OPTIMAL DESIGN, PROCUREMENT AND SUPPORT OF MULTIPLE REPAIRABLE EQUIPMENT AND LOGISTIC SYSTEMS

by

Thomas P. Moore

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Industrial Engineering and Operations Research

APPROVED:

Wolter J. Fabrycky, Chairman

Donald R. Drew

Marilyn S. Jones

Laurence J. Moore

Hanif D. Sherali

May, 1986

Blacksburg, Virginia

OPTIMAL DESIGN, PROCUREMENT AND SUPPORT OF MULTIPLE REPAIRABLE EQUIPMENT AND LOGISTIC SYSTEMS

OPTIMAL DESIGN, PROCUREMENT AND SUPPORT OF MULTIPLE REPAIRABLE EQUIPMENT AND LOGISTIC SYSTEMS

by

Thomas P. Moore

Wolter J. Fabrycky, Chairman

Industrial Engineering and Operations Research

(ABSTRACT)

A concept for the mathematical modeling of multiple repairable equipment and logistic systems (MREAL systems) is developed. These systems consist of multiple populations of repairable equipment, and their associated design, procurement, maintenance, and supply support. MREAL systems present management and design problems which parallel the management and design of multiple, consumable item inventory systems. However, the MREAL system is more complex since it has a repair component.

The MREAL system concept is described in a classification hierarchy which attempts to categorize the components of such systems. A specific mathematical model (MREAL1) is developed for a subset of these components. Included in MREAL1 are representations of the equipment reliability and maintainability design problem, the maintenance capacity problem, the retirement age problem, and the population size problem, for each of the multiple populations. MREAL1 models the steady state stochastic behavior of the equipment repair facilities using an approximation which is based upon the finite source, multiple server queuing system. System performance measures included in MREAL1 are: the expected MREAL total system life cycle cost (including a shortage cost penalty); the steady state expected number of shortages; the probability of catastrophic failure in each equipment population; and two budget based measures of effectiveness.

Two optimization methods are described for a test problem developed for MREAL1. The first method computes values of the objective function and the constraints for a specified subset of the

solution space. The best feasible solution found is recorded. This method can also examine all possible solutions, or can be used in a manual search. The second optimization method performs an exhaustive enumeration of the combinatorial programming portion of MREAL1, which represents equipment design. For each enumerated design combination, an attempt is made to find the optimal solution to the remaining nonlinear discrete programming problem. A sequential unconstrained minimization technique is used which is based on an augmented Lagrangian penalty function adapted to the integer nature of MREAL1. The unconstrained minimization is performed by a combination of Rosenbrock's search technique, the steepest descent method, and Fibonacci line searches, adapted to the integer nature of the search. Since the model contains many discrete local minima, the sequential unconstrained minimization is repeated from different starting solutions, based upon a heuristic selection procedure. A gradient projection method provides the termination criteria for each unconstrained minimization.

Acknowledgements

This dissertation could not have been completed without the extreme patience and unending encouragement of Professors Wolter Fabrycky and Hanif Sherali. Their guidance and technical insights were crucial to the development of the ideas contained herein. The advice and suggestions received from Professors Donald Drew, Marilyn Jones, and Laurence Moore were also invaluable.

The technical help I received from the many professionals in the Virginia Tech Computing Center made it possible to decipher the intricacies of IBM machinery and documentation, thus enabling the successful writing of the MREAL1 software, and the formatting and printing of this dissertation. I am also grateful to and for their patience and cooperation providing me with more than adequate office and administrative resources in Whittemore Hall, sometimes at their own personal inconvenience.

During the preparation for and the writing of this dissertation, several other people provided me with considerable encouragement. I would like to thank for his support, encouragement, and for taking the time to introduce me to a large number of logisticians who have a strong interest in this research. I must also recognize the ideas and suggestions received from of Rensselaer Polytechnic Institute during the period he served on my advisory committee. Finally, I would like to thank my father, , for his continual support and inspiration, and for taking the time to critically read this dissertation during a very busy Christmas season.

While preparing to write this dissertation, the Department of Industrial Engineering and Operations Research at Virginia Tech employed me to work on US Navy contract N00039-84-C-0346T. I am grateful for the opportunity provided by the Department, and by the Navy, to have worked on problems related to the ones described in this dissertation. I must also thank the Logistics Education Foundation of the Society of Logistics Engineers for awarding me their first Annual Dissertation Proposal Award. I very much appreciate the Foundation's belief that the proposed work was worthy of their support.

