.

Cards 4-16: Safety Parameter Input Values

(I3, 12I6), [J,IQ (J1,J2), J2=1, K1, J1=J3, J4)]

One or more of these cards appear for each of the rows of Table 4. J is the row index. The array IQ (6x6) is used for temporary storage of the element cost vector pending calculation of the state transition cost; the rows correspond to the possible states of the Jth row; the columns contain the element cost vectors. Prior to the read operation, all elements are set to -1, denoting an impossible or precluded retrofit. J3 is the index of the current state +1; J4 is the maximum index within the row; if J3 is greater than J4, the fields except for J are irrelevant. K1 is the length of the element cost vector. For J equal to 8, two cards may appear since the length of the element cost vector times the number of states may exceed the twelve fields allowed by the format. The trailer card is in exactly the same format with J repeated followed by the remaining cost element vectors. By Table 4 row, the cost element vector lengths are:

Row	1	2	3	4	5	6	7	8	9	10	11	12	13
Length	1	2	2	2	4	1	3	4	3	3	1	1	2

Control-Modification Card Input

(A6,2I3,I7) [I,I1, I2, I4]

These cards allow the user to modify the definition of the preceding fire zone and to control the variants of the problem to be solved. Each of these and its function are defined. A Δ denotes a blank space.

SOLVE Δ - other fields irrelevant. This card indicates that the fire zone currently resident is to be solved. That is, the complete family of retrofit solutions is to be obtained.

NEXT $\Delta\Delta$ - other fields irrelevant. This card indicates that the next fire zone is to be read. No further variants of the currently resident zone are desired.

END $\Delta\Delta\Delta$ - other fields irrelevant. This card indicates the end of the batch run.

CHANGE $\alpha \beta \gamma$ This card indicates that a cost change is to be performed on the resident fire zone. The field α is the index of the row (Building Safety Factor) of Table 4 to be changed; β , the index of the column (state). The change to be made is dependent upon γ as follows:

$\gamma < 0$ the cost is made arbitrarily high. The indicated state is not to be considered.

$\gamma = 0$ all costs in row α , except β , are made arbitrarily high. The indicated state is required to be in every solution and becomes the input state.

$\gamma > 0$ γ becomes the cost for row α , column β .

REQUIR $\alpha \beta$ - remaining fields irrelevant. The mandatory safety requirement denoted by α , where $\alpha = 1 \rightarrow S_a$, $\alpha = 2 \rightarrow S_b$, $\alpha = 3 \rightarrow S_c$, and $\alpha = 4 \rightarrow R$, is changed by the percentage β . For example, if $\alpha = 1$. A new $S_a^{(n)}$ is calculated as $S_a^{(n)} = (1 + \frac{\beta}{100}) \times S_a^{(o)}$.

B.4 Description of Reports Produced

The output is described in terms of the content (and interpretation) of the reports generated and the rules which determine the number and ordering of reports. Each report is a single page except where otherwise noted.

System Status Report

The first report is the System Status Report. It is produced only once for each batch of fire zones. It is a single page and is comprised of facsimiles of Tables 1, 6, and 4 as the tables have been constructed via DATA statements.

Fire Zone Input Report

The Fire Zone Input Report appears for each zone read. It is produced prior to the solution of the optimization problem and consists of a cosmetically enhanced version of the fire zone input, augmented by the mandatory requirements and the retrofit costs for all possible changes.

The first line contains the fire zone identification and Table 1 indices as they appeared on the input except defaulted to highest risk if input was illegal. The second line corresponds to input card 3 with the age of building and construction numeric codes replaced by text and the remaining numeric fields identified by name. The third line contains the mandatory safety requirements.

The tabular portion of the report contains, for each fire safety factor and under appropriate column headings, the input state index, the prescriptive state index, the element cost vector, (as input or a default value) and the cost (computed or a default value) associated with a transition from the current state to the indicated state. The default rules are: for the existing state, both the element cost vector and the cost are defaulted to zero; for all states with lesser values (lower index) than the current state and all states precluded via a negative element cost vectors are defaulted to arbitrarily large numbers which appear on the report as fields of asterisks(*).

The remainder of the report consists of a simplified version of the modification cards (if any) encountered before a SOLVE card initiates a print of the SOLVE line and the calculation of the family of optimum solutions.

For a CHANGE card the line printed is CHANGE Row α Column β Index γ cost $\delta_1, \dots, \delta_n$ where α and β are the row and column, respectively, modified by the CHANGE card, γ is the input state index and δ_1 through δ_n are the negatives of the costs associated with the modified row.

For a REQUIR card, the line printed is the index of the requirement which was changed, the percentage by which it was changed and the new requirement.

Simplex Matrix Report

Following each SOLVE line (whether in a Fire Zone Input Report or a Modified Problem Report) is a Simplex Matrix Report. For the first

solution of the fire zone, this is a multi-page report containing the matrix used by the optimization algorithm. The row headings indicate the source of the equation (1-13 from Table 4 rows, 14-16 from the footnoted interdependencies, 17-20 from the mandatory requirements, and 21 the objective function), and the column headings indicate the variable index correspondence between the optimization matrix column and the row-column of Table 4. For modifications of the fire zone, this report is condensed into a single page report which contains only the last five rows of the optimization matrix. The condensed version contains all elements of the matrix subject to modification via the CHANGE and REQUIR cards.

Modified Problem Report

The Modified Problem Report is produced for each set of CHANGE and/or REQUIR cards following a SOLVE. That is, for every modified fire zone, the report is produced, unless the problem as input is not solved.¹ It consists of a one line print of MODIFIED PROBLEM α , where α is the problem identification number, followed by the simplified version of the pertinent CHANGE and REQUIR cards exactly as in the Fire Zone Input Report and followed by a SOLVE.

Solution Reports

Following each SOLVE line, regardless of the report in which it appears, is the Bounding Solution Report. Following this report are two sets of Alternate Solution Reports. All three solution reports are nearly identical; thus the descriptions will be combined with the differences noted.

The first line of each report contains the problem number, problem name, and an ALTERNATE index which is interpreted as follows:

- 0 → Bounding Solution-Input
- 1 → Bounding Solution Modified
- 10 $\alpha\beta$ → first set of alternates; $\alpha\beta$ as a two digit number is the index of the row of Table 4 which is varied.
- 2 $\alpha\beta\gamma$ → Second set of alternates; $\alpha\beta$ is the row; γ is the column of the cost varied.

The five line tableau(s) which may appear on the right hand side contain(s) information as follows:

¹ In this case, the report (minus the first line) actually appears as a coda to the fire zone input report.

TOTAL COST appears for every solution. It gives the total cost for the solution and indicates the surplus over each of the mandatory safety requirements attained by the solution.

PRESCRIPTIVE appears only for the Bounding Solution Report. The first line gives the cost of the prescriptive retrofits. Asterisks (*) indicate that one or more of the prescriptive indices is precluded. The remaining lines show the surplus, if any, over each of the requirements. Surpluses can only appear in the General Safety Requirement for the prescriptive solution.

The left hand portion of the report parallels Table 4. For each of the rows, the states appearing on input or in the solution generates a print line. For the Bounding Solution Report, the prescriptive solution generates a print line; for the modified problems, the state corresponding to a cost modification generates a print line; and for the alternates, each cost changed generates a print line. Each of the lines is identical in format except for the row heading on the first line printed from each row. Under the heading VARIABLE are the index (within the optimization matrix) and the value of the variable in the solution. Under the heading TABLE 4 is the column index of the state. Under COST is the total transition cost (from input to the indicated state). Asterisks (*) indicate a precluded state (i.e., an arbitrarily high cost). The GRADE is the Table 4 safety value associated with the state. The remaining information on the line shows the reason the line was included in the print. It will always consist of some combination of S, I, P, *, F, or C. These terms are defined as:

S → state is in solution

I → input state

P → prescriptive state

* → cost modified by alternate logic

F → state forced into solution via a CHANGE

C → state precluded by a CHANGE or cost input by a CHANGE

There are two classes of alternate solutions produced. They were selected to mitigate the impact of non-integer solutions and to give the user an automatic set of alternatives, thus enabling him to accommodate externalities, as well as answer some "what if" questions expected to be common to many users.

The first class of alternates precludes any retrofit within a row of Table 4 considering the rows in turn and one at a time. The number of solutions in this class will always be thirteen minus the number of rows for which no retrofit is allowed. However, one or more of these

may duplicate the bounding solution; this will occur for every row for which the bounding solution contained the existing state.

The second class of solutions is obtained by changing, in turn, the cost associated with each of the fifty-three variables. If the variable is in the bounding solution, its cost is adjusted so that it is precluded from an alternate solution. If the variable is not in the original solution; the costs of the other variables within the TABLE 4 row are adjusted so that the variable is forced into the optimal solution. The number of alternates in this class is exactly the number of allowable states with a maximum of fifty-three.

Some of the cost perturbations may result in non-feasible solutions (i.e., there is no acceptable way to attain compliance). This fact, the fact that alternatives may include duplicates, and the fact that the number of alternates generated is dependent upon both the input states and the retrofits precluded, means that there is no a priori way to get a good estimate for the number of unique solutions presented.

APPENDIX C

Sample Computer Runs

This appendix contains three computer output products: 1) a compilation of each of the programs used; 2) an annotated sample input listing; and 3) the abbreviated set of reports produced by processing the input shown. The program compilations are the versions used to produce the sample output shown in Chapter 2. All of the subroutines are exactly as they were for the case application presented in Chapter 2 and as described in Appendix B.

The Program MAIN has been changed for the example presented in this section in order to limit the number of alternative solutions. The first class¹ of alternative solutions has been limited to two, the first two which would normally appear.

The second class² of alternative solutions has been limited to three. Each of these changes involves only the changing of a limit on a DO loop. The only other deviation from the program description given in Appendix B consists of a patch executed after a NEXT card and prior to the fire zone input. This patch allows the CS array³ to be changed between fire zone inputs.

The sample input is identical to that for the third floor fire zone of the case application presented in Chapter 2. In addition, it consists

¹ The first class of alternatives is determined by the rows of Table 4. For example, suppose X_{ij} appears in the optimal solution as a retrofit, then the costs of all potential retrofits for the i^{th} feature are made arbitrarily high. This step guarantees that no potential retrofits associated with that feature can appear in the optimal solution. The number of solutions generated in this class is equal to the number of features which moved to a higher state in the optimal solution.

² This class of solutions is generated in the following manner: if X_{ij} appears in the optimal solution as a retrofit then its cost is made arbitrarily high thus precluding it from consideration; or if X_{ij} did not appear in the optimal solution as a retrofit and it is a potential retrofit, then its cost is held constant and the costs of all other potential retrofits for that feature are made arbitrarily high. The number of solutions generated in the second class is equal to the number of variables in the original linear programming problem minus the number of retrofit states which cannot be attained.

³ CS is a 13 by 5 by 5 array containing the unit costs used in the linear programming problem. The first two dimensions correspond to the rows and columns of Table 4. The third dimension corresponds to the indices of the element cost vector.

of a set of modifications carefully chosen so as to illustrate the use of the CHANGE and REQUIR options. Exhibit C.1 shows the input stream as it actually entered the computer.

Immediately following Exhibit C.1 are the report sets produced by the input stream. The number of solutions presented was limited so as to keep the example reasonably sized. Since Section 2.4.2 dealt with the output format and its interpretation, an extensive interpretation of the results will not be given. The intent of the example is to illustrate: 1) the flexibility of the system; 2) the inputs required; and 3) the form, content, and sequence of the reports produced. It is not intended to provide a line by line description of the input-output stream.

Exhibit C.1 Sample Input Run Stream

```

3 1 1 2 1 2 3 1 1 2 2 1 1 FLOOR 3
3 4 2 3 2
3 2 3 3 32
1
2 2000 0 2000 0
3 0 1200 0 1200
4 0 3 -1 -1
5 0 0 30 0 30 0 0 0 30 0 0 30
6 1 1 -1
7 0 0 3 0 0 3
8 -1 -1 -1 320 0 0 0 -1 -1 -1 -1 -1
8 1090 240 0 2
9 0 1 -1 -1 -1 -1
10 0 25 1 -1 -1 -1
11 1
12 2000 5610 7610 8700
13 2000 0 7610 0 7610 1090
SOLVE
CHANGE 9 2 0
CHANGE 11 3 1200
CHANGE 5 2 -1
REQUIR 2 20
SOLVE
END

```

Exhibit C.2 Sample Output Run Stream

Table 1. Occupancy Risk Factors

1. Patient Mobility	1.0	1.6	3.2	4.5	
2. Patient Density	1.0	1.2	1.5	2.0	
3. Zone Location	1.1	1.2	1.4	1.6	
4. Ratio of P to A	1.0	1.1	1.2	1.5	4.0
5. Average Age	1.0	1.2			

Table 6. Mandatory Safety Requirements

Zone Location	Containment SA		Extinguishment SB		People Movement SC	
	New	Exist	New	Exist	New	Exist
First Floor	9.0	4.0	6.0	3.0	6.0	1.0
Above 1st Floor	14.0	8.0	8.0	5.0	9.0	3.0

Table 4. Parameter Values with Prescriptive Indices

1. Construction	3							
First Floor		-2.0	.0	-2.0	.0	.0	2.0	2.0
Second Floor		-7.0	-2.0	-4.0	-2.0	-2.0	2.0	4.0
Third Floor		-9.0	-7.0	-9.0	-7.0	-7.0	2.0	4.0
4th & Above		-13.0	-7.0	-13.0	-7.0	-9.0	-7.0	4.0
2. Interior Finish Corr & Exit	2	-5.0	.0	3.0				
3. Interior Finish Rooms	2	-3.0	1.0	3.0				
4. Corridors Partitions/Walls	2	-10.0	.0	1.0	2.0			
5. Doors to Corridor	3	-10.0	.0	1.0	2.0			
6. Zone Dimensions	4	-6.0	-4.0	-2.0	.0	1.0		
7. Vertical Openings	4	-14.0	-10.0	.0	2.0	3.0		
8. Hazardous Areas	5	-11.0	-5.0	-6.0	-2.0	.0		
9. Smoke Control	2	-2.0	.0	3.0	4.0			
10. Emergency Movement Routes	3	-8.0	-2.0	.0	3.0	5.0		
11. Manual Fire Alarm	2	-4.0	1.0	2.0				
12. Smoke Detection & Alarm	1	.0	2.0	3.0	4.0	5.0		
13. Automatic Sprinklers	1	.0	2.0	8.0	10.0			

Exhibit C.2 (Cont.)

Problem 3 Floor 3 Table 1 Indices 3 4 2 3 2

Problem 3 EXIST F FSNT Floor 3 Patients 32

SA = 8.0 SB = 5.0 SC = 3.0 R = 6.6

	I	P	Input ^a	Cost ^b	Input	Cost	Input	Cost	Input	Cost	Input	Cost
1. Construction	3	3	*****	*****	*****	*****	0	0				
2. Interior Finish Corr & Exit	1	2	0	0	2000	800	2000	800				
			0		0		0					
3. Interior Finish Rooms	1	2	0	0	0	600	0	600				
			0		1200		1200					
4. Corridors Partitions/Walls	2	2	*****	*****	0	0	0	1200	*****	*****		
			*****		0		3		*****			
5. Doors to Corridor	1	3	0	0	0	3000	30	9000	30	10800		
			0		0		0		0			
			0		30		0		0			
			0		0		0		30			
6. Zone Dimensions	2	4	*****	*****	0	0	1	15000	1	19650	*****	*****
7. Vertical Openings	3	4	*****	*****	*****	*****	0	0	0	3750	0	3900
			*****		*****		0		0		0	
			*****		*****		0		3		3	
8. Hazardous Areas	1	5	0	0	*****	*****	320	720	*****	*****	1090	7453
			0		*****		0		*****		240	
			0		*****		0		*****		0	
			0		*****		0		*****		2	
9. Smoke Control	1	2	0	0	0	1400	*****	*****	*****	*****		
			0		1		*****		*****			
10. Emergency Movement Routes	3	3	*****	*****	*****	*****	0	0	0	4650	*****	*****
			*****		*****		0		25		*****	
			*****		*****		0		1		*****	
11. Manual Fire Alarm	2	2	*****	*****	0	0	1	1000				
12. Smoke Detection & Alarm	1	1	0	0	2000	4500	5510	12623	7610	17123	8700	19375
13. Automatic Sprinklers	1	1	0	0	2000	4000	7610	15220	7610	17673		
					0		0		1090			

SOLVE

^a Numbers in this column are actual element counts.

^b A cost of 0 implies the Building Safety Feature is already in that state.
 A cost greater than or equal to 0 is an estimate of the retrofit cost.
 A cost of ***** indicates a regression in score or impossible retrofit.