Finally, I owe an unrepayable debt of gratitude to my parents, brothers, and sisters-in-law for their emotional support, and encouragement, throughout this long and often trying process. I wish to dedicate this work to my family and friends, and, most especially, to all those who find the courage and strength to struggle through the dissertation process.

Table of Contents

List of Illustrations

Chapter I

INTRODUCTION

The origins of Operations Research (OR) are found in the efforts of the teams of scientists who applied the scientific method to military problems in World War II. According to Hillier and Lieberman, "These teams of scientists were... instrumental in winning the Air Battle of Britain, the Island Campaign in the Pacific, (and) the Battle of the North Atlantic. . . ."[1] Trefethen believes that these successes were largely dependent upon the interdisciplinary[2] composition of the teams and that, "The consideration of a problem in terms of its relationship to an entire operation, making necessary the study of additional, operationally related problems, was, probably, the one achievement which most clearly distinguished World War II operational research from earlier research activities."[3] This interdisciplinary, systems approach[4] to problem solving is part of the reason why operations research is a most appropriate choice of discipline for the study of multiple repairable equipment and logistic systems.

Hillier and Lieberman believe that the development of Operations Research is catalyzed by the problems which result from an organization's ". . .tendency. . . to grow into relatively autonomous

---

1 F.S. Hillier and G.J. Lieberman, Introduction to Operations Research, 3d ed. (San Francisco: Holden-Day, Inc., 1980), p. 2.

2 Now that the work of these scientists has led to the establishment of operations research as a new discipline, whether or not it is still "interdisciplinary" is a question of semantics.

3 F.N. Trefethen, "A History of Operations Research" in Operations Research for Management, ed. J.F. McCloskey and F.N. Trefethen (Baltimore: Johns Hopkins Press, 1954), p. 11.

4 The word "system" refers to a complex collection of elements functioning together to form a whole. The "systems approach" to problem solving refers to a process in which an understanding of the functions and operation of the overall system is obtained. This helps the problem solver determine if a particular solution will help or hinder the goals of the overall system.

empires with their own goals and value systems, thereby losing sight of how their activities and objectives mesh with those of the overall organization."[5] Stein believes that the area of military logistics has split into such autonomous empires (i.e. supply, maintenance, transportation and contracting) and that this has resulted in the overall suboptimization of logistics in US military organizations. Stein names this phenomenon "functional parochialism."[6] By this Stein means that the autonomous empires (the functions) into which military logistics has split do not sufficiently concern themselves with the problems of, and the effects of events in, the other empires. Stein is a proponent of Logistic Systems Theory, which attempts to limit functional parochialism by the application of the systems approach, and by the use of the tools of operations research in the design optimization of logistic systems.

This dissertation deals with the application of the systems approach and the tools and methodologies of operations research to a broad class of logistics problems. This class of problems deals with the design, procurement, support and management of a particular type of system, the multiple repairable equipment and logistic (MREAL) system. The first three chapters of this dissertation define the system, describe some previous models, and propose a specific map which may be useful for understanding the complex components of the system. The remainder of the dissertation describes a specific mathematical model of a subset of the system and two solution techniques. These two optimization techniques have provided solutions to a test problem for the model. Due to the scope and complexity of MREAL systems, this model does not encompass all of the system components defined in the first three chapters. Instead, it encompasses a holistic subset of these components.

Stein is not alone in his perception that logistics, especially the logistics of populations of repairable equipment, has been divided into empires which may lead to the suboptimization of the overall system. The US Department of Defense recognized this problem some time ago when it established

---

[5] Hillier and Lieberman, Introduction to Operations Research, p. 1.

[6] Stein, R.G., "Futuristic Logistics," Logistics Spectrum 19 (Spring 1985), p. 48.

the concept of Integrated Logistics Support (ILS). Integrated Logistics Support has been defined by Blanchard as, "A management function. . . to assure a system that will not only meet performance requirements, but which can be expeditiously and economically supported. . . . ILS assures the integration of the various elements of support. . ."[7]

ILS can be divided into two main processes: planning and execution. In the ILS planning process, tools from operations research and other fields are applied using a systems approach to establish the management policies and procedures, and the equipment and facilities which will best (optimally) provide the logistics (supply, maintenance, transportation, procurement, disposal, etc.) required. The ILS execution process consists of the performance of this management and the creation of these facilities and equipment so that the overall system, not just the logistic components, operate optimally.

Logistics

Logistics is an inconsistently defined term. Coyle and Bardi [1984] offer several definitions of logistics, each having a different perspective. The first definition they offer is given by the Council of Logistics Management, formerly known as the National Council of Physical Distribution Management (NCPDM):

> The definition of LOGISTICS is the process of planning, implementing and controlling the efficient, cost effective flow and storage of raw materials, in-process inventory, finished goods, and related information from point of origin to point of consumption for the purpose of conforming to customer requirements.[8]

Coyle and Bardi also offer a definition of logistics from the Society of Logistics Engineers:.

---

[7] B.S. Blanchard, Logistics Engineering and Management, 3d ed. (Englewood Cliffs, NJ: Prentice-Hall, 1986), p. 298.

[8] Correspondence received on 3 December 1985 from the Council of Logistics Management, 2803 Butterfield Road, Oak Brook, IL 60521.

> The area of support management used throughout the life of the product or system to efficiently utilize resources assuring the adequate consideration of logistics elements during all phases of the life cycle so that timely influence on the system assures an effective approach to resource expenditures.[9]

These perspectives on the definition of logistics fall into two broad categories: the logistics of consumable, non-repairable items, and the logistics of repairable, non-consumable units of equipment. The former category has been studied by many researchers, (Fabrycky and Banks [1966], Hillier and Lieberman [1980], Hadley and Whitin [1963], Venkatesan [1984], Veinott [1966], etc.) and comprises the body of mathematical, statistical and stochastic techniques classically known as inventory theory.

The study and modeling of the logistics of repairable equipment may ultimately be recognized as a second branch of inventory theory. Whereas consumable items are held in order to satisfy customer demands one time only, repairable items are held to satisfy customer demands repeatedly for a certain period of time. From this perspective, repairable equipment logically represents an extension of classical inventory theory, in which the assumption that an inventory item may satisfy a demand exactly one time is relaxed.

When the assumption above is relaxed, the units (repairable equipment) in the inventory become subject to failures which prevent their further use. In this dissertation, the term repairable equipment will refer to any physical entity which is usually repaired when its ability to perform its mission (satisfy a demand) is degraded. The term may refer to vehicles, machinery, computers, buildings, bridges, ships, weapons, radios, electronic gear, or even human beings.[10] An unit of repairable equipment may be an end item itself, or may be a component of some larger item of repairable equipment, which drives the demand for the unit.

---

9 Harrison, G.L., "The Acquisition Logistician," Logistics Spectrum (Spring 1981):11-12. Cited in Coyle and Bardi [28], p. 30.

10 Humans are, to a great extent, repairable and non-consumable. They are often acquired by an organization to satisfy a demand, and they require logistic support.

The focus of this dissertation is on the application of the systems approach of operations research to the ILS planning process for repairable equipment of the mechanical and/or electronic nature. However, the concepts presented herein may also be applicable to systems whose "units of repairable equipment" are not actually repairable, but which can be reused until broken.

The Repairable Equipment and Logistic System

Throughout this dissertation the phrase "repairable equipment and logistic system" will be used when referring a single system which consists of a population of repairable equipment which is designed, acquired, deployed, and managed to meet a demand. The phrase also refers to the related logistic support elements, i.e. the procurement (including design), maintenance, disposal, supply, contracting and management activities which conspire to keep the equipment in the condition necessary to meet the demand. This phrase will be abbreviated as REAL system.

The demand for the repairable equipment in the REAL system must be known to the extent that equipment can be designed and/or procured to meet this demand. It may be unknown to the extent that its quantity, duration, and frequency are not known, or are stochastic, with known or estimated parameters. The population of repairable equipment may consist of identical units of equipment, or it may consist of nearly identical units, or it may consist of dissimilar units of repairable equipment whose only common characteristic is the ability to satisfy, equally well, the demand upon the system.

The definition of a REAL system will be applied in the broadest possible sense. The REAL system has a lifetime which starts when the first decision is made to think of possible ways to satisfy a demand, and ends when the last unit of equipment and the last operating logistic component are retired. The REAL system includes all the units of equipment used to satisfy the demand; the demand itself; the repair facilities or activities which repair the equipment; the support activities which supply the repair facilities; other activities which support the equipment; the equipment and repair facility designs considered in the planning phase; and the policies and procedures which manage the system.