TABLE 1. OCCUPANCY RISK FACTORS

1. PATIENT MOBILITY	1.0	1.6	3.2	4.5	
2. PATIENT DENSITY	1.0	1.2	1.5	2.0	
3. ZONE LOCATION	1.1	1.2	1.4	1.6	1.6
4. RATIO OF P TO A	1.0	1.1	1.2	1.5	4.0
5. AVERAGE AGE	1.0	1.2			

TABLE 6. MANDATORY SAFETY REQUIREMENTS

ZONE LOCATION	CONTAINMENT SA		EXTINGUISHMENT SB		PEOPLE MOVEMENT SC	
	NEW EXIST	NEW EXIST	NEW EXIST	NEW EXIST	NEW EXIST	NEW EXIST
FIRST FLOOR	9.0	4.0	6.0	3.0	6.0	1.0
ABOVE 1ST FLOOR	14.0	8.0	8.0	5.0	5.0	3.0

TABLE 4. PARAMETER VALUES WITH PRESCRIPTIVE INDICES

1. CONSTRUCTION	3	3	3	3	3	3	3
FIRST FLOOR	-2.0	.0	-2.0	.0	-2.0	.0	2.0
SECOND FLOOR	-4.0	-2.0	-4.0	-4.0	-2.0	-2.0	4.0
THIRD FLOOR	-9.0	-7.0	-9.0	-9.0	-7.0	-7.0	4.0
4TH & ABOVE	-13.0	-7.0	-13.0	-13.0	-7.0	-9.0	4.0
INTERIOR FINISH CORR & EXIT	2	-5.0	.0	3.0			
INTERIOR FINISH ROOMS	2	-3.0	1.0	3.0			
CORRIDORS PARTITIONS/WALLS	2	-10.0	.0	1.0	2.0		
DOORS TO CORRIDOR	3	-10.0	.0	1.0	2.0		
ZONE DIMENSIONS	4	-6.0	-4.0	-2.0	.0	1.0	
VERTICAL OPENINGS	4	-14.0	-10.0	.0	2.0	3.0	
HAZARDOUS AREAS	5	-11.0	-5.0	-6.0	-2.0	.0	
SMOKE CONTROL	2	-2.0	.0	3.0	4.0		
EMERGENCY MOVEMENT ROUTES	3	-8.0	-2.0	.0	3.0	5.0	
MANUAL FIRE ALARM	2	-4.0	1.0	2.0			
SMOKE DETECTION & ALARM	1	.0	2.0	3.0	4.0	5.0	
AUTOMATIC SPRINKLERS	1	.0	2.0	8.0	10.0		

TABLE 1 INDICES 3 4 2 3 2

PROBLEM 3 FLOOR 3

PROBLEM 3 EXIST F FSNT FLOOR 3 PATIENTS 32

SA= 8.0 SB= 5.0 SC= 3.0 R= 6.6

	I	P	INPUT	COST	INPUT	COST	INPUT	COST	INPUT	COST	INPUT	COST
1. CONSTRUCTION	3	3	*****	*****	*****	*****	0	0				
2. INTERIOR FINISH CORR & EXIT	1	2	0	0	2000	800	2000	800				
3. INTERIOR FINISH ROOMS	1	2	0	0	1200	600	1200	600				
4. CORRIDORS PARTITIONS/WALLS	2	2	*****	*****	0	0	0	1200	*****	*****	*****	*****
5. DOORS TO CORRIDOR	1	3	0	0	0	3000	30	9000	30	10800		
6. ZONE DIMENSIONS	2	4	*****	*****	0	0	1	15000	1	15000	1	15000
7. VERTICAL OPENINGS	3	4	*****	*****	*****	*****	0	0	0	3750	0	3000
8. HAZARDOUS AREAS	1	5	0	0	*****	*****	320	720	*****	*****	1000	7053
9. SMOKE CONTROL	1	2	0	0	0	1400	*****	*****	*****	*****	*****	*****
10. EMERGENCY MOVEMENT ROUTES	3	3	*****	*****	*****	*****	0	0	0	4650	*****	*****
11. MANUAL FIRE ALARM	2	2	*****	*****	0	0	1	1000				
12. SMOKE DETECTION & ALARM	1	1	0	0	2000	4500	5510	12623	7610	17123	8700	19575
13. AUTOMATIC SPRINKLERS	1	1	0	0	2000	4000	7610	15220	7610	17673	1090	

SOLVE

PARAMETER	1		2		3		4		5		6		7		8		9		10						
	1-1	1-2	1-3	2-1	2-2	2-3	3-1	3-2	3-3	4-1	4-2	4-3	5-1	5-2	5-3	6-1	6-2	6-3	7-1	7-2	7-3	8-1	8-2	8-3	
1. CONSTRUCTION	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
2. INTERIOR FINISH CORR & EX	.0	.0	.0	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
3. INTERIOR FINISH ROOMS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
4. CORRIDORS PARTITIONS/WALLS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
5. DOORS TO CORRIDOR	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6. ZONE DIMENSIONS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
7. VERTICAL OPENINGS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
8. HAZARDOUS AREAS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
9. SMOKE CONTROL	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
10. EMERGENCY MOVEMENT ROUTES	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
11. MANUAL FIRE ALARM	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
12. SMOKE DETECTION & ALARM	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
13. AUTOMATIC SPPINKLERS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
14. FOW 4 ROW 5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
15. FOW 6 ROW 10	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
16. ROW 5 ROW 4	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
17. SA * CONTAINMENT	-7.0	2.0	4.0	-5.0	.0	3.0	-3.0	1.0	3.0	-10.0	1.0	3.0	-10.0	1.0	3.0	-10.0	1.0	3.0	-10.0	1.0	3.0	-10.0	1.0	3.0	-10.0
18. SB * EXTINGUISHMENT	-7.0	2.0	4.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
19. SC * PEOPLE MOVEMENT	.0	.0	.0	-5.0	.0	3.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
20. SG * GENERAL	-7.0	2.0	4.0	-5.0	.0	3.0	-3.0	1.0	3.0	-10.0	1.0	3.0	-10.0	1.0	3.0	-10.0	1.0	3.0	-10.0	1.0	3.0	-10.0	1.0	3.0	-10.0
21. COST ROW	-393000	-393000	0	0	-755	-755	0	0	-600	-393000	-600	-393000	-600	-393000	-600	-393000	-600	-393000	-600	-393000	-600	-393000	-600	-393000	-600

	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	4-2	4-3	4-4	5-1	5-2	5-3	5-4	6-1	6-2	6-3	6-4	6-5	7-1	7-2
1	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
2	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
3	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
4	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
5	.0	.0	.0	1.0	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	.0	.0	.0	.0	1.0	1.0	1.0	1.0	1.0	.0	.0
7	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	1.0	1.0
8	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
9	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
10	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
11	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
12	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
13	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
14	.0	.0	.0	-1.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
15	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
16	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
17	.0	1.0	2.0	-10.0	.0	1.0	2.0	.0	.0	.0	.0	.0	-14.0	-10.0
18	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
19	.0	.0	.0	-10.0	.0	1.0	2.0	-6.0	-4.0	-2.0	.0	1.0	-14.0	-10.0
20	.0	1.0	2.0	-10.0	.0	1.0	2.0	-6.0	-4.0	-2.0	.0	1.0	-14.0	-10.0
21	0	-1200	-393000	0	-3000	-9000	-10800	-393000	0	-15000	-19650	-393000	-393000	-393000

	25	26	27	28	29	30	31	32	33	34	35	36	37	38
	7-3	7-4	7-5	8-1	8-2	8-3	8-4	8-5	9-1	9-2	9-3	9-4	10-1	10-2
1	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
2	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
3	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
4	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
7	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
8	.0	.0	.0	1.0	1.0	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0
9	.0	.0	.0	.0	.0	.0	.0	.0	1.0	1.0	1.0	1.0	.0	.0
10	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	1.0	1.0
11	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
12	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
13	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
14	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
15	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
16	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
17	.0	2.0	3.0	-11.0	-5.0	-6.0	-2.0	.0	.0	.0	.0	.0	.0	.0
18	.0	.0	.0	-11.0	-5.0	-6.0	-2.0	.0	.0	.0	.0	.0	.0	.0
19	.0	2.0	3.0	.0	.0	.0	.0	.0	-2.0	.0	3.0	4.0	-8.0	-2.0
20	.0	2.0	3.0	-11.0	-5.0	-6.0	-2.0	.0	-2.0	.0	3.0	4.0	-8.0	-2.0
21	0	-3750	-3900	0	-393000	-720	-393000	-7452	0	-1400	-393000	-393000	-393000	-393000

	39	40	41	42	43	44	45	46	47	48	49	50	51	52
	10-3	10-4	10-5	11-1	11-2	11-3	12-1	12-2	12-3	12-4	12-5	13-1	13-2	13-3
1	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
2	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
3	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
4	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
7	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
8	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
9	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
10	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
11	.0	.0	.0	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0
12	.0	.0	.0	.0	.0	.0	1.0	1.0	1.0	1.0	1.0	.0	.0	.0
13	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	1.0	1.0	1.0
14	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
15	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
16	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
17	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
18	.0	.0	.0	-4.0	1.0	2.0	.0	2.0	3.0	4.0	5.0	.0	2.0	8.0
19	.0	3.0	5.0	.0	.0	.0	.0	2.0	3.0	4.0	5.0	.0	1.0	4.0
20	.0	3.0	5.0	-4.0	1.0	2.0	.0	2.0	3.0	4.0	5.0	.0	2.0	8.0
21	0	-4650	-393000	-393000	0	-1000	0	-4500	-12622	-17122	-19575	0	-4000	-15220

	53	54	55	56	57	58	59	60 =	RHS	ROW
1	13-4	4-E	6-6	5-5	0-0	0-0	0-0	0-0	1.0	1
2	.0	.0	.0	.0	.0	.0	.0	.0	1.0	2
3	.0	.0	.0	.0	.0	.0	.0	.0	1.0	3
4	.0	1.0	.0	.0	.0	.0	.0	.0	1.0	4
5	.0	.0	.0	.0	.0	.0	.0	.0	1.0	5
6	.0	.0	.0	.0	.0	.0	.0	.0	1.0	6
7	.0	.0	.0	.0	.0	.0	.0	.0	1.0	7
8	.0	.0	.0	.0	.0	.0	.0	.0	1.0	8
9	.0	.0	.0	.0	.0	.0	.0	.0	1.0	9
10	.0	.0	.0	.0	.0	.0	.0	.0	1.0	10
11	.0	.0	.0	.0	.0	.0	.0	.0	1.0	11
12	.0	.0	.0	.0	.0	.0	.0	.0	1.0	12
13	1.0	.0	.0	.0	.0	.0	.0	.0	1.0	13
14	.0	1.0	.0	.0	.0	.0	.0	.0	.0	14
15	.0	.0	1.0	.0	.0	.0	.0	.0	1.0	15
16	.0	.0	.0	1.0	.0	.0	.0	.0	1.0	16
17	10.0	.0	.0	.0	-1.0	.0	.0	.0	8.0	17
18	10.0	.0	.0	.0	.0	-1.0	.0	.0	5.0	18
19	5.0	.0	.0	.0	.0	.0	-1.0	.0	3.0	19
20	10.0	.0	.0	.0	.0	.0	.0	-1.0	6.6	20
21	-17672	0	0	0	0	0	0	0	0	21

PARAMETER	PROBLEM 3	FLOOR 3	ALTERNATE 0	VARIABLE TABLE 4		GRADE	COST	S	I	P	TOTAL COST	SURPLUS	
				3	4								
1. CONSTRUCTION		3	1,000	3		4.00	.00				17858.22		
2. INTERIOR FINISH CORR & EXIT		4	.000	1		-5.00	.00					SA= 8.0	
		5	.000	2		.00	800.00					SURPLUS= -.0	
		6	1,000	3		3.00	800.00					SB= 5.0	
3. INTERIOR FINISH ROOMS		7	.500	1		-3.00	.00					SURPLUS= -.0	
		8	.000	2		1.00	600.00					SC= 3.0	
		9	.500	3		3.00	600.00					SURPLUS= -.0	
4. CORRIDORS PARTITIONS/WALLS		11	1,000	2		.00	.00					SG= 6.6	
5. DOORS TO CORRIDOR		14	.000	1		-10.00	.00						
		15	1,000	2		.00	3000.00					PFSCRIPTIVE 42652.50	
		16	.000	3		1.00	5000.00					SA= 8.0	
6. ZONE DIMENSIONS		19	1,000	2		-4.00	.00					SURPLUS= .0	
		21	.000	4		.00	19650.00					SB= 5.0	
7. VERTICAL OPENINGS		25	.000	3		.00	.00					SC= 3.0	
		26	.000	4		2.00	3750.00					SURPLUS= .0	
		27	1,000	5		3.00	3900.00					SG= 6.6	
		28	.000	1		-11.00	.00						
		30	.333	3		-6.00	720.00						
8. HAZARDOUS AREAS		32	.667	5		.00	7452.50						
	9. SMOKE CONTROL		33	.000	1		-2.00	.00					
			34	1,000	2		.00	1400.00					
10. EMERGENCY MOVEMENT ROUTES		39	1,000	3		.00	.00						
	11. MANUAL FIRE ALARM		43	.000	2		.00	.00					
			44	1,000	3		2.00	1000.00					
12. SMOKE DETECTION & ALARM		45	.500	1		.00	.00						
		46	.500	2		2.00	4500.00						
13. AUTOMATIC SPRINKLERS		50	1,000	1		.00	.00						

PARAMETER	PROBLEM 3	FLOOR 3	ALTERNATE 1002				
			VARIABLE	TABLE 4	COST	GFADE	
1. CONSTRUCTION		3 1.000	3	.00	4.00	S I	TOTAL COST 33550.18 SA= 8.0 SURPLUS= -.0 SB= 5.0 SURPLUS= 4.5 SC= 3.0 SURPLUS= -.0 SG= 6.6 SURPLUS= 2.9
2. INTERIOR FINISH CORR & EXIT		4 1.000	1	.00	-5.00	S I	
		5 .000	2	800.00	.00	*	
		6 .000	3	800.00	3.00	*	
3. INTERIOR FINISH ROOMS		7 .000	1	.00	-3.00	I	
		9 1.000	3	600.00	3.00	S	
4. CORRIDOFS PARTITIONS/WALLS		11 1.000	2	.00	.00	S I	
5. DOORS TO CORRIDOR		14 .000	1	.00	-10.00	I	
		15 1.000	2	3000.00	.00	S	
6. ZONE DIMENSIONS		19 1.000	2	.00	-4.00	S I	
7. VERTICAL OPENINGS		25 .000	3	.00	.00	I	
		27 1.000	5	3900.00	3.00	S	
8. HAZARDOUS AREAS		28 .000	1	.00	-11.00	I	
		30 1.000	3	720.00	-6.00	S	
9. SMOKE CONTROL		33 .000	1	.00	-2.00	I	
		34 1.000	2	1400.00	.00	S	
10. EMERGENCY MOVEMENT ROUTES		39 .000	3	.00	.00	I	
		40 1.000	4	4650.00	3.00	S	
11. MANUAL FIRE ALARM		43 1.000	2	.00	1.00	S I	
12. SMOKE DETECTION & ALARM		45 .250	1	.00	.00	S I	
		46 .750	2	4500.00	2.00	S	
13. AUTOMATIC SPRINKLERS		50 .100	1	.00	.00	S I	
		53 .900	4	17672.50	10.00	S	

ALTERNATE 2021

PROBLEM 3 FLOOR 3

PARAMETER

PARAMETER	VARIABLE	TABLE	COST	GRADE		TOTAL COST
1. CONSTRUCTION	3 1.000	3	.00	4.00	S I	33550.19
2. INTERIOR FINISH CURR & EXIT	4 1.000	1	.00	-5.00	* S I	SA= 3.0 SURPLUS= -1.0
3. INTERIOR FINISH ROOMS	7 .000	1	.00	-3.00	I	SB= 5.0 SURPLUS= 4.5
	9 1.000	3	600.00	3.00	S	SC= 3.0 SURPLUS= -1.0
4. CORRIDOR PARTITIONS/WALLS	11 1.000	2	.00	.00	S I	SG= 6.6 SURPLUS= 2.9
5. DOORS TO CORRIDOR	14 .000	1	.00	-10.00	I	
	15 1.000	2	3000.00	.00	S	
6. ZONE DIMENSIONS	19 1.000	2	.00	-4.00	S I	
7. VERTICAL OPENINGS	25 .000	3	.00	.00	I	
	27 1.000	5	3900.00	3.00	S	
8. HAZARDOUS AREAS	28 .000	1	.00	-11.00	I	
	30 1.000	3	720.00	-6.00	S	
9. SMOKE CONTROL	33 .000	1	.00	-2.00	I	
	34 1.000	2	1400.00	.00	S	
10. EMERGENCY MOVEMENT ROUTES	39 .000	3	.00	.00	I	
	40 1.000	4	4550.00	3.00	S	
11. MANUAL FIRE ALARM	43 1.000	2	.00	1.00	S I	
12. SMOKE DETECTION & ALARM	45 .250	1	.00	.00	S I	
	46 .750	2	4500.00	2.00	S	
13. AUTOMATIC SPRINKLERS	50 .100	1	.00	.00	S I	
	53 .500	4	17672.50	10.00	S	

ALTERNATE 2022

PROBLEM 3 FLOOR 3

PARAMETER	VARIABLE	TABLE 4	COST	GRADE	TOTAL COST	SURPLUS
1. CONSTRUCTION	3 1.000	3	.00	4.00	21966.87	
	4 .000	1	.00	-5.00		
	5 1.000	2	300.00	.00	SA= 8.0	SURPLUS= -.0
3. INTERIOR FINISH ROOMS	7 .000	1	.00	-3.00	SB= 5.0	SURPLUS= -.0
	9 1.000	3	600.00	3.00	SC= 3.0	SURPLUS= -.0
	11 1.000	2	.00	.00	SG= 6.6	SURPLUS= .4
5. DOORS TO CORRIDOR	14 .000	1	.00	-10.00		
	15 1.000	2	3000.00	.00		
6. ZONE DIMENSIONS	19 1.000	2	.00	-4.00		
	25 .000	3	.00	.00		
7. VERTICAL JPLINGS	27 1.000	5	300.00	3.00		
	28 .000	1	.00	-11.00		
8. HAZARDOUS AREAS	30 1.000	3	720.00	-6.00		
	33 .000	1	.00	-2.00		
9. SMOKE CONTROL	34 1.000	2	1400.00	.00		
	35 1.000	3	.00	.00		
11. MANUAL FIRE ALARM	43 1.000	2	.00	1.00		
	45 .000	1	.00	.00		
12. SMOKE DETECTION & ALARM	46 1.000	2	4500.00	2.00		
	50 .000	1	.00	.00		
13. AUTOMATIC SPRINKLERS	53 .400	4	17472.50	10.00		

MODIFIED PROBLEM	3	FLOOR 3			
CHANGE ROW 9 STATE 2 (2)			-393000.00	-1400.00	-393000.00 -353000.00
CHANGE ROW 11 STATE 3 (2)			-393000.00	.00	-1200.00
CHANGE ROW 5 STATE 2 (1)			.00	-393000.00	-5000.00 -10800.00
FEQUIR 2	20.00				6.00
SOLVE					

PARAMETER TABLE 4 INDICES	1		2		3		4		5		6		7		8		9		10		
	1-1	1-2	1-3	2-1	2-2	2-3	3-1	3-2	3-3	4-1	4-2	4-3	5-1	5-2	5-3	6-1	6-2	6-3	7-1	7-2	7-3
1. CONSTRUCTION	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
2. INTERIOR FINISH CORP & EX	.0	.0	.0	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
3. INTERIOR FINISH ROOMS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
4. CORRIDORS PARTITIONS/WALLS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
5. DOORS TO CORRIDOR	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6. ZONE DIMENSIONS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
7. VEPTICAL OPENINGS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
8. HAZARDOUS AREAS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
9. SMOKE CONTROL	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
10. EMERGENCY MOVEMENT ROUTES	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
11. MANUAL FIRE ALARM	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
12. SMOKE DETECTION & ALARM	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
13. AUTOMATIC SPRINKLERS	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
14. ROW 4 ROW 5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
15. ROW 6 ROW 10	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
16. ROW 5 ROW 4	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
17. SA * CONTAINMENT	-7.0	2.0	4.0	-5.0	.0	3.0	-3.0	1.0	3.0	-10.0	3.0	-3.0	1.0	3.0	-10.0	3.0	-10.0	3.0	-10.0	3.0	-10.0
18. SB * EXTINGUISHMENT	-7.0	2.0	4.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
19. SC * PEOPLE MOVEMENT	.0	.0	.0	-5.0	.0	3.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
20. SG * GENERAL	-7.0	2.0	4.0	-5.0	.0	3.0	-3.0	1.0	3.0	-10.0	3.0	-3.0	1.0	3.0	-10.0	3.0	-10.0	3.0	-10.0	3.0	-10.0
21. COST ROW	-393000	-393000	0	0	-759	-759	0	0	-600	-393000	-600	0	-600	-393000	-600	-393000	-600	-393000	-600	-393000	-600