The term "design" will be used in two primary contexts with regard to the REAL system. One context refers to the physical design of the repairable equipment. The phrase "equipment design" will generally be used in this case. The second context refers to the broader choices made by the system manager in establishing the policies which manage the REAL system. This will generally be referred to as "system design."[11]

When performing the system design of a REAL system, only one kind of equipment is usually involved.[12] However, there are two possibilities with regard to design alternatives for the repairable equipment in a REAL system: either a single choice of design is available, or multiple choices are available. The term single equipment, single design (SESD) will refer to the former, while the term single equipment, multiple design (SEMD) will refer to the latter. Both of these terms have parallels in classical inventory theory. For instance, Fabrycky and Banks [1966] refer to the single item, single source (SISS) and the single item, multiple source (SIMS) problems.

---

11 The reader should note that, in this dissertation, the word "system," and the phrase "system design" refer to the REAL system as a whole. In order to avoid the confusion which seems to occur in many sources due to the ambiguous use of the word "system", this word will not be used when referring to the repairable equipment itself.

12 The crucial element in trying to distinguish between a REAL system and an MREAL system is not the similarity between individual units of repairable equipment, but is rather the extent to which this equipment is used to meet the same type of demand.

The Multiple Repairable Equipment and Logistic System

When an organization owns more than one REAL system, (each held to meet a different demand) this collection of systems forms a "supersystem." This supersystem will be called a multiple repairable equipment and logistic system. This name will be abbreviated MREAL system. The organization which possesses an MREAL system has all the attendant problems of ILS planning for each REAL system in its MREAL system, and has the additional problem of choosing how to allocate the organization's resources among the REAL systems.

One of the important multiple decisions in an MREAL system involves the choice of a specific equipment design for each REAL system. The term "design combination" will be used to refer to the set of choices for the physical design of the repairable equipment in each REAL system. There are two possible cases with regard to design combinations: either all of the REAL systems in an MREAL system have a single design alternative, i.e. no choice is possible with regard to equipment design; or more than one alternative exists for at least one REAL system in the MREAL system. The former situation will be referred to as the multiple equipment, single design (MESD) situation, while the latter will be referred to as the multiple equipment, multiple design (MEMD) situation. The latter is the broadest of the possible REAL/MREAL systems. Both of the MESD and MEMD problems have parallels in classical inventory theory. Fabrycky and Banks [1966] refer to the multiple item, single source (MISS) and the multiple item, multiple source (MIMS) problems.

There are many decisions which must be made in order to design and manage an MREAL system. These decisions are described in detail later in this chapter. To model and optimize decisions for MREAL systems requires techniques from such operations research specialties as queuing theory, simulation, and discrete nonlinear optimization. To create the best possible decision support systems for managers[13] of MREAL systems will require techniques and tools from computer science, and probably from decision analysis, artificial intelligence, and expert systems.

The Importance of MREAL Systems

The significance of using the best possible models and optimization techniques on MREAL systems should not be overlooked. Typically, these systems are very expensive to design, acquire and support. In the United States alone, a great deal of money is spent annually just on the acquisition of repairable equipment. In 1983 for example, non-farm business in the US spent $12.87 billion on the acquisition of electrical machinery, $9.77 billion on motor vehicles, and $5.58 billion on aircraft.[14] These amounts refer only to the equipment itself. They do not include the support portion of their respective REAL systems. The maintenance portion of a REAL system adds significantly to the total system life cycle cost. In one US industry, local passenger transportation, $2.17 billion was expended on maintenance in 1982.[15]

If REAL systems are costly, MREAL systems are even more so. Figure 1 on page 9 shows the amounts invested in equipment only by the US Department of Defense (DOD) for various types of military REAL systems. These are just a few of the REAL systems which make up the Department of Defense's MREAL system. Thus, it is very important to use the best possible models and optimization methods on MREAL systems.

---

13 The term system manager will be used throughout this dissertation to refer to the person or group of people who have overall responsibility for and/or exercise control over a REAL or MREAL system.

14 "1983 Expenditures by US Non-farm Business", Survey of Current Business, (Washington, D.C.: US Department of Commerce, January, 1983).

15 "US Passenger Transit Industry", Statistical Abstract of the United States 1984, 104th Edition, (Washington, D.C.: US Department of Commerce, December, 1983).

Type of Equipment	Value (in billions)
Aircraft	$75.771
Ships and Service Craft	$63.759
Vehicles	$17.035
Communications & Electronic Equipment	$ 3.816

Figure 1. Value of Equipment Owned by US Military Services as of September 30, 1981: Extracted from Real and Personal Property (Washington, D.C.: Directorate for Information, Operations and Reports, Washington HQ Services, US Department of Defense, April, 1982), p. 19.

The Difficulties Entailed in Modeling MREAL Systems

While the suboptimal management of MREAL systems clearly puts at risk large of amounts of capital, the modeling and optimization of these systems is neither easy nor inexpensive itself. MREAL systems have a number of strongly interacting decision variables, each of which complicates the modeling process. For example, Hodgson believes that just the reliability and maintainability portions of a REAL system ". . .are separate but related concepts which. . . are intricately involved in system and subsystem interactions. Numerous outside influences contribute to. . . the actual performance that is ultimately achieved."[16]

Hodgson also names some of the decision variables which he feels are significant factors influencing the REAL system. Among these are the physical design characteristics of materials; redundancy; manufacturing process controls and testing; the support factors concerning maintenance personnel and policies; support facilities and equipment; and the level of operator training. Each of these factors may have a significant effect upon two of the key measures of REAL system performance: total MREAL system life cycle cost and the degree to which the imbedded REAL systems meet

[16] Hodgson, G.M., "Reliability and Maintainability in the Air Force," Air Force Journal of Logistics VIII (Summer, 1984), p. 10.

their respective demands. Hodgson recognizes that the best system performance can only be achieved when the possible effects of these factors are considered early in the conceptual definition and design phases of the REAL systems. Unfortunately, there are numerous barriers to building models of MREAL systems which take these considerations into account.

An MREAL system model must make use of forecasts of system costs, equipment performance, repair facility performance, and logistics support performance. There is inherent uncertainty in these forecasts. In addition, some of these forecasts concern the stochastic behavior of equipment failure and repair, and the statistical dependence between system components. Thus there is uncertainty built upon uncertainty, not all of which is statistically describable.

The creation of accurate models is also made difficult by the fact that ILS planning must often be done before the repairable equipment has been built or before production units of the equipment exist. This implies that there will be very little data available from which forecasts of reliability functions or other required data can be developed. (A complaint which is frequently heard in the modeling community is that the required data never available.)

Another difficulty arises when the modeler attempts to measure the performance of an MREAL system. Most system managers recognize several measures of performance, such as system costs, equipment availability, staffing requirements, and such intangibles as convenience and customer/user satisfaction. The system manager wishes to maximize or minimize, as appropriate, each of these performance measures. In addition, the system manager may insist that certain upper or lower limits on these performance measures must not be violated. As a result, it can easily become unclear which measures of performance should be placed in the objective function, and which should be treated as constraints.

A further complication to the modeling process is the fact that MREAL systems often exist under transient conditions. They may have to undergo the transition from a peacetime to a wartime situation, the equipment acquisition may be uneven over time, or the equipment may be subject to severe physical shock. The modeling of such transient events is far more difficult than the modeling of steady state situations.

This enormous complexity within MREAL systems has meant that a wide variety of modeling techniques have been employed in MREAL system models. For example, engineers and economists have wrestled with the issue of when to repair and when to replace (Blanchard and Fabrycky [1981], and Rau [1970]). Logisticians and operations researchers have analyzed supply support for repair parts (Berlucchi [1982], Blanchard [1981], Falk and Rappoport [1980], Frisch [1983] and Sherbrooke [1968]). Reliability and maintenance engineers have examined the stochastic nature of machine failure and repair (Black and Proschan [1959], Brown [1984], Gross, Kahn and Marsh [1977], Lureau [1974], Mirasol [1964] and Morse [1958]).

While the list of research work applicable to MREAL systems is long, the portion which takes a global, systems view of the system is short indeed. While this may be attributable to the complexity of the MREAL system, it is also attributable to two other causes. First, only in the last decade has relatively inexpensive computing power has been available with the speed and capacity to handle models of the size and complexity needed to encompass an entire MREAL system. Secondly, the logistics community has only recently begun to embrace the concepts of systems engineering. In 1985, Newdorf pointed out that, "In the middle 1970's, as the logistics disciplines developed and became quantifiable, communications between logisticians and engineers expanded. The logisticians began to apply systems engineering to the logistics problems."[17] However, as recently as 1984, Stein was discussing the functional parochialism of the logistics community.[18] It is clear that the concepts embodied in systems engineering have not reached some parts of the logistics community, even 10 years after Newdorf says they were first introduced.