	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	4-2	4-3	4-4	5-1	5-2	5-3	5-4	6-1	6-2	6-3	6-4	6-5	7-1	7-2
1	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
2	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
3	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
4	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
5	.0	.0	.0	1.0	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	.0	.0	.0	.0	1.0	1.0	1.0	1.0	1.0	.0	.0
7	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	1.0	1.0
8	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
9	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
10	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
11	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
12	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
13	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
14	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
15	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
16	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
17	.0	1.0	2.0	-10.0	.0	1.0	2.0	.0	.0	.0	.0	.0	-14.0	-10.0
18	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
19	.0	.0	.0	-10.0	.0	1.0	2.0	-6.0	-4.0	-2.0	.0	1.0	-14.0	-10.0
20	.0	1.0	2.0	-10.0	.0	1.0	2.0	-6.0	-4.0	-2.0	.0	1.0	-14.0	-10.0
21	0	-1200	-393000	0	-393000	-9000	-10800	-393000	0	-15000	-19650	-393000	-393000	-393000

	25	26	27	28	29	30	31	32	33	34	35	36	37	38
	7-3	7-4	7-5	8-1	8-2	8-3	8-4	8-5	9-1	9-2	9-3	9-4	10-1	10-2
1	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
2	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
3	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
4	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
7	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
8	.0	.0	.0	1.0	1.0	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0
9	.0	.0	.0	.0	.0	.0	.0	.0	1.0	1.0	1.0	1.0	.0	.0
10	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	1.0	1.0
11	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
12	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
13	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
14	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
15	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
16	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
17	.0	2.0	3.0	-11.0	-5.0	-6.0	-2.0	.0	.0	.0	.0	.0	.0	.0
18	.0	.0	.0	-11.0	-5.0	-6.0	-2.0	.0	.0	.0	.0	.0	.0	.0
19	.0	2.0	3.0	.0	.0	.0	.0	.0	-2.0	.0	3.0	4.0	-8.0	-2.0
20	.0	2.0	3.0	-11.0	-5.0	-6.0	-2.0	.0	-2.0	.0	3.0	4.0	-8.0	-2.0
21	0	-3750	-3900	0	-393000	-720	-393000	-7452	-393000	-1400	-393000	-393000	-393000	-393000

	39	40	41	42	43	44	45	46	47	48	49	50	51	52
	10-3	10-4	10-5	11-1	11-2	11-3	12-1	12-2	12-3	12-4	12-5	13-1	13-2	13-3
1	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
2	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
3	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
4	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
5	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
6	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
7	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
8	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
9	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
10	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
11	.0	.0	.0	1.0	1.0	1.0	.0	.0	.0	.0	.0	.0	.0	.0
12	.0	.0	.0	.0	.0	.0	1.0	1.0	1.0	1.0	1.0	.0	.0	.0
13	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	1.0	1.0	1.0
14	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
15	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
16	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
17	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
18	.0	.0	.0	-4.0	1.0	2.0	.0	2.0	3.0	4.0	5.0	.0	2.0	8.0
19	.0	3.0	5.0	.0	.0	.0	.0	2.0	3.0	4.0	5.0	.0	1.0	4.0
20	.0	3.0	5.0	-4.0	1.0	2.0	.0	2.0	3.0	4.0	5.0	.0	2.0	8.0
21	0	-4650	-393000	-393000	0	-1200	0	-4500	-12622	-17122	-19575	0	-4000	-11220

	53	54	55	56	57	58	59	60 =	RHS	POW
1	13-4	4-5	6-6	5-5	0-0	0-0	0-0	0-0	1.0	1
2	.0	.0	.0	.0	.0	.0	.0	.0	1.0	2
3	.0	.0	.0	.0	.0	.0	.0	.0	1.0	3
4	.0	1.0	.0	.0	.0	.0	.0	.0	1.0	4
5	.0	.0	.0	.0	.0	.0	.0	.0	1.0	5
6	.0	.0	.0	.0	.0	.0	.0	.0	1.0	6
7	.0	.0	.0	.0	.0	.0	.0	.0	1.0	7
8	.0	.0	.0	.0	.0	.0	.0	.0	1.0	8
9	.0	.0	.0	.0	.0	.0	.0	.0	1.0	9
10	.0	.0	.0	.0	.0	.0	.0	.0	1.0	10
11	.0	.0	.0	.0	.0	.0	.0	.0	1.0	11
12	.0	.0	.0	.0	.0	.0	.0	.0	1.0	12
13	1.0	.0	.0	.0	.0	.0	.0	.0	1.0	13
14	.0	1.0	.0	.0	.0	.0	.0	.0	.0	14
15	.0	.0	1.0	.0	.0	.0	.0	.0	1.0	15
16	.0	.0	.0	1.0	.0	.0	.0	.0	1.0	16
17	10.0	.0	.0	.0	-1.0	.0	.0	.0	8.0	17
18	10.0	.0	.0	.0	.0	-1.0	.0	.0	6.0	18
19	5.0	.0	.0	.0	.0	.0	-1.0	.0	3.0	19
20	10.0	.0	.0	.0	.0	.0	.0	-1.0	6.6	20
21	-17672	0	0	0	0	0	0	0	0	21

ALTERNATE 1

PROBLEM 3 FLOOR 3

PARAMETERS

PARAMETERS	VARIABLE	TABLE 4	COST	GRADE	S I P	TOTAL COST
1. CONSTRUCTION	3 1.000	3	.00	4.00	S IP	23619.83
2. INTERIOR FINISH CORR & EXIT	4 .000	1	.00	-5.00	I	SA= 8.0 SURPLUS= -.0
	5 .000	2	800.00	.00	P	
	6 1.000	3	800.00	3.00	S	
3. INTERIOR FINISH ROOMS	7 .621	1	.00	-3.00	S I	SB= 5.0 SURPLUS= -.0
	8 .000	2	600.00	1.00	P	
	9 .379	3	600.00	3.00	S	
4. COFFIDORS PARTITIONS/WALLS	11 1.000	2	.00	.00	S IP	SG= 6.6 SURPLUS= -.0
5. DOORS TO COFFIDOR	14 .000	1	.00	-10.00	I	PRESCRIPTIVE 42652.50
	15 .000	2	*****	.00	C	
	16 1.000	3	9000.00	1.00	S P	
6. ZONE DIMENSIONS	19 1.000	2	.00	-4.00	S I	SA= 9.0 SURPLUS= .0
	21 .000	4	19650.00	.00	P	SB= 6.0 SURPLUS= -1.0
7. VERTICAL OPENINGS	25 .545	3	.00	.00	S I	SC= 3.0 SURPLUS= .0
	26 .000	4	3750.00	2.00	P	
	27 .455	5	3900.00	3.00	S	
8. HAZARDOUS AREAS	28 .000	1	.00	-11.00	I	SG= 6.6 SURPLUS= 2.4
	30 .106	3	720.00	-6.00	S	
	32 .894	5	7452.50	.00	S P	
9. SMOKE CONTROL	33 .000	1	*****	-2.00	I	FS P
	34 1.000	2	1400.00	.00	P	
10. EMERGENCY MOVEMENT ROUTES	39 1.000	3	.00	.00	S IF	
11. MANUAL FIRE ALARM	43 1.000	2	.00	1.00	S IP	C
	44 .000	3	1200.00	2.00	C	
	45 .182	1	.00	.00	S IP	
12. SMOKE DETECTION & ALARM	46 .818	2	4500.00	2.00	S	
	50 1.000	1	.00	.00	S IP	
13. AUTOMATIC SPRINKLERS						

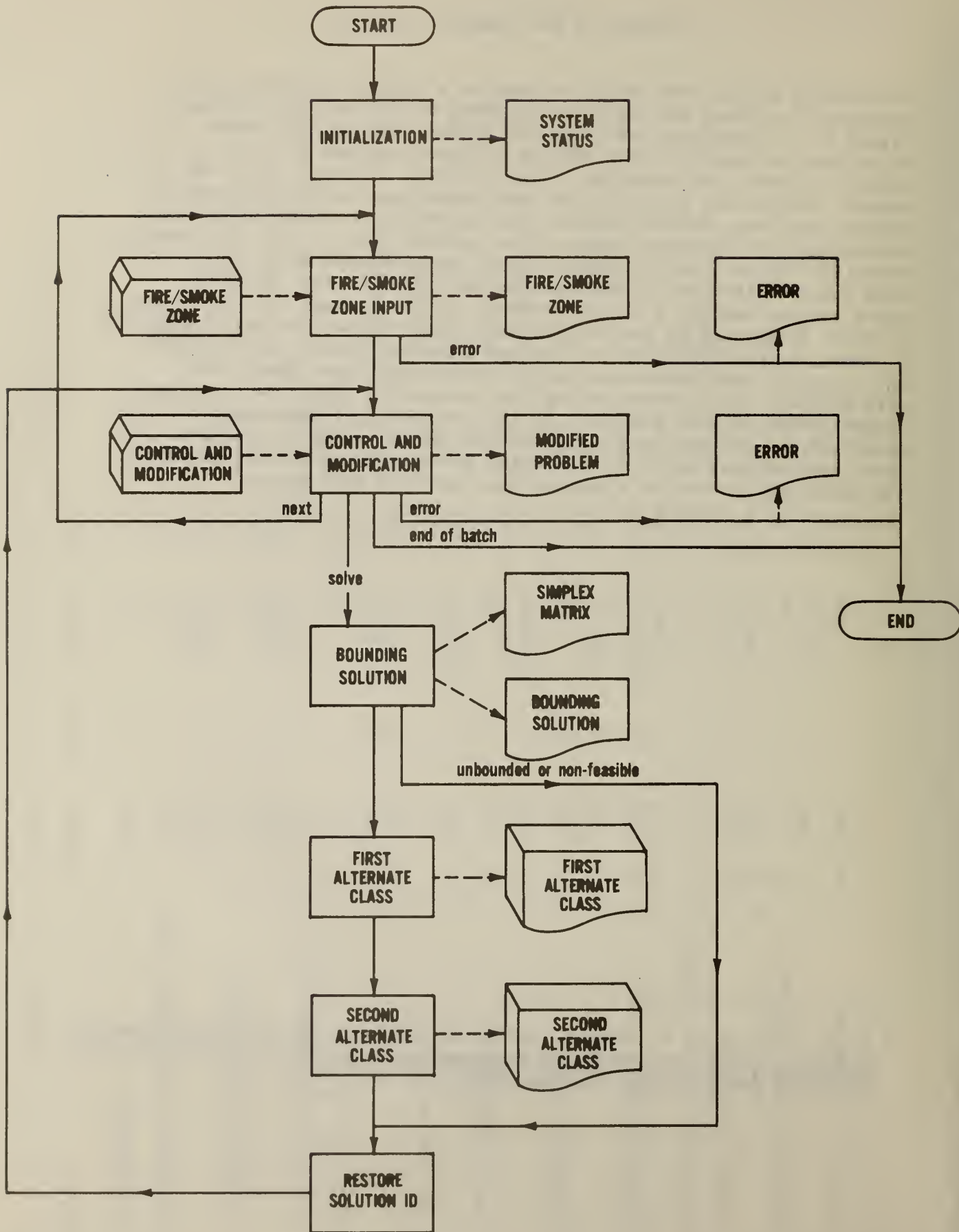
APPENDIX D

Listing of the Computer Programs

The purpose of this section is to provide a complete listing of all programs (including subroutines) contained in the software package. Figure D.1, a flowchart outlining the inputs, outputs and inter-relationships among programs and subroutines, has been provided to promote a better understanding of the software package. Since the number, length, and complexity of the various portions of the software package varies considerably, readers are encouraged to familiarize themselves with the flowchart prior to any serious study of the programming details of the software package. Since the MAIN program is considerably more involved than any of the subroutines listed in this appendix, those readers seeking a fuller knowledge of the package are encouraged to refer frequently to the MAIN program documentation section of Appendix B. Most of the subroutines in the remaining sections of this appendix can be easily read and understood by users having some familiarity with FORTRAN. It is worth noting that subroutine RVSM PX was developed independently of this project. It is a mathematical programming package which may be used to solve a wide variety of optimization problems. Those readers interested in a detailed description of subroutine RVSM PX are encouraged to obtain a copy of the report by Hall et al.¹

¹ W. G. Hall, R. H. F. Jackson, and P. B. Saunders, The National Bureau of Standards Linear and Quadratic Programming Subroutines, National Bureau of Standards Report 10695, February 1972.

FIGURE D.1 FLOWCHART OF COMPUTER PROGRAM



D.1 Listing of MAIN Program

FOR, IS LIFES.MAIN, LIFES.MAIN
FOR S 4F1 E -05/02/79-17:22:15 (.0)

MAIN PROGRAM

1*	INTEGER BLANK		
2*	DIMENSION A	(21,61)	@ SIMPLEX MATRIX
3*	DIMENSION IT4X	(13)	@ CURRENT INDEX
4*	DIMENSION C	(53)	@ PROTOTYPE COST CECTOR
5*	DIMENSION IT4S	(13,2)	@ TABLE 4 TO SIMPLEX LINKAGE
6*	DIMENSION IST4	(60,2)	@ SIMPLEX TO TABLE 4 LINKAGE
7*	DIMENSION HFORCE	(60)	@ IN OR OUT FORCED INDICATOR
8*	DIMENSION G	(13,5)	@ GRADES
9*	DIMENSION HPAR	(21,5)	@ HOLLERITH PARQMETER NAMES
10*	DIMENSION JPR	(13)	@ PRESCRIPTIVE FIX VECTOR
11*	DIMENSION X	(150)	@ SIMPLEX SOLUTION VECTOR
12*	DIMENSION Y	(150)	@ PROTOTYPE SOLUTION VECTOR
13*	DIMENSION Z	(100)	
14*	DIMENSION B	(23,23)	@ SIMPLEX INVERSE
15*	DIMENSION L	(153)	@ SIMPLEX GENERAL PURPOSE VECTOR

```

16* DIMENSION HOUT(6)
17* DIMENSION CS(13,5,5) @ ST COSTS * PAFA,COLUMN,VECTOR LENGTH
18* DIMENSION T6(3,2,2) @ TABLE 6 * FACTOR, AGE, FLOOR
19* DIMENSION T4A(4,3,3) @ TABL 4 * ROW 1 $ FLOOR, PROT, INDEX
20* DIMENSION IT1X(5) @ TABLE 1 INPUT INDICES
21* DIMENSION IT1M(5) @ TABLE 1 IROW MAXIMA
22* DIMENSION HT1(5,5) @ TABLE 1 LINE HEADS
23* DIMENSION T1(5,5) @ TABLE 1
24* DIMENSION IQ(6,6) @ UNIT INPUT * COLUMN,VECTOR LENGTH
25* DIMENSION LV(13) @ VECTOR LENGTH BY PARAVETER
26* DIMENSION AGE(2)
27* DIMENSION CON(3)
28* DATA ((IT4S(I,J),I=1,13),J=1,2) /
29* 1 1, 4, 7, 10, 14, 18, 23, 28, 33, 37, 42, 45, 50,
30* 2 3, 6, 9, 13, 17, 22, 27, 32, 36, 41, 44, 49, 53/
31* DATA ((IST4( I,J),J=1,2),I=1,56) /
32* 1 1, 1, 1, 2, 1, 3,
33* 2 2, 1, 2, 2, 2, 3,
34* 3 3, 1, 3, 2, 3, 3,
35* 4 4, 1, 4, 2, 4, 3, 4, 4,
36* 5 5, 1, 5, 2, 5, 3, 5, 4,
37* 6 6, 1, 6, 2, 6, 3, 6, 4, 6, 5,
38* 7 7, 1, 7, 2, 7, 3, 7, 4, 7, 5,
39* 8 8, 1, 8, 2, 8, 3, 8, 4, 8, 5,
40* 9 9, 1, 9, 2, 9, 3, 9, 4,
41* A 10, 1, 10, 2, 10, 3, 10, 4, 10, 5,
42* B 11, 1, 11, 2, 11, 3,
43* C 12, 1, 12, 2, 12, 3, 12, 4, 12, 5,
44* D 13, 1, 13, 2, 13, 3, 13, 4,
45* E 4,5,6,6,5,5 /
46* DATA ((G(I,J),I=1,13),J=1,5) /
47* 1 0.0,-5.0,-3.0,-10.0,-10.0,-6.0,-14.0,-11.0,-2.0,-8.0,-4.0,-.0,.0,
48* 2 2.0,.0,1.0,.0,.0,-4.0,-10.0,-5.0,.0,-2.0, 1.0, 2.0, 2.0,
49* 3 2.0,3.0,3.0,1.0,1.0,-2.0,.0,-6.0,3.0,.0,2.0,3.0,8.0,
50* 4 .0,.0,.0,2.0,2.0,.0,2.0,-2.0,4.0,3.0,.0,4.0,10.0,
51* 5 .0,.0,0.0,0.0,1.0,3.0,.0,.0,5.0,.0,5.0,.0 /
52* DATA ((HPAR(I,J),J=1,5),I= 1, 9) /
53* 1 'CONSTR','UCTION',' ',' ',' ','
54* 2 'INTERI','OR FIN','ISH C','OFFR & ','EXIT ','
55* 3 'INTERI','OR FIN','ISH R','OOMS ',' ','
56* 4 'CORRID','ORS PA','RTITIO','NS/WAL','LS ','
57* 5 'DOORS ','TO COR','RIDOR ',' ',' ','
58* 6 'ZONE D','IMENSI','ONS ',' ',' ','
59* 7 'VERTIC','AL OPE','NINGS ',' ',' ','
60* 8 'HAZARD','OUS AR','EAS ',' ',' ','
61* 9 'SMOKE ','CONTRO','L ',' ',' ','/
62* DATA ((HPAR(I,J),J=1,5),I=10,18) /
63* A 'EMERGE','NCY MO','VEMENT',' ROUTE','S ','
64* B 'MANUAL',' FIRE ','ALARM ',' ',' ','
65* C 'SMOKE ','DETECT','ION & ','ALARM ',' ','
66* D 'AUTOMA','TIC SP','RINKLE','RS ',' ','
67* F 'ROW 4 ','ROW 5 ',' ',' ',' ','
68* G 'ROW 6 ','ROW 10',' ',' ',' ','
69* H 'ROW 5 ','ROW 4 ',' ',' ',' ','
70* I 'SA * C','ONTAIN','MENT ',' ',' ','
71* J 'SB * E','XTINGU','ISHMEN','T ',' ','/
72* DATA ((HPAR(I,J),J=1,5),I=19,21) /
73* K 'SC * P','EOPLE ','MOVEME','NT ',' ','

```

```

74* L 'SG * G', 'ENERAL', ' ' ' ' ' ' ' '
75* M 'COST R', 'OW ' ' ' ' ' ' ' '
76* DATA (((T4A(I,J,K),K=1,3),J=1,3),I=1,4) /
77* 1 -2.0, 0.0, 0.0, -2.0, 0.0, 0.0, 0.0, 2.0, 2.0,
78* 2 -7.0, -2.0, 0.0, -4.0, -2.0, 0.0, -2.0, 2.0, 4.0,
79* 3 -9.0, -7.0, 0.0, -9.0, -7.0, 0.0, -7.0, 2.0, 4.0,
80* 4 -13.0, -7.0, 0.0, -13.0, -7.0, 0.0, -9.0, -7.0, 4.0 /
81* DATA (((T6(I,J,K),J=1,2),I=1,3),K=1,2) /
82* 1 9.0, 4.0, 6.0, 3.0, 6.0, 1.0,
83* 2 14.0, 8.0, 8.0, 5.0, 9.0, 3.0 /
84* DATA ((T1(I,J),J=1,5),I=1,5) /
85* 1 1.0, 1.6, 3.2, 4.5, 9.9,
86* 2 1.0, 1.2, 1.5, 2.0, 9.9,
87* 3 1.1, 1.2, 1.4, 1.6, 1.6,
88* 4 1.0, 1.1, 1.2, 1.5, 4.0,
89* 5 1.0, 1.2, 9.9, 9.9, 9.9 /
90* DATA ITIM / 4,4,5,5,2 /
91* DATA JPR / 3,2,2,2,3,4,4,5,2,3,2,1,1 /
92* DATA ((HT1(I,J),J=1,4),I=1,5) /
93* 1 'PATIEN', 'T ', 'MOBILI', 'Y ',
94* 2 'PATIEN', 'T ', 'DENSIT', 'Y ',
95* 3 'ZONE ', ' ', 'LOCATI', 'ON',
96* 4 'RATIO ', 'OF', 'P TO A', ' ',
97* 5 'AVERAG', 'E ', 'AGE ', ' ', ' ' /
98* DATA AGE / 'NEW ', 'EXIST ' /
99* DATA CON / 'UNPROT', 'PROT ', 'F RSNT' /
100* DATA ((CS( 1,I,J),J=1,5),I=1,5) /
101* 1 1.0,1.0,1.0,1.0,1.0,
102* 2 1.0,1.0,1.0,1.0,1.0,
103* 3 1.0,1.0,1.0,1.0,1.0,
104* 4 1.0,1.0,1.0,1.0,1.0,
105* 5 1.0,1.0,1.0,1.0,1.0 /
106* DATA ((CS( 2,I,J),J=1,5),I=1,5) /
107* 1 1.0,1.0,1.0,1.0,1.0,
108* 2 0.4,0.5,1.0,1.0,1.0,
109* 3 0.4,0.5,1.0,1.0,1.0,
110* 4 1.0,1.0,1.0,1.0,1.0,
111* 5 1.0,1.0,1.0,1.0,1.0 /
112* DATA ((CS( 3,I,J),J=1,5),I=1,5) /
113* 1 1.0,1.0,1.0,1.0,1.0,
114* 2 0.4,0.5,1.0,1.0,1.0,
115* 3 0.4,0.5,1.0,1.0,1.0,
116* 4 1.0,1.0,1.0,1.0,1.0,
117* 5 1.0,1.0,1.0,1.0,1.0 /
118* DATA ((CS( 4,I,J),J=1,5),I=1,5) /
119* 1 1.0,1.0,1.0,1.0,1.0,
120* 2 1.0,1.0,1.0,1.0,1.0,
121* 3 10.0,400.0,1.0,1.0,1.0,
122* 4 1.0,1.0,1.0,1.0,1.0,
123* 5 1.0,1.0,1.0,1.0,1.0 /
124* DATA ((CS( 5,I,J),J=1,5),I=1,5) /
125* 1 1.0,1.0,1.0,1.0,1.0,
126* 2 300.0,700.0,100.0,60.0,1.0,
127* 3 300.0,700.0,100.0,60.0,1.0,
128* 4 300.0,700.0,100.0,60.0,1.0,
129* 5 1.0,1.0,1.0,1.0,1.0 /
130* DATA ((CS( 6,I,J),J=1,5),I=1,5) /
131* 1 1.0,1.0,1.0,1.0,1.0,

```

```

132*      2 1.0,1.0,1.0,1.0,1.0,
133*      3 15000.0,1.0,1.0,1.0,1.0,
134*      4 19650.0,1.0,1.0,1.0,1.0,
135*      5 1.0,1.0,1.0,1.0,1.0 /
136*      DATA ((CS( 7, I, J),J=1,5),I=1,5) /
137*      1 1.0,1.0,1.0,1.0,1.0,
138*      2 1.0,1.0,1.0,1.0,1.0,
139*      3 1.0,1.0,1.0,1.0,1.0,
140*      42,7,430.0,1250.0,1.0,1.0,
141*      5 3.9,500.0,1300.0,1.0,1.0 /
142*      DATA ((CS( 8, I, J),J=1,5),I=1,5) /
143*      1 1.0,1.0,1.0,1.0,1.0,
144*      2 1.0,1.0,1.0,1.0,1.0,
145*      3 2.25,10.00,500.0,1300.0,1.0,
146*      4 1.0,1.0,1.0,1.0,1.0,
147*      5 2.25,10.00,500.0,1300.0,1.0 /
148*      DATA ((CS( 9, I, J),J=1,5),I=1,5) /
149*      1 1.0,1.0,1.0,1.0,1.0,
150*      2 3.9,1400.0,1.0,1.0,1.0,
151*      3 1.0,1.0,1.0,1.0,1.0,
152*      4 1.0,1.0,1.0,1.0,1.0,
153*      5 1.0,1.0,1.0,1.0,1.0 /
154*      DATA ((CS(10, I, J),J=1,5),I=1,5) /
155*      1 1.0,1.0,1.0,1.0,1.0,
156*      2 1.0,1.0,1.0,1.0,1.0,
157*      3 1.0,1.0,1.0,1.0,1.0,
158*      4 3.9,30.0,3900.0,1.0,1.0,
159*      5 1.0,1.0,1.0,1.0,1.0 /
160*      DATA ((CS(11, I, J),J=1,5),I=1,5) /
161*      1 1.0,1.0,1.0,1.0,1.0,
162*      2 1.0,1.0,1.0,1.0,1.0,
163*      3 1000.0,1.0,1.0,1.0,1.0,
164*      4 1.0,1.0,1.0,1.0,1.0,
165*      5 1.0,1.0,1.0,1.0,1.0 /
166*      DATA ((CS(12, I, J),J=1,5),I=1,5) /
167*      1 1.0,1.0,1.0,1.0,1.0,
168*      2 2.25,1.0,1.0,1.0,1.0,
169*      3 2.25,1.0,1.0,1.0,1.0,
170*      4 2.25,1.0,1.0,1.0,1.0,
171*      5 2.25,1.0,1.0,1.0,1.0 /
172*      DATA ((CS(13, I, J),J=1,5),I=1,5) /
173*      1 1.0,1.0,1.0,1.0,1.0,
174*      2 2.00,1.0,1.0,1.0,1.0,
175*      3 2.00,1.0,1.0,1.0,1.0,
176*      4 2.0,2.25,1.0,1.0,1.0,
177*      5 2.0,1.0,1.0,1.0,1.0 /
178*      DATA LV / 1,2,2,2,4,1,3,4,2,3,1,1,2 /
179*      C L(3) 3 MEANS GOOD ** 1 NUMERICAL BUT SOLVED ** 2 INFEASIBLE **3 UNBOUNDED
180*      C ** 4 SYSTEM ERROR *
181*      C X(83) SHOULD HAVE COST
182*      WRITE (6,910)
183*      910 FORMAT ('1 ')
184*      WRITE (6,911)
185*      WRITE (6,911)
186*      WRITE (6,911)
187*      911 FORMAT ('+',4X,'TABLE 1. OCCUPANCY RISK FACTORS')
188*      WRITE (6,924)
189*      924 FORMAT (' ')

```

```

190*      DO 1000 I=1,5
191*      I1=IT1M(I)
192*      WRITE (6,912) I,(HT1(I,J),J=1,2)
193*      912 FORMAT ('0',I1,'. ',A6,A2)
194*      WRITE (6,913) (HT1(I,J),J=3,4),(T1(I,J),J=1,I1)
195*      913 FORMAT (5X,A6,A2,2X,5F5.1)
196* 1000 CONTINUE
197*      WRITE (6,914)
198*      WRITE (6,914)
199*      914 FORMAT ('0          ')
200*      WRITE (6,915)
201*      WRITE (6,915)
202*      WRITE (6,915)
203*      915 FORMAT ('+',8X,'TABLE 6. MANDATORY SAFETY REQUIREMENTS')
204*      WRITE (6,924)
205*      WRITE (6,916)
206*      916 FORMAT (23X,'CONTAINMENT',5X,'EXTINGUISHMENT',2X,'PEOPLE MOVEMENT'
207*      1)
208*      WRITE (6,917)
209*      917 FORMAT (27X,'SA',15X,'SB',15X,'SC')
210*      WRITE (6,918)
211*      918 FORMAT (' ZONE LOCATION ',3(8X,'NEW EXIST'))
212*      WRITE (6,919) ((T6(I,J,1),J=1,2),I=1,3)
213*      919 FORMAT (' FIRST FLOOR ',3(5X,2F6.1))
214*      WRITE (6,920) ((T6(I,J,2),J=1,2),I=1,3)
215*      920 FORMAT (' ABOVE 1ST FLOOR',3(5X,2F6.1))
216*      WRITE (6,914)
217*      WRITE (6,914)
218*      WRITE (6,921)
219*      WRITE (6,921)
220*      WRITE (6,921)
221*      921 FORMAT ('+TABLE 4. PARAMETER VALUES WITH PRESCRIPTIVE INDICES')
222*      WRITE (6,924)
223*      DO 1010 I=1,13
224*      I1=IT4S(I,2)-IT4S(I,1)+1
225*      IF (I.NE.1) GO TO 1005
226*      WRITE (6,922) I,(HPAR(I,J),J=1,5),JPR(I)
227*      WRITE (6,927) (T4A(1,2,J),J=1,2),(T4A(1,2,J),J=1,2),(T4A(1,3,J),J
228*      1=1,3)
229*      WRITE (6,928) (T4A(2,2,J),J=1,2),(T4A(2,2,J),J=1,2),(T4A(2,3,J),J
230*      1=1,3)
231*      WRITE (6,929) (T4A(3,2,J),J=1,2),(T4A(3,2,J),J=1,2),(T4A(3,3,J),J
232*      1=1,3)
233*      WRITE (6,930) (T4A(4,2,J),J=1,2),(T4A(4,2,J),J=1,2),(T4A(4,3,J),J
234*      1=1,3)
235*      927 FORMAT (18X,'FIRST FLOOR ',12X,7F6.1)
236*      928 FORMAT (18X,'SECOND FLOOR',12X,7F6.1)
237*      929 FORMAT (18X,'THIRD FLOOR ',12X,7F6.1)
238*      930 FORMAT (18X,'4TH & ABOVE ',12X,7F6.1)
239*      GO TO 1010
240* 1005 WRITE (6,922) I,(HPAR(I,J),J=1,5),JPR(I),(G(I,J),J=1,I1)
241*      922 FORMAT (I3,'. ',5A6,I6,5F6.1)
242* 1010 CONTINUE
243*      BLANK=' '
244*      STAR='*****'
245*      PRES='PPPPPP'
246*      DO 20 I=1,21 @ INITIALIZE SIMPLEX MATRIX
247*      DO 10 J=1,61

```

```

248*      A(I,J)=0.0
249*      10 CONTINUE
250*      20 CONTINUE
251*      DO 70 I=1,13          @ SIMPLEX ENTRIES BY ROW
252*      A(I,61)=1.0          @ RIGHT HAND SIDE
253*      I1=IT4S(I,1)        @ FIRST SIMPLEX VARIABLE FOR ROW I
254*      I2=IT4S(I,2)        @ LAST SIMPLEX VARIABLE FOR ROW I
255*      DO 60 I3=I1,I2
256*      J1=IST4(I3,2)        @ TABLE 4 ROW FOR SIMPLEX VARIABLE I3
257*      A(I,I3)=1.0
258*      IF (I.EQ.6) GO TO 30          @ NOT IN S1
259*      IF (I.GE.9.AND.I.LE.12) GO TO 30 @ NOT IN S1
260*      A(17,I3)=G(I,J1)*1.0001
261*      30 IF (I.GE.2.AND.I.LE.7) GO TO 40          @ NOT IN S2
262*      IF (I.GE.9.AND.I.LE.10) GO TO 40          @ NOT IN S2
263*      A(18,I3)=G(I,J1)*1.0001
264*      40 X1=1.0
265*      IF (I.EQ.1.OF.I.EQ.3.OF.I.EQ.4) GO TO 50 @ NOT IN S3
266*      IF (I.EQ.8.OR.I.EQ.11) GO TO 50 @ NOT IN S3
267*      IF (I.EQ.13) X1=.5 @ SPRINKLER MULTIPLIER
268*      A(19,I3)=X1*G(I,J1)*1.0001 @ S3
269*      50 A(20,I3)=G(I,J1)*1.0001 @ SG
270*      60 CONTINUE
271*      70 CONTINUE
272*      IA=1
273*      100 READ (5,902) (IT4X(I),I=1,13) , (HOUT(I),I=1,5) @ CUFRENT INDICES I
274*      WRITE (6,910)
275*      ICHAN=0
276*      902 FORMAT (13I3,6A6)
277*      L(3)=0
278*      A(14,54)=1.0
279*      A(15,55)=1.0
280*      A(16,56)=1.0 @ X55-X41
281*      A(14,14)=0.0
282*      A( 4,54)=0.0
283*      A(15,37)=0.0
284*      A(10,55)=0.0
285*      A(16,14)=0.0
286*      A(16,15)=0.0
287*      A(16,10)=0.0
288*      A( 5,56)=0.0
289*      A(17,57)=-1.0
290*      A(18,58)=-1.0
291*      A(19,59)=-1.0
292*      A(20,60)=-1.0
293*      900 FORMAT (23I3)
294*      903 FORMAT (' ',2X,6A6,/)
295*      READ (5,902) (IT1X(I),I=1,5)
296*      R=1.0
297*      DO 1020 I=1,5
298*      I1=IT1X(I)
299*      IF (I1.LE.0.OR.I1.GT.IT1M(I)) I1=IT1M(I)
300*      IT1X(I)=I1
301*      R=R*T1(I,I1)
302*      1020 CONTINUE
303*      DO 105 I=1,60
304*      HFORCE(I)='
305*      105 CONTINUE

```

```

306*      READ (5,900)  IPROB,NAGE,NCON,IFL,IPA
307*      WRITE (6,973) IPROB,(HOUT(I),I=1,6),(ITIX(I),I=1,5)
308* 973 FORMAT ('+PFOBLEM',I4,3X,6A6,3X,'TABLE 1 INDICES ',5I2)
309*      IF (NAGE.NE.2) NAGE=1          @ DEFAULT AGE OF BUILDING
310*      IF (NCON.LT.1.OR.NCON.GT.3) NCON=1      @ DEFAULT PROTECTION
311*      WRITE (6,971) IPROB,AGE(NAGE),CON(NCON),IFL,IPA
312* 971 FORMAT ('OPROBLEM',I4,3X,A6,3X,A6,' FLOOR',I3,' PATIENTS',I4)
313*      A(17,26)=0.0          @ GRADE OPTIONS  ROW 7  COLS 4 AND 5
314*      A(17,27)=0.0
315*      A(19,26)=0.0
316*      A(19,27)=0.0
317*      A(20,26)=0.0
318*      A(20,27)=0.0
319*      IF (IFL.EQ.1.OR.NCON.EQ.1) GO TO 140    @ FIRST FLOOR OR UNPROTECTED
320*      A(17,26)=2.0
321*      A(19,26)=2.0
322*      A(20,26)=2.0
323*      A(17,27)=3.0
324*      A(19,27)=3.0
325*      A(20,27)=3.0
326* 140 A(17,51)=0.0          @ DEFAULT FOW 12 COLUMN 2
327*      A(18,51)=0.0
328*      A(19,51)=0.0
329*      A(20,51)=0.0
330*      IF (NCON.EQ.1) GO TO 150    @ UNPROTECTED TYPE OF CONSTRUCTION
331*      A(17,51)=2.0
332*      A(18,51)=2.0
333*      A(20,51)=2.0
334*      A(19,51)=1.0
335* 150 A(19,33)=0.0          @ ROW 9  COLUMN 1
336*      A(20,33)=0.0
337*      IF (IPA.LT.31) GO TO 152
338*      A(19,33)=-2.0
339*      A(20,33)=-2.0
340* 152 A(14,61)=1.0
341*      A(15,61)=1.0
342*      A(16,61)=1.0
343*      IF (IT4X( 5).GT.1) GO TO 154
344*      A(14,14)=-1.0          @ -X51+X45=0
345*      A(14,54)=1.0          @ ADD X45 TO ROW 4
346*      A( 4,54)=1.0
347*      A(14,61)=0.0
348* C  CONDITIONS ** AND ... ARE NOT ACCOMMODATED IN THIS SECTION NOW  * * * * *
349* 154 IF (NAGE.EQ.2) F=0.6*R          @ EXISTING BUILDING
350*      I=IFL
351*      IF (I.GT.4) I=4
352*      DO 165 I1=1,3
353*      A(17,I1)=T4A(I,NCON,I1)
354*      A(18,I1)=T4A(I,NCON,I1)
355*      A(20,I1)=T4A(I,NCON,I1)
356*      G(1,I1) =T4A(I,NCON,I1)
357* 165 CONTINUE
358*      IF (I.GT.1) I=2
359*      A(17,61)=T6(1,NAGE,I)
360*      A(18,61)=T6(2,NAGE,I)
361*      A(19,61)=T6(3,NAGE,I)
362*      A(20,61)=R
363*      WRITE (6,923) (A(I,61),I=17,20)