---

[17] Newdorf, A.S., "Logistics Trends for the New Electronic Systems," Logistics Spectrum 19(2) (Summer 1985), p. 13.

[18] Stein, p. 48.

Decision Variables in MREAL Systems

Organizations which possess an MREAL system must make daily decisions concerning the controllable variables[19] which affect the performance of the system. The following sections describe some of these decision variables and the questions they raise. The answers to these questions represent the system manager's decisions concerning the design and management of the system. These decisions will materially affect the performance of the MREAL system.

Decision Variable	*Definition*
Procurement Profile	The planned sequence of quantities of repairable equipment to be procured in each period of time during the lifetime of a REAL system. How many units, and when, should equipment be acquired to meet the demand for each REAL system? At what locations should the equipment be placed?
Retirement Age	The age at which repairable equipment is routinely retired and replaced with new equipment of the same design. This age may be stated in terms of calendar time, hours of operation, number of on/off cycles, or, for production equipment, a production quantity.
Maintenance Levels	The number of classes of maintenance, i.e. types of repair facilities, to be operated in the MREAL system. The US Army, for example, has three: organizational maintenance, intermediate maintenance, and depot maintenance.

[19] Throughout this dissertation the variables over which the system manager has direct control will be called decision variables. Those variables which cannot be significantly controlled or influenced by the system manager will be called system parameters.

Level of Repair The maintenance level at which each different type of equipment failure is repaired. This decision variable requires that a taxonomy of failures be defined for each equipment design.

Repair Channels The number of servers or repair channels to be operated for a specific maintenance level in a particular REAL system, i.e. for an equipment design. The locations of these repair channels must also be determined.

Queue Discipline The rule which determines the order in which failed equipment is repaired if all repair channels are occupied. Some standard queue disciplines are: first in, first out (FIFO); last in, first out (LIFO); and shortest estimated processing time (SPT).

Server Crossover The extent to which repair channels may be used to repair equipment from different REAL systems. If a server in REAL system A is free and there is a queue of failed equipment in REAL system B, should the server from A work on the equipment from system B?

Equipment Design The specification of a make, model, manufacturer, and operating characteristics for a repairable equipment to be procured for use in a specific REAL system. The term procurement source may also be used synonymously with the term equipment design. Of particular interest with respect to a design are its procurement cost, annual operating cost, failure profile, repair profile, and the built-in operating envelope.[20]

Failure Profile The age dependent sequence of probability distributions for the time between failures for an equipment design. This is a decision variable only

[20] The term "built-in operating envelope" refers to the extent to which the equipment can limit damage to itself through built-in limit devices. The other way to insure that equipment is operated within its operating envelope is through appropriate operator training.

to the extent that, in the ILS planning phase, the system manager may control the design process enough to be able to influence the failure profile obtained in the repairable equipment which is ultimately procured.

Repair Profile

The age dependent sequence of probability distributions for the time to repair a failed unit of equipment. This is a decision variable only to the extent that the system manager may be able to influence the repair profile through either the design of the repairable equipment or through the design of the repair channels.

Repair/Replace Rule

The decision rules used to determine when to replace a unit of equipment which has experienced catastrophic damage[21] prior to reaching retirement age. These rules may be based on the amount and type of catastrophic damage, and upon the age of the equipment at the time the damage occurred.

Inspection & Testing

Those policies which govern the frequency of inspection and/or testing of the repairable equipment in a REAL system. These policies should dictate the actions to take for each possible outcome of each possible inspection or test. They should also govern what inspections and/or tests are to be performed, and where. An example of an inspection and testing policy is the US Army's engine oil testing program, in which metal particle analysis is conducted periodically on samples of engine oil from vehicles. If the analysis suggests that bearing failure may occur in the near future, an engine rebuild is ordered.

---

21 Catastrophic damage refers to any failure of the repairable equipment which is caused by acts of God, acts of war, or the operation of the equipment outside its operating envelope. This damage may or may not be economically repairable.

Preventive Maintenance Those policies which establish the type and frequency of preventive maintenance (PM) to be performed on units of repairable equipment. Such maintenance is performed even on units which appear to be in perfect operating condition. These policies must include a determination of the location at which the preventive maintenance is performed.