```

```

364*      923 FORMAT ('OSA=',F4.1,' SB=',F4.1,' SC=',F4.1,' P=',F4.1)
365*      WRITE (6,972)
366*      972 FORMAT ('0',35X,'I P',4X,5('INPUT',3X,'COST',5X))
367*      DO 110 I=1,13 @ OVER ALL TABLE 4 ROWS
368*      I1=IT4S(I,1) @ FIRST SIMPLEX VARIABLE OF ROW I
369*      I2=IT4S(I,2) @ LAST SIMPLEX VARIABLE OF ROW I
370*      I3=I1+IT4X(I)-1
371*      HFORCE(I3)=' I '
372*      901 FORMAT (I3,5F8.0)
373*      I3=I1+JPR(I)-1
374*      FLD(30,6,HFORCE(I3))=FLD(30,6,PRES)
375*      DO 510 J1=1,5
376*      DO 500 J2=1,5
377*      IQ(J1,J2)=-1
378*      500 CONTINUE
379*      510 CONTINUE @ FLAG BUFFER
380*      J3=IT4X(I)+1 @ INDEX OF 1ST TO READ
381*      J4=I2-I1+1 @ INDEX OF LAST TO READ
382*      K1=LV(I) @ VECTOR LENGTH
383*      K2=J4-J3+1 @ NUMBER OF COLUMNS
384*      K4=K2 @ ASSUME ONE CARD INPUT
385*      513 J1=K1*K4 @ WORDS IN READ
386*      IF (J1.LE.12) GO TO 516 @ ONE CARD INPUT ONLY
387*      K5=J4
388*      K4=K4-1
389*      J4=J4-1
390*      GO TO 513
391*      516 READ (5,965) J,((IQ(J1,J2),J2=1,K1),J1=J3,J4)
392*      965 FORMAT (I3,12I6)
393*      IF (J3.GT.J4) IQ(J3,1)=-1
394*      IF (J.NE.I) GO TO 540
395*      IF (K2.EQ.K4) GO TO 520 @ C NO SECOND CARD
396*      READ (5,965) J,((IQ(J1,J2),J2=1,K1),J1=K5,K2) @ READ TRAILER
397*      IF (J.NE.I) GO TO 540
398*      520 K4=IT4X(I)
399*      J4=I2-I1+1
400*      DO 530 J2=1,6
401*      IQ(K4,J2)=0 @ CLEAR CURRENT STATE VECTOR
402*      530 CONTINUE
403*      DO 560 J1=1,6 @ ALL COLUMNS
404*      IF (IQ(J1,1).GE.0) GO TO 560 @ NO SUPPRESS FLAG -EITHER IN OR COST
405*      DO 550 J2=1,6 @ FLAG LOOP
406*      IQ(J1,J2)=999999999
407*      550 CONTINUE
408*      560 CONTINUE
409*      J6=I1 @ 1ST SIMPLEX VARIABLE INDEX
410*      DO 620 J2=1,J4 @ NUMBER OF COLUMNS IN INPUT
411*      X1=0.0
412*      IF (J2.EQ.IT4X(I)) GO TO 610
413*      X1=999999999
414*      IF (J2.LT.IT4X(I)) GO TO 610
415*      IF (IQ(J2,1).GT.999999990) GO TO 610 @ SUPPRESS FLAG
416*      590 X1=0.0 @ INITIAL COST STORAGE FOR SUM
417*      DO 600 J1=1,K1 @ ALL ELEMENTS OF VECTOR
418*      X2=IQ(J2,J1)
419*      X1=X1+X2*CS(I,J2,J1)
420*      600 CONTINUE
421*      610 C(J6)=X1

```



```

422*      X1=X1+.99
423*      L(J6)=X1
424*      J6=J6+1
425* 620 CONTINUE
426*      K2=I2-I1+1
427*      WRITE (6,966) J,(HPAR(I,J6),J6=1,5), IT4X(I),JPR(I),
428*      1((IQ(J3,1), L(I1+J3-1)),J3=1,K2)
429*      IF (K1.LE.1) GO TO 110 @ ONLY ONE ELEMENT IN COST VECTOR
430* 966 FORMAT ('0',I2,'. ',4A6,A5,2I2,2X,5(1X,I6,1X,I7,2X))
431*      DO 630 J1=2,K1
432*      WRITE (6,967) (IQ(J3,J1),J3=1,K2)
433* 967 FORMAT (31X,5(11X,I6))
434* 630 CONTINUE
435* 110 CONTINUE
436*      GO TO 640
437* 540 WRITE (6,968) I
438* 968 FORMAT ('0PARAMETER CARD',I3,' INCORRECT - RUN ABORTED')
439*      STOP
440* 640 X1=0.0 @ INITIALIZE SEARCH FOR MAXIMUM COST
441*      DO 120 I=1,53
442*      IF (C(I).GT.X1.AND.C(I).LT.999999990) X1=C(I) @ MAXIMUM LEGITIMATE CO
443* 120 CONTINUE
444*      X1=20.0*X1 @ MAGNITUDE OF SIMPLEX DUMMY COST
445*      DO 130 I=1,53
446*      IF (C(I).LT.0.0.OR.C(I).GT.X1) C(I)=X1 @ REPLACE FLAGS WITH ARTIFICIA
447* 130 CONTINUE
448*      X2=.95*X1 @ LIMIT FOR OBJECTIVE FUNCTION
449*      Z(54)=X2
450*      IALT=0
451*      DO 5 I=1,14
452*      L(I)=0
453* 5 CONTINUE
454*      L(5)=1
455*      L(14)=1
456*      DO 170 I=1,53
457*      A(21,I)=-C(I)
458*      Z(I)=A(20,I)
459* 170 CONTINUE
460* 300 READ (5,960) I,I1,I2,I4 @ READ CHANGE INPUT
461* 970 FORMAT (1X,A6,2I3,I7,4I3)
462* 960 FORMAT (A6,2I3,I7)
463*      IF (I.EQ.'SOLVE ') GO TO 370
464*      IF (I.EQ.'NEXT ') GO TO 800
465*      IF (I.EQ.'END ') STOP
466*      IF (I.EQ.'CHANGE') GO TO 320
467*      IF (I.NE.'REQUIR') GO TO 310
468*      J=I1+16
469*      X3=I2
470*      A(J,61)=(1.0+.01*X3)*A(J,61)
471*      GO TO 320
472* 310 WRITE (6,961) I,I1,I2,I4
473* 961 FORMAT ('0USER MOD INCORRECT ',A6,2I3,I7,' - RJN ABORTED')
474*      STOP
475* 320 IALT=1
476*      IF (ICHAN.EQ.0) WRITE (6,962) IPROB,(HOUT(J4),J4=1,6)
477* 962 FORMAT ('0MODIFIED PROBLEM',I4,3X,6A6)
478*      ICHAN=1
479*      IF (I.NE.'REQUIR') GO TO 325

```

```

480*      WRITE (6,963) I,I1 ,X3,A(J,€1)
481* 963 FORMAT ('0',A6, I3,F6.2,F11.2)
482*      GO TO 300
483* 325 J1=IT4S(I1,1)
484*      J2=IT4S(I1,2)
485*      J4=J1+I2-1
486*      J=I1
487*      IF (J4.LT.J1.OR.J4.GT.J2) GO TO 310 @ INDEX OUT OF RANGE
488*      FLD(€,€,HFORCE(J4))=FLD(0,€,'CCCCCC')
489*      IF (I4.LT.0) GO TO 330 @ SUPPRESS
490*      IF (I4.EQ.0) GO TO 340 @ FORCE IN W ORIGINAL COST
491*      A(21,J4)=-I4 @ COST GIVEN ON INPUT
492*      C(J4)=I4
493*      GO TO 360
494* 330 A(21,J4)=-X1 @ COST AS INPUT
495*      C(J4)=X1
496*      GO TO 360
497* 340 IT4X(J)=I2
498*      DO 350 J3=J1,J2
499*      IF (J3.EQ.J4) GO TO 350 @ INPUT STATE
500*      A(21,J3)=-X1
501*      C(J3)=X1
502*      FLD(€,€,HFORCE(J4))=FLD(€,€, 'FFFFFF')
503* 350 CONTINUE
504* 360 WRITE (6,964) I,I1,I2,IT4X(J),(A(21,J3),J3=J1,J2)
505* 964 FORMAT ('0',A€, ' ROW',I3, ' STATE',I2, ' (' ,I2, ' )',2X,5F11.2)
506*      GO TO 300
507* 370 ICHAN=0
508*      WRITE (6,963) I
509*      DO 375 I=1,13
510*      I1=IT4S(I,1)+JPR(I)-1
511*      FLD(30,€,HFORCE(I1))=FLD(30,€,PRES)
512* 375 CONTINUE
513*      L(1)=0
514*      L(2)=0
515*      L(3)=0
516* 950 FORMAT (1X,10F12.2) @ * * * * *
517*      WRITE (6,910)
518*      CALL PRSMX (A,21,€1,HPAR,5,IST4,€0,2,IALT)
519* 172 CALL RVSMX(A,21,B,23,21,€1,L,X,0,0,0)
520*      CALL FEASIB(A,€1,21,X,L,X2)
521*      IF (L(3).GT.1) GO TO 280
522*      DO 180 I=1,150
523*      Y(I)=X(I)
524* 180 CONTINUE
525*      X(81)=-X(81)
526*      DO 175 I3=1,4
527*      X(I3+81)=A(I3+16,€1)
528*      X(I3+90)=-X(I3+81)
529* 175 CONTINUE
530*      X(90)=0.0
531*      DO 178 I3=1,53
532*      J=' '
533*      FLD(30,€,J)=FLD(30,€,HFORCE(I3))
534*      IF (J.EQ.BLANK) GO TO 178
535*      I5=I3
536*      I6=IST4(I3,1) @ TABLE 4 FOW FOR THIS VARIABLE
537*      I7=IT4X(I€)-JPR(I6) @ CUFFENT COLUMN MINUS PRESCRIPTIVE COLUMN

```

```

538*      IF (I7.GT.0) I5=I5+I7          @CURRENT BETTER THA PRESCRIPTIVE
539*      IF (X(90).LT.0.0) GO TO 176    @ PRESCRIPTIVE COST IMPOSSIBLE
540*      X(90)=X(90)+C(I5)              @ CUMULATIVE PRESCRIPTIVE COST
541*      IF (C(I5).GE.X2) X(90)=-1.0
542*      176 DO 177 I4=1,4              @ ACCUMULATE GRADES FOR PRESCRIPTIOM
543*      X(I4+90)=X(I4+90)+A(I4+16,I5)
544*      177 CONTINUE
545*      178 CONTINUE
546*      X(150)=X2
547*      CALL PRSOL (IPROB,IALT,C,Z,HFORCE,IST4,60,2,IT4X,X,IT4S,13,HPAR,21
548*      1,5,L,A,61,HOUT)
549*      IF (L(3).GT.1) GO TO 260
550*      IF (IALT.LT.0) GO TO 280
551*      DO 230 I=1,2
552*      L(1)=0
553*      L(2)=0
554*      L(3)=0
555*      IALT=I+1000
556*      I1=IT4S(I,1)                   @ FIRST SIMPLEX INDEX FOR THIS FOW
557*      I2=IT4S(I,2)                   @ LAST SIMPLEX INDEX FOR THIS FOW
558*      CALL MOTEST (HFORCE,I1,I2,I4)
559*      IF (I4.NE.0) GO TO 230
560*      I4=IT4X(I)+I1-1                 @ SIMPLEX OF CURRENT STATE
561*      IF (X(I4).GT.0.99) GO TO 230
562*      200 DO 210 I3=1,150
563*      X(I3)=Y(I3)
564*      IF (I3.GT.53) GO TO 210
565*      Z(I3)=A(20,I3)                  @ PRESENT GRADE VALUES
566*      A(21,I3)=-C(I3)
567*      FLD( 0, 6,HFORCE(I3))=FLD( 0, 6,BLANK)
568*      FLD(12,12,HFORCE(I3))=FLD(12,12,BLANK)
569*      FLD(30,6,HFORCE(I3))=FLD(30,6,BLANK)
570*      210 CONTINUE
571*      DO 220 I3=I1,I2
572*      IF (I3.EQ.I4) GO TO 220
573*      FLD(0,6,HFORCE(I3))=FLD(0,6,STAR)
574*      A(21,I3)=-X1
575*      220 CONTINUE
576*      CALL RVSMPX(A,21,B,23,21,61,L,X,0,1,0)
577*      CALL FEASIB(A,61,21,X,L,X2)
578*      IF (L(3).GT.1) GO TO 230
579*      X(81)=-X(81)
580*      DO 225 I3=1,4
581*      X(I3+81)=A(I3+16,61)
582*      225 CONTINUE
583*      X(150)=X2
584*      CALL PRSOL (IPROB,IALT,C,Z,HFORCE,IST4,60,2,IT4X,X,IT4S,13,HPAR,21
585*      1,5,L,A,61,HOUT)
586*      230 CONTINUE
587*      IA=2
588*      DO 270 I=1,3
589*      I4=IST4(I,1)
590*      I1=IT4S(I4,1)
591*      I2=IT4S(I4,2)
592*      CALL MOTEST (HFOFCE,I1,I2,I4)
593*      IF (I4.NE.0) GO TO 270
594*      L(1)=0
595*      L(2)=0

```

*13*****

*53*****

```

596*      L(3)=0
597*      IALT=2000+10*IST4(I,1)+IST4(I,2)
598*      IF (C(I).GE.X2) GO TO 270
599*      DO 240 I1=1,150
600*      X(I1)=Y(I1)
601*      IF (I1.GT.53) GO TO 240
602*      Z(I1)=A(20,I1) @ PRESENT GRADE VALUES
603*      A(21,I1)=-C(I1)
604*      FLD( 0, 6,HFORCE(I1))=FLD( 0, 6,BLANK)
605*      FLD(12,12,HFORCE(I1))=FLD(12,12,BLANK)
606* 240 CONTINUE
607*      FLD(0,6,HFORCE(I ))=FLD(0,6,STAR)
608*      I4=0
609*      I1=IST4(I,1) @ TABLE 4 ROW
610*      I5=IT4S(I1,1) @ INDEX OF FIRST SIMPLEX VARIABLE THIS RO
611*      I2=IT4S(I1,2) @ INDEX OF LAST SIMPLEX VARIABLE THIS FO
612*      I6=I5+IT4X(I1)-1
613*      DO 250 I3=I5,I2 @ OVER ALL VARIABLES THIS ROW
614*      IF (I3.EQ.I6) GO TO 250 @ DONT COUNT IS CURRENT OF INPUT INDEX
615*      IF (C(I3).GT.X2) GO TO 250
616*      I4=I4+1 @ LEGITIMATE ALTERNATIVE COUNT
617* 250 CONTINUE
618*      IF (I4.EQ.0) GO TO 270 @ NO OPTION FOR THIS PARAMETER
619*      IF (Y(I).GE.0.05) GO TO 256 @ VARIABLE IS NOW IN BASIS - AND CAN BE O
620*      DO 253 I3=I5,I2 @ VARIABLE IS OUT BUT CAN BE IN
621*      IF (I3.EQ.I) GO TO 253
622*      A(21,I3)=-X1 @ ALL COSTS EXCEPT I6TH
623* 253 CONTINUE
624*      GO TO 260
625* 256 A(21,I)=-X1 @ FOFCE OUT
626* 260 I3=1 @ DUMMY TO GET STATEMENT NUMBER
627*      CALL RVSMPX(A,21,B,23,21,61,L,X,0,1,0)
628*      CALL FEASIB(A,61,21,X,L,X2)
629*      IF (L(3).GT.1) GO TO 270
630*      X(81)=-X(81)
631*      DO 265 I3=1,4
632*      X(I3+81)=A(I3+16,61)
633* 265 CONTINUE
634*      X(150)=X2
635*      CALL PRSOL (IPROB,IALT,C,Z,HFORCE,IST4,60,2,ITAX,X,IT4S,13,HPAR,21
636*      1,5,L,A,61,HOUT)
637* 270 CONTINUE
638* 280 WRITE (6,910)
639*      DO 290 I1=1,150
640*      X(I1)=Y(I1)
641*      IF (I1.GT.53) GO TO 290
642*      Z(I1)=A(20,I1)
643*      A(21,I1)=-C(I1)
644*      FLD( 0, 6,HFORCE(I1))=FLD( 0, 6,BLANK)
645*      FLD(12,12,HFORCE(I1))=FLD(12,12,BLANK)
646* 290 CONTINUE
647*      GO TO 300
648* 800 READ (5,815) I,I1,(X(I2),I2=1,5) @ TEMP *****
649*      IF (I.GT.13) GO TO 100 @ TEMP *****
650*      DO 810 I2=1,5 @ TEMP *****
651*      CS(I,I1,I2)=X(I2) @ TEMP *****
652* 810 CONTINUE @ TEMP *****
653* 815 FORMAT (2I3,5F6.0) @ TEMP *****
654*      GO TO 800 @ TEMP *****
655*      END

```

D.2 Listing of Subroutine PRSOL (PRint SOLutions)

```

1*      SUBROUTINE PRSOL (IPROB,IALT,C,Z,HFORCE,IST4,N1,N2,IT4X,X,IT4S,N3,
2*      1 HPAR,N4,N5,L,A,N6,HOUT)
3*      DIMENSION C(1),Z(1),HFORCE(1),IST4(N1,N2),IT4X(1),X(1),IT4S(N3,N2
4*      1),HPAR(N4,N5),L(1),A(N4,N6),HOUT(1)
5*      SOLN='SSSSSS'
6*      X(100)=0.0
7*      IF (IALT.NE.0.AND.IALT.NE.1)      X(100)=10.0
8*      C CONDITIONS ** AND ... ARE NOT ACCOMMODATED IN THIS SECTION NOW      * * * * *
9*      IF (X(14).LT.0.05) GO TO 60      @ X51 NOT IN BASIS SIDE CONDITIONS DO
10*     DO 10 I=10,13      @ CLEAR BASIS AND GRADES FOR ROW 4 WHICH CANNOT BE
11*     X(I)=0.0      @ FOUND WITH X51 IN THE SOLUTION. THIS IS CONDITI
12*     Z(I)=0.0      @ * FROM TABLE 4
13*     10 CONTINUE
14*     DO 20 I=1,4      @ INITIALIZE CONSTRUCTED SURPLUS GRADES
15*     X(I+56)=-X(I+81)      @ INITIAL SURPLUS IS NEGATIVE OF REQUIREMENT
16*     20 CONTINUE
17*     DO 40 I=1,53      @ OVER ALL VARIALBES
18*     DO 30 J=1,4      @ OVER ALL REQUIREMENTS
19*     IF (A(J+16 ,I).LT.0.0000R.A(J+16 ,I).GT.0.0) X(J+56)=X(J+56)+X(I)*
20*     1Z(I)
21*     30 CONTINUE
22*     40 CONTINUE
23*     DO 50 I=57,60      @ OVER SURPLUS VARIABLES
24*     IF (X(I).GE.-0.01) GO TO 50      @ SOLUTION IS SSTILL VALID
25*     WRITE (6,916) ((X(J),X(J+25)),J=57,60)      @ WRITE INVALID SOLUTION
26*     RETURN
27*     50 CONTINUE
28*     60 WRITE (6,900) IPROB,(HOUT(I),I=1,6),IALT
29*     WRITE (6,901)
30*     K=0
31*     IF (L(3).GE.2) GO TO 64
32*     DO 62 I=1,53
33*     IF (X(I).LT.0.0005) GO TO 62
34*     IF (C(I).LT.X(150)) GO TO 62
35*     L(3)=2
36*     GO TO 64
37*     62 CONTINUE
38*     64 IF (L(3).EQ.2) WRITE (6,912)
39*     IF (L(3).EQ.3) WRITE (6,913)
40*     IF (L(3).GE.4) WRITE (6,914)
41*     IF (L(3).GT.1) RETURN
42*     DO 80 I=1,53
43*     I1=IST4(I,1)      @ ROW
44*     I2=IST4(I,2)      @ COLUMN
45*     IF (I2.NE.1) GO TO 70
46*     WRITE (6,909)
47*     CALL RHTAB (K,X)
48*     WRITE (6,903) I1,(HPAR(I1,J),J=1,5)      @ FIRST OF T4 ROW

```

```

49*      CALL RHTAB (K,X)
50*      65 I3=J
51*      70 IF (X(I).GT.0.05) FLD(12,6,HFORCE(I))=FLD(12,6,SOLN)
52*      IF (HFORCE(I).EQ.' ') GO TO 80
53*      IF (I3.EQ.0) GO TO 75
54*      WRITE (6,909)
55*      CALL RHTAB (K,X)
56*      75 I3=1
57*      XI=X(I)
58*      IF (XI.LE.0.0) XI=J.0
59*      CI=C(I)
60*      IF (C(I).GT.X(150)) CI=1.0E 15
61*      WRITE (6,902) I,XI ,IST4(I,2),CI ,Z(I),HFORCE(I)
62*      80 CONTINUE
63*      IF (X(81).LT.0.05) IALT=-1
64*      IF (X(81).LT.0.05) WRITE (6,910)
65*      IF (L(3).EQ.1) WRITE (6,911)
66*      IF (L(3).EQ.2) WRITE (6,912)
67*      IF (L(3).EQ.3) WRITE (6,913)
68*      IF (L(3).GE.4) WRITE (6,914)
69*      RETURN
70*      900 FORMAT ('1',19X,'PROBLEM',I4,3X,6A6,3X,'ALTERNATE',I5)
71*      901 FORMAT ('0PARAMETER',25X,'VARIABLE TABLE 4 COST GRADE'//)
72*      902 FORMAT ('+',34X,I2,F6.3,I6,4X,F8.2,F8.2,3X,A6)
73*      903 FORMAT (' ',I2,' ',5A6)
74*      909 FORMAT (' ')
75*      910 FORMAT ('0INITIAL STATE PASSES ALL REQUIREMENTS')
76*      911 FORMAT ('0BAD NUMERICAL PROBLEM * SMALL PIVOT USED')
77*      912 FORMAT ('0NON-FEASIBLE PROBLEM')
78*      913 FORMAT ('0UNBOUNDED PROBLEM')
79*      914 FORMAT ('0SYSTEM ERROR')
80*      915 FORMAT ('0ILLEGAL STATE INCLUDED IN OPTIMUM SOLUTION * IGNORE')
81*      916 FORMAT ('0NEXT ALTERNATE INVALID * ROW 4 AND 5 CONDITIONS * SURPLU
82*      IS AND RHS',4(2F5.1,3X))
83*      END

```

D.3 Listing of Subroutine FEASIB (FEASIBILITY test)

```

1*      SUBROUTINE FEASIB (A,I1,I2,X,L,X2)
2*      DIMENSION A(I2,I1),X(1),L(1)
3*      X3=-X2
4*      DO 10 I=1,53
5*      IF (X(I).LT.0.001) GO TO 10
6*      IF (A(21,I).LT.X3) GO TO 20
7*      10 CONTINUE
8*      RETURN
9*      20 L(3)=2
10*     RETURN
11*     END

```

D.4 Listing of Subroutine MOTEST (Modification TEST)

```

1*      SUBROUTINE MOTEST (HFORCE,I1,I2,I3)
2*      DIMENSION HFORCE(1)
3*      J=0
4*      I3=0
5*      DO 10 I=I1,I2
6*      FLD(6,6,J)=FLD(6,6,HFORCE(I))
7*      IF (J.EQ.' ' .OR. J.EQ.' C ' ) GO TO 10
8*      I3=1
9*      RETURN
10*     10 CONTINUE
11*     RETURN
12*     END

```

D.5 Listing of Subroutine RHTAB (Right Hand TABLES)

```

1*      SUBROUTINE RHTAB (K,X)
2*      DIMENSION X(1)
3*      K=K+1
4*      IF (K.GT.11.AND.X(100).GT.0.0) RETURN
5*      IF (K.EQ. 3) WRITE (6,904) X(81)
6*      IF (K.EQ. 5) WRITE (6,905) X(82),X(57)
7*      IF (K.EQ. 7) WRITE (6,906) X(83),X(58)
8*      IF (K.EQ. 9) WRITE (6,907) X(84),X(59)
9*      IF (K.EQ.11) WRITE (6,908) X(85),X(60)
10*     IF (K.LT.15) RETURN
11*     IF (K.EQ.15.AND.X(90).LT.0.0) WRITE (6,909)
12*     IF (K.EQ.15.AND.X(90).GE.0.0) WRITE (6,910) X(90)
13*     IF (K.EQ.17) WRITE (6,905) X(82),X(91)
14*     IF (K.EQ.19) WRITE (6,906) X(83),X(92)
15*     IF (K.EQ.21) WRITE (6,907) X(84),X(93)
16*     IF (K.EQ.23) WRITE (6,908) X(85),X(94)
17*     RETURN
18*     904 FORMAT ('+',82X,'TCTAL COST',F13.2)
19*     905 FORMAT ('+',82X,'SA=',F4.1,' SURPLUS=',F5.1)
20*     906 FORMAT ('+',82X,'SB=',F4.1,' SURPLUS=',F5.1)
21*     907 FORMAT ('+',82X,'SC=',F4.1,' SURPLUS=',F5.1)
22*     908 FORMAT ('+',82X,'SG=',F4.1,' SURPLUS=',F5.1)
23*     909 FORMAT ('+',82X,'PRESCRIPTIVE IMPOSSIBLE')
24*     910 FORMAT ('+',82X,'PRESCRIPTIVE',F11.2)
25*     END

```

D.6 Listing of Subroutine RVSMPX (ReVised SiMPlex)

```

1*          SUBROUTINE RVSMPX (A,MA,B,MB,MT,NT,L,X,TOLP,INV,KQP)
2*          C
3*          C *****
4*          C
5*          DIMENSION A(MA,1),B(MB,1),L(1),X(1),IB(5),IP(5),PR(5)
6*          C
7*          C * * CHECK FOR NEGATIVE RIGHT HAND SIDES.
8*          C
9*          DO 10 I=1,M
10*         IF (A(I,NT).GE.0) GO TO 10
11*         WRITE (6,1140)
12*         L(3)=2
13*         RETURN
14*    10     CONTINUE
15*          C
16*          C * * SET UP CONSTANTS FOR LATER USE.
17*          C
18*          N=NT-1
19*          M=MT-1
20*          M2=M+2
21*          INVC=M/2+5
22*          ML=L(1)
23*          MSMX=14
24*          MG=L(2)
25*          MPN=M+N
26*          MP1=M+1
27*          ME=M-ML-MG
28*          NMAX=N+ML+ME+2*MG
29*          ASTAR='*'
30*          ITP=N-KQP
31*          ITC=KQP+ML+MG
32*          L(2)=0
33*          IF (INV.GT.0) INV=1
34*          C
35*          C * * COMPUTE EPSILON AND CAPITAL EPSILON.
36*          C
37*          EPS=TOLP
38*          IF (EPS.GT.0.0) GO TO 30
39*          XX=1.0E-5
40*          IF (EPS.LT.0.0) XX=-EPS
41*          Y=0.0
42*          DO 20 I=1,M
43*             DO 20 J=1,N
44*                EPS=A(I,J)
45*                IF (EPS.LT.0.0) EPS=-EPS
46*    20     Y=Y+EPS
47*          EPS=XX*Y/(M*N)
48*    30     IF (L(3).EQ.0) GO TO 50
49*          DO 40 I=3,MSMX
50*    40     L(I)=0
51*          L(4)=1
52*          L(5)=1
53*          L(8)=1
54*          L(12)=1
55*          L(13)=1
56*          L(MSMX)=1
57*    50     CEPS=10.0*EPS
58*          L(1)=0
59*          DO 60 I=1,M
60*             DO 60 J=1,N
61*                XX=ABS(A(I,J))

```



```

62*      60      IF (XX.LT.EPS.AND.XX.GT.0.0) L(1)=L(1)+1
63*      IF (L(4).NE.0) WRITE (6,980) EPS,CEPS,L(1)
64*      C
65*      C * * SET UP L1(.)-VARIABLE TYPE INDICATOR.
66*      C
67*      L1OFF=MSMX
68*      L2OFF=L1OFF+NMAX
69*      I=1
70*      DO 70 J=1,N
71*          L(L1OFF+I)=0
72*      70      I=I+1
73*      IF (ML.EQ.0) GO TO 90
74*      DO 80 J=1,ML
75*          L(L1OFF+I)=1
76*      80      I=I+1
77*      90      IF (MG.EQ.0) GO TO 110
78*      DO 100 J=1,MG
79*          L(L1OFF+I)=2
80*          L(L1OFF+I+MG)=3
81*      100     I=I+1
82*      110     IF (ME.EQ.0) GO TO 130
83*      DO 120 J=1,ME
84*          L(L1OFF+I+MG)=3
85*      120     I=I+1
86*      C
87*      C * * INITIALIZE B(.,.), THE BASIS INVERSE, TO THE IDENTITY MATRIX.
88*      C * * ALSO SET B(.,M2), THE VALUES OF THE BASIC VARIABLES.
89*      C
90*      130     DO 150 I=1,M2
91*          DO 140 J=1,M2
92*      140         B(I,J)=0.0
93*          IF (I.GT.M) GO TO 150
94*          B(I,I)=1.0
95*          B(I,M2)=A(I,NT)
96*      150     CONTINUE
97*      IF (INV.GT.0) GO TO 170
98*      C
99*      C * * INITIALIZE THE X'S, B(M2,.)-THE MULTIPLIERS, AND B(M2,M2)-THE
100*      C * * ARTIFICIAL OBJECTIVE FUNCTION VALUE. FOR ANY ZERO-VALUED
101*      C * * ARTIFICIALS IN THE BASIS, RESET B(.,M2) TO 0.
102*      C
103*      DO 160 I=1,NMAX
104*      160      X(I)=-EPS/100.0
105*      170      J=1
106*      DO 200 I=1,NMAX
107*          I1=L(L1OFF+I)
108*          IF (I1.EQ.0.OR.I1.EQ.2) GO TO 200
109*          IF (INV.GT.0) GO TO 180
110*          X(I)=A(J,NT)
111*          L(L2OFF+J)=I
112*      180      IF (I1.EQ.5) B(J,M2)=0.0
113*          IF (I1.LT.3) GO TO 190
114*          B(M2,J)=1.0
115*          B(M2,M2)=B(M2,M2)+B(J,M2)
116*      190      J=J+1
117*      200      CONTINUE
118*      IF (INV.EQ.2) GO TO 820
119*      C

```

```

120*   C * * INITIALIZATIONS FOR PHASE I.
121*   C
122*       NA=0
123*       IPHASE=1
124*       MOBJ=M2
125*       ITER=0
126*       FIVM=1.0E28
127*       IMP=0
128*       OBJ=-B(MOBJ,M2)
129*       IF (INV.EQ.1) GO TO 820
130*   C
131*   C * * CHECK FOR IMMEDIATE BASIC FEASIBLE SOLUTION.
132*   C
133*       IF (MLoLToM) GO TO 220
134*   C
135*   C * * INITIALIZATIONS FOR PHASE II.
136*   C
137*   210   MOBJ=MP1
138*       IPHASE=2
139*       ITER=0
140*       FIVM=1.0E28
141*       IMP=0
142*       OBJ=-B(MOBJ,M2)
143*   C
144*   C *****
145*   C ** STEPS 1 AND 2-COMPUTE THE C' TO GET S.**
146*   C *****
147*   C
148*   220   ICPSX=0
149*   230   NA=0
150*       IF (INV.GT.0) GO TO 240
151*       JS=0
152*       IF (L(2)EQ0ORICPSXEQ0) GO TO 240
153*       IF (L(2)/INVC*INVC.EQ.L(2)) INV=2
154*       IF (INV.EQ.2) GO TO 130
155*   C
156*   C * * CALCULATE THE INDICATORS-C'.
157*   C
158*   240   B(MOBJ,MP1)=0.0
159*       DO 340 J5=1,NMAX
160*           IF (INV.EQ.0) GO TO 250
161*           J=JS
162*           GO TO 260
163*   250   J=J5
164*           IF (X(J).GE.0.0) GO TO 340
165*           IF (KQP.EQ.0) GO TO 260
166*           IF (JLEITCANDX(ITP+J)GE00) GO TO 340
167*           IF (J.GT.ITP.AND.J.LE.ITP+ITC.AND.X(J-ITP).GE.0.0) GO TO 340
168*   260   J1=L(L1OFF+J)
169*           IF (J1.EQ.4) GO TO 340
170*           IF (J1.EQ.0) GO TO 270
171*           J2=J-N
172*           IF (J.GT.N+ML+MG) J2=J2-MG
173*   C
174*   C * * COMPUTE C' FOR SLACK, SURPLUS, OR ARTIFICIAL VARIABLES.
175*   C
176*       XX=(-1)**(J1+1)*B(MOBJ,J2)
177*       IF (J1.EQ.3) XX=XX-1.0

```

```

178*          GO TO 290
179*          C
180*          C * * COMPUTE C' FOR REAL VARIABLES.
181*          C
182*          270          XX=0.0
183*                   IF (IPHASE.EQ.2) XX=A(MP1,J)
184*                   DO 280 I=1,M
185*          280          XX=XX+B(MOBJ,I)*A(I,J)
186*          C
187*          C * * TRANSFER IF INVERTING OR PIVOTING OUT ARTIFICIALS.
188*          C
189*          290          IF (INV.GT.0.0.OR.ICPSX.EQ.1) GO TO 300
190*          C
191*          C * * TEST FOR MINIMUM C' IF DOING A NORMAL PIVOT.
192*          C
193*                   IF (XX.LE.B(MOBJ,MP1)) GO TO 340
194*          300          B(MOBJ,MP1)=XX
195*                   JS=J
196*                   J3=J2
197*                   IF (INV.GT.0) GO TO 350
198*                   IF (ICPSX.EQ.0) GO TO 340
199*          C
200*          C * * IF TRYING TO PIVOT OUT ARTIFICIALS, CHECK THIS COLUMN. IF OK TO
201*          C * * ENTER, COMPUTE PIVOT ELEMENT. IF PIVOT IS BIG ENOUGH, PIVOT.
202*          C * * OTHERWISE, CONTINUE ON TO NEXT COLUMN.
203*          C
204*                   IF (XX.LT.-EPS) GO TO 340
205*                   JS=J
206*                   IF (J1.EQ.0) GO TO 310
207*                   B(IR,MP1)=(-1)**(J1+1)*B(IR,J2)
208*                   GO TO 330
209*          310          B(IR,MP1)=0.0
210*                   DO 320 K=1,M
211*          320          B(IR,MP1)=B(IR,MP1)+B(IR,K)*A(K,JS)
212*          330          IF (B(IR,MP1).GT.EPS) GO TO 350
213*                   IF (B(IR,MP1).GT.-EPS) GO TO 340
214*                   GO TO 350
215*          340          CONTINUE
216*                   IF (ICPSX.EQ.1) GO TO 680
217*                   IF (B(MOBJ,MP1).LT.EPS) GO TO 580
218*          C
219*          C *****
220*          C *** STEP 3-CALCULATE A'(I,S). ***
221*          C *****
222*          C
223*          350          I1=L(L1CFF+JS)
224*                   IB(1)= ' X( '
225*                   IP(1)=JS
226*                   IF (I1.EQ.0) GO TO 360
227*                   IB(1)= ' S( '
228*                   IP(1)=JS-N
229*                   IF (I1.LT.3) GO TO 360
230*                   IB(1)= ' A( '
231*                   IP(1)=IP(1)-ML-MG
232*          360          DO 410 I=1,MP1
233*          C
234*          C * * COMPUTE REAL OBJECTIVE COST IF NOT ALREADY COMPUTED.
235*          C

```

```

236*          IF (I0.LT.MP1) GO TO 370
237*          IF (IPHASE.EQ.2) GO TO 410
238*          B(I,MP1)=0.0
239*          IF (I1.EQ.0) B(I,MP1)=A(MP1,JS)
240*          IF (I1.EQ.3) B(I,MP1)=-1.0
241*          GO TO 380
242*      C
243*      C * * GENERATE THE REST OF THE COLUMN.
244*      C
245*      370          B(I,MP1)=0.0
246*          I2=L(L2OFF+I)
247*          IF (I2.EQ.-1) GO TO 380
248*          IF (L(L1OFF+I2).EQ.5) GO TO 410
249*      380          IF (I1.EQ.0) GO TO 390
250*          B(I,MP1)=B(I,MP1)+(-1)**(I1+1)*B(I,J3)
251*          GO TO 410
252*      390          DO 400 K=1,M
253*      400              B(I,MP1)=B(I,MP1)+B(I,K)*A(K,JS)
254*      410          CONTINUE
255*      C
256*      C * * IF PIVOTING OUT AN ARTIFICIAL, SET THE VALUE OF THIS ARTIFICIAL
257*      C * * TO ZERO. IF IT IS GT EPS SET L(3)=1 (COMPUTATIONAL INCONSISTENCY)
258*      C
259*          IF (ICPSX.EQ.0) GO TO 420
260*          IF (B(IR,M2).GT.EPS) L(3)=1
261*          B(IR,M2)=0.0
262*          GO TO 480
263*      C
264*      C *****
265*      C *** STEP 4-FORM QUOTIENTS TO GET R. ***
266*      C *****
267*      C
268*      420          IR=0
269*          APS=1.0E28
270*          IF (INV.GT.0) APS=0.0
271*          DO 450 I=1,M
272*              IF (INV.GT.0) GO TO 430
273*              IF (B(I,MP1).LE.0.0) GO TO 450
274*              XX=B(I,M2)/B(I,MP1)
275*              IF (XX.GT.APS) GO TO 450
276*              IF (XX.LT.APS.OR.IR.EQ.0) GO TO 440
277*              IF (B(IR,MP1).GE.0.B(I,MP1)) GO TO 450
278*              GO TO 440
279*      C
280*      C * * IF INVERTING, FIND THE LARGEST ABSOLUTE ELEMENT AMONG
281*      C * * THOSE ROWS NOT ALREADY PIVOTED IN.
282*      C
283*      430          IF (L(L2OFF+I).NE.-1) GO TO 450
284*          XX=ABS(B(I,MP1))
285*          IF (XX.LE.APS) GO TO 450
286*      440          IR=I
287*          APS=XX
288*      450          CONTINUE
289*          IF (IR.GT.0) GO TO 460
290*      C
291*      C * * IF INVERTING, AND THERE IS NO PLACE TO PUT THIS
292*      C * * VARIABLE, SKIP IT AND TRY AGAIN LATER.
293*      C

```

```

294*      IF (INV.GT.0) GO TO 863
295*      C
296*      C * * THE PROBLEM IS UNBOUNDED.
297*      C
298*      L(3)=3
299*      IX=1
300*      IF (L(MSMX).EQ.0) RETURN
301*      WRITE (6,1100) (ASTAR,I=1,120),JS
302*      WRITE (6,1120) ITER,IPHASE,OBJ,(ASTAR,I=1,120)
303*      GO TO 720
304*      460  XX=ABS(B(IR,MP1))
305*      IF (XX.GT.CEPS) GO TO 480
306*      C
307*      C * * IF PIVOT IS TOO SMALL, SET TO ZERO AND TRY AGAIN.
308*      C
309*      IF (XX.LT.EPS) GO TO 470
310*      C
311*      C * * IF EPS.LT.PIVOT.LT.CEPS, PRINT WARNING AND CONTINUE.
312*      C
313*      IF (L(5).NE.0) WRITE (6,1020) ITER,IPHASE,B(IR,MP1)
314*      GO TO 480
315*      470  B(IR,MP1)=0.0
316*      GO TO 420
317*      C
318*      C *****
319*      C *** STEP 5-CALCULATE NEW B(.,MT) AND X. ***
320*      C *****
321*      C
322*      480  IF (INV.GT.0) GO TO 490
323*      I1=L(L2OFF+IR)
324*      I2=L(L1OFF+I1)
325*      X(I1)=-EPS/100.0
326*      IF (L(L1OFF+I1).EQ.3) L(L1OFF+I1)=4
327*      IB(2)= X(      )
328*      IP(2)=I1
329*      IF (I2.EQ.0) GO TO 490
330*      IB(2)= S(      )
331*      IP(2)=I1-N
332*      IF (I2.LT.3) GO TO 490
333*      IB(2)= A(      )
334*      IP(2)=IP(2)-ML-MG
335*      490  L(L2OFF+IR)=JS
336*      B(IR,M2)=B(IR,M2)/B(IR,MP1)
337*      DO 510 I=1,M2
338*          IF (I.EQ.0) GO TO 500
339*          B(I,M2)=B(I,M2)-B(IR,M2)*B(I,MP1)
340*          IF (I.GT.M) GO TO 510
341*      500  IF (INV.GT.0) GO TO 510
342*          I1=L(L2OFF+I)
343*          X(I1)=B(I,M2)
344*      510  CONTINUE
345*      IF (INV.EQ.2) GO TO 520
346*      X(NMAX+1)=-B(MP1,M2)
347*      OBJ=-B(MOBJ,M2)
348*      C
349*      C *****
350*      C *** STEPS 6 AND 7-UPDATE THE INVERSE AND THE MULTIPLIERS. ***
351*      C *****

```

```

352*      C
353*      520  DO 530 K=1,M
354*          B(IR,K)=B(IR,K)/B(IR,MP1)
355*      530  CONTINUE
356*      DO 550 I=1,M2
357*          IF (I.EQ.IR) GO TO 550
358*          IF (I.EQ.M2.AND.IPHASE.EQ.2) GO TO 550
359*          DO 540 K=1,M
360*              B(I,K)=B(I,K)-B(I,MP1)*B(IR,K)
361*      540  CONTINUE
362*      550  CONTINUE
363*          IF (INV.GT.0) GO TO 850
364*          ITER=ITER+1
365*          L(2)=L(2)+1
366*      C
367*      C * * STORE THE MINIMUM PIVOT.
368*      C
369*          IF (B(IR,MP1).GT.PIVM) GO TO 560
370*          PIVM=B(IR,MP1)
371*          IMP=ITER
372*      560  I1=5+(IPHASE-1)*4
373*      C
374*      C * * CHECK TO PRINT ITERATION SUMMARY.
375*      C
376*          IF (L(I1+1).EQ.0) GO TO 570
377*          IF (ITER/L(I1+1)*L(I1+1).NE.ITER) GO TO 570
378*          WRITE (6,1060) IPHASE,ITER,B(IR,MP1),OBJ,(B(I),IP(I),I=1,2)
379*      C
380*      C * * CHECK TO PRINT ITERATION BASIS.
381*      C
382*      570  IF (L(I1+2).EQ.0) GO TO 220
383*          IF (ITER/L(I1+2)*L(I1+2).NE.ITER) GO TO 220
384*          IX=2
385*          GO TO 720
386*      C
387*      C *****
388*      C *** END OF PHASE. ***
389*      C *****
390*      C
391*      580  IX=3
392*          IF (IPHASE.EQ.2) GO TO 700
393*          IQ=0
394*      C
395*      C * * COMPUTE THE NUMBER OF ARTIFICIALS STILL IN THE BASIS.
396*      C
397*          NA=0
398*          DO 590 J=1,NMAX
399*              IF (L(L1OFF+J).NE.3) GO TO 590
400*              IF (X(J).GT.CEPS) GO TO 600
401*              NA=NA+1
402*      590  CONTINUE
403*          IF (B(MOBJ,M2).GT.(ME+MG)*CEPS) GO TO 600
404*      C
405*      C * * IF THE ARTIFICIAL OBJECTIVE FUNCTION IS ACCEPTABLE BUT ARTIFICIALS
406*      C * * ARE STILL IN THE BASIS, GO TRY TO PIVOT THEM OUT.
407*      C
408*          IF (NA.GT.0) GO TO 620
409*      C

```

```

410*      C * * IF NO ARTIFICIALS ARE IN THE BASIS AND THE ARTIFICIAL OBJECTIVE
411*      C * * FUNCTION IS NOT QUITE ZERO, PRINT WARNING OF ROUND-OFF ERROR.
412*      C
413*          IF (B(MOBJ,M2).GT.EPS) GO TO 610
414*          GO TO 700
415*      C
416*      C * * THE PFOBLEM IS INFEASIBLE.
417*      C
418*      600      L(3)=2
419*              IF (L(MSMX).EQ.0) RETURN
420*              WRITE (6,1090) (ASTAR,I=1,120),B(MOBJ,MP1)
421*              WRITE (6,1120) ITER,IPHASE,OBJ.(ASTAR,I=1,120)
422*              IX=1
423*              GO TO 720
424*      610      IF (L(MSMX).NE.0) WRITE (6,1130) (ASTAR,I=1,240)
425*              L(3)=1
426*              GO TO 700
427*      C
428*      C * * FIND THE ARTIFICIAL STILL IN THE BASIS.
429*      C
430*      620      IQ1=0
431*              DO 630 J=1,NMAX
432*                  J1=L(L1OFF+J)
433*                  IF (J1.EQ.3) IQ1=IQ1+1
434*                  IF (IQ1.GT.IQ) GO TO 650
435*      630          CONTINUE
436*      640          L(3)=4
437*                  IF (L(MSMX).NE.0) WRITE (6,970)
438*                  IX=1
439*                  GO TO 720
440*      C
441*      C * * DETERMINE WHICH BASIC VARIABLE IT IS.
442*      C
443*      650      DO 660 IR=1,M
444*                  I=L(L2OFF+IR)
445*                  IF (I.EQ.J) GO TO 670
446*      660          CONTINUE
447*                  GO TO 640
448*      C
449*      C * * SET ARTIFICIALS IN AT ZERO SWITCH, AND GO TRY TO PIVOT IT OUT.
450*      C
451*      670      ICPSX=1
452*              GO TO 230
453*      C
454*      C * * INCREMENT IQ, THE COUNTER FOR CURRENT NUMBER OF ARTIFICIALS THAT
455*      C * * CAN'T BE PIVOTED OUT.
456*      C
457*      680      IQ=IQ+1
458*              IF (NA.GT.IQ) GO TO 620
459*      C
460*      C * * COUNT AND CHECK THE ARTIFICIALS STILL IN THE BASIS.
461*      C
462*              JS=0
463*              DO 690 I=1,M
464*                  I1=L(L2OFF+I)
465*                  IF (L(L1OFF+I1).NE.3) GO TO 690
466*                  JS=JS+1
467*      C

```

```

468* C * * IF IN AND GT EPSILON, INFEASIBLE. OTHERWISE SET TO 0 AND CONTINUE.
469* C
470* IF (B(I,M2).GT.CEPS) GO TO 600
471* IF (B(I,M2).GT.EPS) L(3)=1
472* X(I1)=0.0
473* L(L1OFF+I1)=5
474* B(I,M2)=0.0
475* 690 CONTINUE
476* IF (JS.EQ.0) GO TO 640
477* IF (L(3).EQ.1) GO TO 610
478* C
479* C *****
480* C *** SUMMARY PRINT. ***
481* C *****
482* C
483* 700 I1=5+(IPHASE-1)*4
484* IF (L(I1+3).EQ.0) GO TO 710
485* WRITE (6,1050) (ASTAR,I=1,120),IPHASE,OBJ,ITER
486* IF (ITER.EQ.0) PIVM=0.0
487* WRITE (6,990) PIVM,IMP
488* IF (IPHASE.EQ.1) WRITE (6,1010) X(NMAX+1)
489* WRITE (6,1030) (ASTAR,I=1,120)
490* C
491* C *****
492* C *** BASIS PRINT. ***
493* C *****
494* C
495* 710 IF (L(I1+4).EQ.0) GO TO 810
496* 720 WRITE (6,1080)
497* DO 730 I2=1,5
498* IB(I2)=0
499* PR(I2)=0.0
500* 730 IP(I2)=0
501* I3=0
502* I4=0
503* IQ1=0
504* DO 800 I2=1,NMAX
505* IF (X(I2).LT.0.0.AND.X(I2).GT.-EPS) GO TO 800
506* IF (I4.EQ.0.L(L1OFF+I2)) GO TO 740
507* IF (I4.EQ.1.AND.0.L(L1OFF+I2).EQ.2) GO TO 740
508* IF (I4.EQ.3.AND.0.L(L1OFF+I2).EQ.5) GO TO 740
509* IF (L(L1OFF+I2).EQ.4) GO TO 780
510* IQ1=1
511* GO TO 790
512* 740 I4=L(L1OFF+I2)
513* IF (I3.EQ.5) GO TO 790
514* I3=I3+1
515* PR(I3)=X(I2)
516* I5=I4+1
517* GO TO (750,760,760,770,780, 770), I5
518* 750 IB(I3)=0 X(
519* IP(I3)=I2
520* GO TO 800
521* 760 IB(I3)=0 S(
522* IP(I3)=I2-N
523* GO TO 800
524* 770 IB(I3)=0 A(
525* IP(I3)=I2-N-ML-MG

```



```

526*          GO TO 800
527*      780      L(3)=4
528*          IX=1
529*          WRITE (6,970)
530*          GO TO 800
531*      790      IF (I3.EQ.0) GO TO 740
532*          WRITE (6,1070) (IB(J),IP(J),PR(J),J=1,I3)
533*          IF (IQ1.EQ.1) WRITE (6,1110)
534*          IQ1=0
535*          I3=0
536*          GO TO 740
537*      800      CONTINUE
538*          IF (I3.GT.0) WRITE (6,1070) (IB(J),IP(J),PR(J),J=1,I3)
539*      810      IF (IX.EQ.1) RETURN
540*          IF (IX.EQ.2) GO TO 220
541*          IF (IPHASE.EQ.2) RETURN
542*          GO TO 210
543*      C
544*      C *****
545*      C *** TOTAL INVERSION ROUTINE. ***
546*      C *****
547*      C
548*      820      J=0
549*          DO 830 I=NT,NMAX
550*              IF (L(L1OFF+I).EQ.2) GO TO 830
551*              J=J+1
552*              L(L2OFF+J)=-1
553*              IF (X(I).LT.0.0) GO TO 830
554*              X(I)=-EPS/100.0
555*              L(L2OFF+J)=I
556*      830      CONTINUE
557*      840      I4=0
558*              JS=0
559*              GO TO 860
560*      C
561*      C * * I4=1 INDICATES THAT A PIVOT WAS PERFORMED.
562*      C
563*      850      I4=1
564*              X(JS)=-EPS/100.0
565*      860      I5=JS
566*              IF (I5.EQ.NMAX) GO TO 880
567*      C
568*      C * * FOR EACH X(I) THAT IS NON-NEGATIVE, GO TRY TO PUT IT IN THE BASIS.
569*      C
570*              I5=I5+1
571*              DO 870 JS=I5,NMAX
572*                  IF (X(JS).GE.0.0) GO TO 220
573*      870      CONTINUE
574*      C
575*      C * * IF AT LEAST ONE PIVOT OPERATION WAS DONE, GO THROUGH X AGAIN.
576*      C
577*      880      IF (I4.EQ.1) GO TO 840
578*              DO 890 J=1,NMAX
579*                  X(J)=-EPS/100.0
580*      C
581*      C * * SET X TO PROPER VALUES AND CHECK FOR FEASIBILITY.
582*      C
583*              I5=0

```

```

584*      DO 910 I=1,M
585*          IF (B(I,M2).LT.-EPS) IS=1
586*          IF (B(I,M2).LT.0.0.AND.B(I,M2).GE.-EPS) B(I,M2)=0.0
587*          I1=L(L2OFF+I)
588*          IF (I1.LT.0) GO TO 900
589*          X(I1)=B(I,M2)
590*          GO TO 910
591*      900      IS=1
592*      910      CONTINUE
593*          IF (IS.EQ.0) GO TO 920
594*      C
595*      C * * INFEASIBLE AFTER INVERSION.
596*      C
597*          L(3)=2
598*          IF (L(MSMX).EQ.0) RETURN
599*          IF (INV.EQ.1) WRITE (6,1040) (ASTAR,I=1,120)
600*          IF (INV.EQ.2) WRITE (6,1000) (ASTAR,I=1,120)
601*          WRITE (6,1120) ITER,IPHASE,OBJ,(ASTAR,I=1,120)
602*          IX=1
603*          GO TO 720
604*      C
605*      C * * IF INVERTING BECAUSE OF INITIAL BASIS, FIX UP L1(.).
606*      C
607*      920      IF (INV.EQ.2) GO TO 960
608*          IS=N+ML+MG+1
609*          IF (IS.GT.NMAX) GO TO 950
610*          DO 930 J=IS,NMAX
611*      930          L(L1OFF+J)=4
612*          IS=0
613*          DO 940 J=1,M
614*              J1=L(L2OFF+J)
615*              IF (L(L1OFF+J1).NE.4) GO TO 940
616*              L(L1OFF+J1)=3
617*              IS=1
618*      940          CONTINUE
619*          IF (IS.EQ.1) GO TO 960
620*      950      INV=0
621*          GO TO 210
622*      960      INV=0
623*          GO TO 240
624*      C
625*      C
626*      C
627*      C
628*      970      FORMAT ('0SYSTEM ERROR- COMPUTATIONAL IMPOSSIBILITY')
629*      980      FORMAT (' EPSILON = 'G13.6,' CAPITAL EPSILON = 'G13.6,' '15,
630*      2 ' NON-ZERO ENTRIES ARE EFFECTIVELY EQUAL TO ZERO.')
631*      990      FORMAT (' *14X'MINIMUM PIVOT WAS'G12.5,' AT ITERATION'14,
632*      2 '.'57X'**)
633*      1000     FORMAT (/////'120A1/' *118X'*/
634*      2 ' * ERROR - THE PROBLEM IS INFEASIBLE. INFEASIBILITY INDICATED
635*      3DURING RE-INVERSION OF THE BASIS MATRIX.'16X'**)
636*      1010     FORMAT (1H+65X'REAL OBJECTIVE FUNCTION ='G15.8)
637*      1020     FORMAT ('0* * WARNING * * SMALL PIVOT ELEMENT AT ITERATION'14,
638*      2 ' OF PHASE'12,' PIVOT ='G14.7)
639*      1030     FORMAT (' *118X'*/'120A1)
640*      1040     FORMAT (/////'120A1/' *118X'*/
641*      2 ' * ERROR - THE VARIABLES SPECIFIED AS COMPRISING AN INITIAL SOL

```

```

642*          3UTION DO NOT FORM A BASIC FEASIBLE SOLUTION.'12X'**)
643* 1050  FORMAT (///// '120A1/' *'118X'*/' *'14X'END OF PHASE'12,
644*      2 ' .    OBJECTIVE FUNCTION ='G18.8,4X'THERE WERE'14,
645*      3 ' ITERATIONS.'17X'**)
646* 1060  FORMAT (///'0PHASE',12,' ITERATION',14,' PIVOT=',G13.6,
647*      2 '    OBJECTIVE FUNCTION='G15.8,' ',A3,I3,') ENTERED THE BASIS.',
648*      3 A3,I3,') LEFT.')
```

```

649* 1070  FORMAT (5(3XA3,I3,')='F12.6))
650* 1080  FORMAT ('0BASIC VARIABLES'/)
651* 1090  FORMAT (///// '120A1/' *'118X'*/
652*      2 ' *  EFROR - THE PROBLEM IS INFEASIBLE. THE CONSTRAINTS ASSOCIATE
653*      3D WITH THE ARTIFICIAL VARIABLES BELOW ARE INCONSISTENT. */
654*      4 ' *  IF NCNE APPEAR, NUMERICAL DIFFICULTIES HAVE BEEN ENCCOUNTERED
655*      5 . THE LARGEST ENTRY IN THE OBJECTIVE FUNCTION ROW',7X,'*/
656*      6 ' *  IS ',G12.5,101X,'**')
657* 1100  FORMAT (///// '120A1/' *'118X'*/
658*      2 ' *  ERROR - THE PROBLEM IS UNBOUNDED. THE VARIABLE X('13,
659*      3 ') CAN ASSUME AN ARBITRARILY LARGE VALUE, THEREBY YIELDING'7X'*/
660*      4 ' *  AN ARBITRARILY LARGE VALUE OF THE OBJECTIVE FUNCTION.'63X'*/
661*      5)
662* 1110  FORMAT (1H )
663* 1120  FORMAT (' *'118X'*/' *  THE ERROR WAS DETECTED AT ITERATION'14,
664*      2 ' OF PHASE'12,
665*      3 ' . AT THAT TIME THE OBJECTIVE FUNCTION VALUE WAS'G15.8,4X'*/
666*      4 ' *  AND THE FCLLOWING VARIABLES WERE BASIC.'77X'*/' *'118X'*/
667*      5 ' '120A1)
668* 1130  FORMAT (///' '120A1/' *'118X'*/
669*      2 ' *  WARNING - COMPUTATIONAL INCCNSISTENCY INDICATED AT THE END O
670*      3F PHASE 1. THE ALGORITHM WILL CONTINUE WITH PHASE 2,'4X'*/
671*      4 ' *  BUT THE USER IS ADVISED TO CRITICIZE THE RESULTS.'67X'*/
672*      5 ' *'118X'*/' '120A1)
673* 1140  FORMAT ('0* *  AT LEAST ONE ELEMENT OF THE RIGHT HAND SIDE COLUMN
674*      2IS LESS THAN ZERO. SUBROUTINE TERMINATES.')
```

```

675*  C
676*      END
```

D.7 Listing of Subroutine PRSMX (PRint SimpleX)

```

1*      SUBROUTINE PRSMX (A,IA,IB,HPAR,IC,IST4,ID,IE,IALT)
2*      DIMENSION A(IA,IB),HPAR(IA,IC),IST4(ID,IE)
3*      DIMENSION L(61)
4*      DO 5 I=1,61
5*      L(I)=A(21,I)
6*      5 CONTINUE
7*      K=1
8*      IF (IALT.NE.0.AND.IALT.NE.1)      K=17
9*      WRITE (6,900) ((I),I= 1,10)
10*     WRITE (6,901) ((IST4(I,1),IST4(I,2)),I= 1,10)
11*     DO 10 I=K,21
12*     IF (I.EQ. 6.OR.I.EQ.11.OR.I.EQ.14) WRITE (6,933)
13*     IF (I.EQ.17.OR.I.EQ.21)          WRITE (6,933)
14*     IF (I.NE.21)
15*     1WRITE (6,902) I,(HPAR(I,J),J=1,5),(A(I,J),J=1,10),I
16*     IF (I.EQ.21) WRITE (6,942) I,(HPAR(I,J),J=1,5),(L(J),J= 1,10),I
17*     10 CONTINUE
18*     WRITE (6,931)
19*     IF (K.EQ.1) WRITE (6,932)
20*     WRITE (6,910) ((I),I=11,24)
21*     WRITE (6,911) ((IST4(I,1),IST4(I,2)),I=11,24)
22*     DO 20 I=K,21
23*     IF (I.EQ. 6.OR.I.EQ.11.OR.I.EQ.14) WRITE (6,933)
24*     IF (I.EQ.17.OR.I.EQ.21)          WRITE (6,933)
25*     IF (I.NE.21)
26*     1WRITE (6,912) I,(A(I,J),J=11,24),I
27*     IF (I.EQ.21) WRITE (6,952) I,(L(J),J=11,24),I
28*     20 CONTINUE
29*     WRITE (6,931)
30*     IF (K.EQ.1) WRITE (6,932)
31*     WRITE (6,910) ((I),I=25,38)
32*     WRITE (6,911) ((IST4(I,1),IST4(I,2)),I=25,38)
33*     DO 30 I=K,21
34*     IF (I.EQ. 6.OR.I.EQ.11.OR.I.EQ.14) WRITE (6,933)
35*     IF (I.EQ.17.OR.I.EQ.21)          WRITE (6,933)
36*     IF (I.NE.21)
37*     1WRITE (6,912) I,(A(I,J),J=25,38),I
38*     IF (I.EQ.21) WRITE (6,952) I,(L(J),J=25,38),I
39*     30 CONTINUE
40*     WRITE (6,931)
41*     IF (K.EQ.1) WRITE (6,932)
42*     WRITE (6,910) ((I),I=39,52)
43*     WRITE (6,911) ((IST4(I,1),IST4(I,2)),I=39,52)
44*     DO 40 I=K,21
45*     IF (I.EQ. 6.OR.I.EQ.11.OR.I.EQ.14) WRITE (6,933)
46*     IF (I.EQ.17.OR.I.EQ.21)          WRITE (6,933)

```

```

47*         IF (I.NE.21)
48*         1WRITE (6,912) I,(A(I,J),J=39,52),I
49*         IF (I.EQ.21) WRITE (6,952) I,(L(J),J=39,52),I
50*         40 CONTINUE
51*         WRITE (6,931)
52*         IF (K.EQ.1) WRITE (6,932)
53*         WRITE (6,920) ((I),I=53,60)
54*         WRITE (6,911) ((IST4(I,1),IST4(I,2)),I=53,60)
55*         DO 50 I=K,21
56*         IF (I.EQ.6.OR.I.EQ.11.OR.I.EQ.14) WRITE (6,933)
57*         IF (I.EQ.17.OR.I.EQ.21) WRITE (6,933)
58*         IF (I.NE.21)
59*         1WRITE (6,922) I,(A(I,J),J=53,61),I
60*         IF (I.EQ.21) WRITE (6,962) I,(L(J),J=53,61),I
61*         50 CONTINUE
62*         RETURN
63*         900 FORMAT ('1PARAMETER',21X,10I8,5X,'ROW')
64*         901 FORMAT (' TABLE 4 INDICES',15X,10(I6,'-',I1)/)
65*         902 FORMAT (I3,'. ',4A6,A2,10F8.1,I8)
66*         910 FORMAT (3X,14I8,5X,'RDW')
67*         911 FORMAT (3X,14(I6,'-',I1)/)
68*         912 FORMAT (I3,14F8.1,I8)
69*         920 FORMAT (3X,8I8,' =      RHS      ROW')
70*         922 FORMAT (I3,8F8.1,' =',F8.1,I8)
71*         931 FORMAT ('0          ')
72*         932 FORMAT ('1          ')
73*         933 FORMAT ('          ')
74*         942 FORMAT (I3,'. ',4A6,A2,11I8)
75*         952 FORMAT (I3,15I8)
76*         962 FORMAT (I3,8I8,' =',2I8)
77*         END

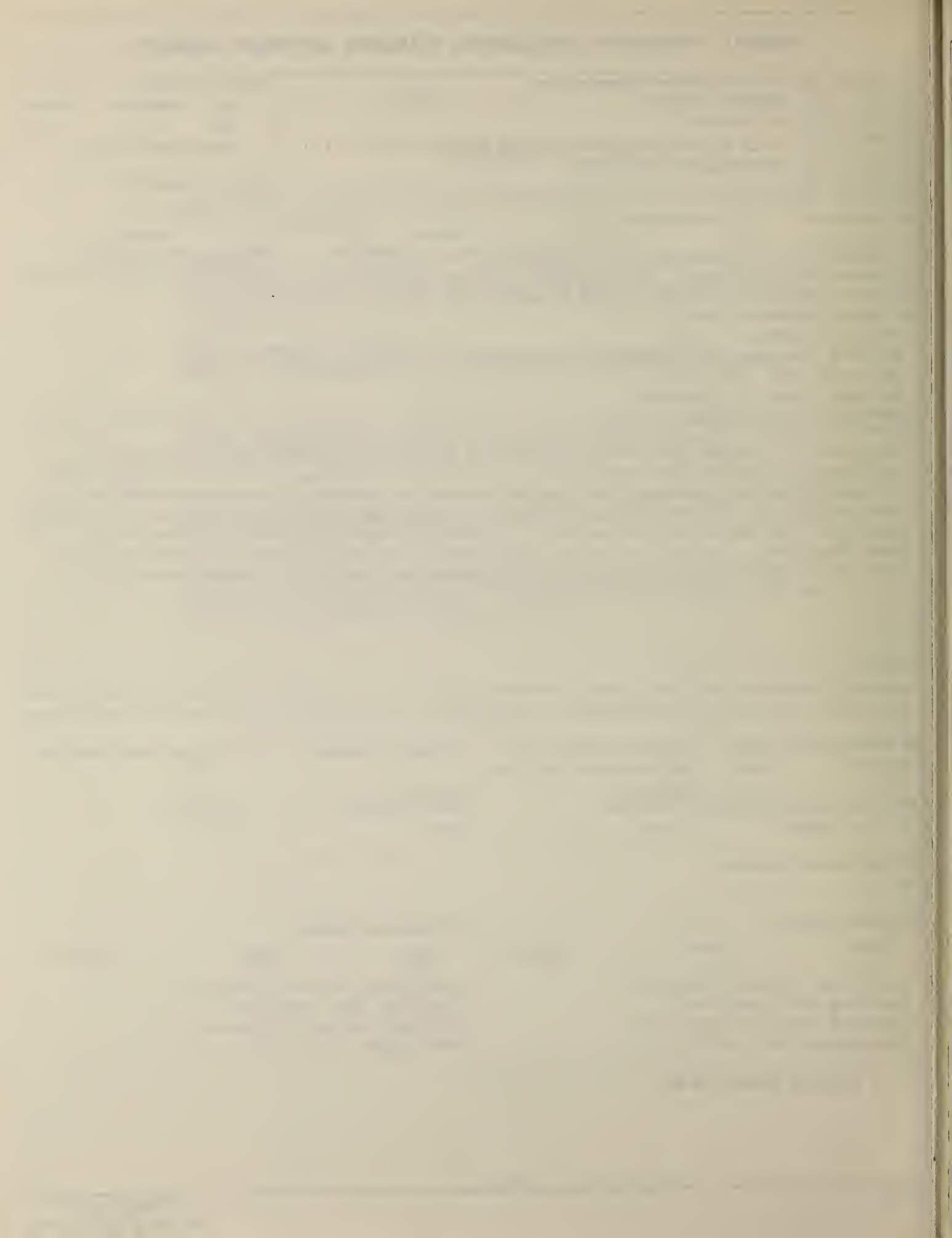
```

REFERENCES

1. Building Construction Cost Data 1978, Robert Snow Means Company, Inc., New York, 1978.
2. Chapman, R. E., P. T. Chen and W. G. Hall, Economic Aspects of Fire Safety in Health Care Facilities: Guidelines for Cost-Effective Retrofits, National Bureau of Standards, NBSIR 79-1902, November 1979.
3. Code for Safety to Life from Fire in Buildings and Structures, National Fire Protection Association, NFPA 101-1973, 1973.
4. Hall, W. G., R. H. F. Jackson and P. B. Saunders, The National Bureau of Standards Linear and Quadratic Programming Subroutines, National Bureau of Standards Report 10695, February 1972.
5. Nelson, H. E. and A. J. Shibe, A System for Fire Safety Evaluation of Health Care Facilities, National Bureau of Standards, NBSIR 78-1555, November 1978.
6. Tragedy of Multiple Death Nursing Home Fires: The Need for a National Commitment to Safety, Select Committee on Aging, Subcommittee on Long-Term Care, September 1976.

FEDERAL INFORMATION PROCESSING STANDARD SOFTWARE SUMMARY

01. Summary date			02. Summary prepared by (Name and Phone)			03. Summary action						
Yr.	Mo.	Day	Robert Chapman			x2278						
7	9	06	05. Software title			New <input type="checkbox"/> Replacement <input type="checkbox"/> Deletion <input type="checkbox"/>						
			FIRE SAFETY EVALUATION SYSTEM RETROFIT COST MINIMIZATION PROCEDURE			Previous Internal Software ID						
						07. Internal Software ID						
04. Software date						NA						
Yr.	Mo.	Day	06. Short title									
7	9	06	NA									
08. Software type			09. Processing mode			10. Application area						
<input type="checkbox"/> Automated Data System <input checked="" type="checkbox"/> Computer Program <input type="checkbox"/> Subroutine/Module			<input type="checkbox"/> Interactive <input checked="" type="checkbox"/> Batch <input type="checkbox"/> Combination			<table border="0" style="width: 100%;"> <tr> <td style="width: 50%; text-align: center;">General</td> <td style="width: 50%; text-align: center;">Specific</td> </tr> <tr> <td> <input type="checkbox"/> Computer Systems Support/Utility <input checked="" type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Bibliographic/Textual </td> <td> <input type="checkbox"/> Management/Business <input type="checkbox"/> Process Control <input type="checkbox"/> Other </td> </tr> </table>			General	Specific	<input type="checkbox"/> Computer Systems Support/Utility <input checked="" type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Bibliographic/Textual	<input type="checkbox"/> Management/Business <input type="checkbox"/> Process Control <input type="checkbox"/> Other
General	Specific											
<input type="checkbox"/> Computer Systems Support/Utility <input checked="" type="checkbox"/> Scientific/Engineering <input type="checkbox"/> Bibliographic/Textual	<input type="checkbox"/> Management/Business <input type="checkbox"/> Process Control <input type="checkbox"/> Other											
11. Submitting organization and address						12. Technical contact(s) and phone						
Applied Economics Group Building Economics and Regulatory Technology Division, CBT, NEL National Bureau of Standards Washington, D.C. 20234						Robert Chapman x2278 William Hall x3855						
13. Narrative												
Program will minimize (optimize) the costs of alternative methods for retrofitting a fire zone in a health care facility so as to achieve compliance to the Life Safety Code. It permits the user to input specific information on potential fire safety retrofits. It also permits the user to preclude, include or change the cost of any potential fire safety retrofit. The least-cost combination of fire safety retrofits for each fire zone is then identified. The cost of strict compliance to the prescriptive provisions of the Life Safety Code as well as from 10 to 20 alternative retrofits are also identified.												
14. Keywords												
Applied Economics; Building Codes; Economic Analysis; Fire Safety; Health Care Facilities; Hospitals; Life Safety; Mathematical Programming; Nursing Homes; Optimization; Renovation												
15. Computer manuf'r and model			16. Computer operating system			17. Programing language(s)		18. Number of source program statements				
UNIVAC 1108A			NBS Computer Services Division			FORTRAN V		1539				
19. Computer memory requirements			20. Tape drives			21. Disk/Drum units		22. Terminals				
14,500 words			None			None		None				
23. Other operational requirements												
None												
24. Software availability					25. Documentation availability							
Available <input checked="" type="checkbox"/> Limited <input type="checkbox"/> In-house only <input type="checkbox"/> Operations Research Division Building 101, Room A-428 National Bureau of Standards Washington, D.C. 20234					Available <input checked="" type="checkbox"/> Inadequate <input type="checkbox"/> In-house only <input type="checkbox"/> Operations Research Division Building 101, Room A428 National Bureau of Standards Washington, D.C. 20234							
26. FOR SUBMITTING ORGANIZATION USE												



U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO. NBSIR 79-1929	2. Gov't. Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE A COMPUTERIZED APPROACH FOR IDENTIFYING COST-EFFECTIVE FIRE SAFETY RETROFITS IN HEALTH CARE FACILITIES		5. Publication Date January 1980	
7. AUTHOR(S) Robert E. Chapman, William G. Hall, Phillip T. Chen		6. Performing Organization Code	
9. PERFORMING ORGANIZATION NAME AND ADDRESS NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, DC 20234		8. Performing Organ. Report No.	
12. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP) Same as #9		10. Project/Task/Work Unit No.	
15. SUPPLEMENTARY NOTES <input checked="" type="checkbox"/> Document describes a computer program; SF-185, FIPS Software Summary, is attached.		11. Contract/Grant No.	
16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.) This study focuses on how a computerized version of the Fire Safety Evaluation System developed by the Center for Fire Research at the National Bureau of Standards can be used to determine equivalence to the Life Safety Code in the least costly manner. This study presents a programmer-oriented discussion of the mathematical, economic and engineering considerations that went into the development of the linear programming algorithm for identifying cost-effective retrofits. Programmer-oriented topics treated in this report include: a discussion of user options; program documentation; format statements; flow charts; sample computer runs; and a complete listing of the computer program.		13. Type of Report & Period Covered Final	
17. KEY WORDS (six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons) Applied economics; building codes; economic analysis; fire safety; health care facilities; hospitals; life safety; mathematical programming; nursing homes; optimization; renovation		14. Sponsoring Agency Code	
18. AVAILABILITY <input checked="" type="checkbox"/> Unlimited <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS <input type="checkbox"/> Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402, SD Stock No. SN003-003- <input checked="" type="checkbox"/> Order From National Technical Information Service (NTIS), Springfield, VA, 22161	19. SECURITY CLASS (THIS REPORT) UNCLASSIFIED	21. NO. OF PRINTED PAGES 124	
	20. SECURITY CLASS (THIS PAGE) UNCLASSIFIED	22. Price \$6.50	

No.	Name	Age
1	John Smith	25
2	Mary Jones	30
3	James Brown	28
4	Elizabeth White	22
5	Robert Green	35
6	Sarah Black	27
7	William Grey	32
8	Ann King	24
9	George Lee	38
10	Margaret Hall	29
11	Richard Young	33
12	Elizabeth Scott	26
13	Thomas Adams	31
14	Mary Baker	23
15	John Wilson	36