Spare Parts Policy The determination of order quantities, stockage levels, reorder points and stockage locations for the spare parts for the equipment in a REAL system in an MREAL system.

Mechanic Training The determination of the level of training to be given to maintenance personnel in a given REAL system. This decision variable is closely tied to the equipment design choice, since some equipment is user-friendly and idiot-proofed, while other equipment is not.

Operator Training The determination of the level of training to be given to equipment operators in a given REAL system. This decision variable is closely tied to the equipment design choice, as the operating envelope can be equipment-limited at one extreme, and operator-controlled at the other.

R & D Length The level of effort in time, capital, and man-hours which is invested in research and development (R&D) of new repairable equipment. To a certain extent, longer R&D yields better equipment.

The acquisition and management of an MREAL system requires that decisions be made for all of these variables, and for all of the REAL systems in the MREAL system. This decision making involves the classic decision paradox of, "not to decide is to decide", in which values will be set for the decision variables even if the system owner makes no overt decisions at all.

Some System Parameters for the General MREAL System

System parameters are the variables or quantities which affect the MREAL system in some way, but over which the system manager has no direct control. While the specific values of some system parameters are known, the values for others can only be estimated. The following list of system parameter definitions is intended to give the reader an intuitive feeling for the kinds of variables that are system parameters in MREAL systems. The list is not an exhaustive one, nor are the definitions and parameter names necessarily the only way these quantities could be defined in an MREAL system model.

REAL System Demand: The demand for the repairable equipment in a REAL system. This quantity may be known or estimated, constant or time varying, probabilistic or deterministic, and geographically distributed or centrally located. All units of repairable equipment which meet this demand may have to be identical, or they may be very different. For example, if the demand in a REAL system is defined as the requirement to transport 2000 people per day between 18 locations in a city, then perhaps vans, taxi, buses and trucks could be mixed in the REAL system to meet this demand.

Number of Equipment Designs: The number of distinct repairable equipment designs being considered in a REAL system. From the system point of view, the important design characteristics are, among others, cost, reliability, maintainability, operability and complexity (in terms of number of different components).

Unit Shortage Cost: The cost to the system of experiencing one unit-day of shortage. This cost may include the cost of lost customers, the cost of meeting the demand by the use of alternate equipment and/or contractual penalty costs.

Annual Interest Rate: The interest rate which best represents the time value of money for the organization owning the MREAL system.

Equipment Procurement Cost: The average procurement cost of one unit of a specific design of repairable equipment.

Equipment Salvage Value Function: The function representing the future cash value of a unit of repairable equipment, as a function of the age at which the equipment is retired.

Equipment Operating Cost: The cost per unit time of operating one unit of repairable equipment. This cost typically includes the cost of fuel, power, scheduled replacement parts, and the operator and crew for routine operations.

Equipment Procurement Lead Time: Since most repairable equipment which might be the subject of a REAL system model is expensive and complex, it is also likely to have a significant procurement lead time. This may be important with respect to the establishment of the REAL system, or it may only be important when the catastrophic loss of equipment occurs.

Repair Channel Procurement Cost: The cost of procuring the tools and test kits, and the cost of personnel training for one repair channel. This may also include a pro-rated portion of the land and building in which the channel resides.

Repair Channel Salvage Value: This is the future cash value of a repair channel. It may be a constant, if the age at which the repair channel will be retired is known. Or it may be a function of this retirement age, which is, in turn, a decision variable.

Repair Channel Operating Cost: The cost per unit time of operating the repair channel. This cost includes power, scheduled tool replacement, consumable parts and material, and personnel salaries and benefits.

Mechanic Training Cost: The cost of hiring and training one mechanic to work in a repair channel in a REAL system.

Operator Training Cost: The cost of hiring and training one equipment operator for a REAL system.

Annual Mechanic Cost: The recurring annual cost of pay, benefits and lost work hours for one mechanic in a REAL system repair channel.

Annual Operator Cost: The recurring annual cost of pay, benefits and lost work hours for one equipment operator in a REAL system.

Equipment Storage Space: The space available in which the repairable equipment must occasionally or permanently reside.

Failure and Repair Breakdown Structure: Once a particular design is chosen for the repairable equipment in a REAL system, the repair actions and parts required to repair a failure become system parameters. Unfortunately, the types of failures and the corresponding repair times and parts may not be known. In this case, the frequencies of each type of failure must be estimated. This estimation is best done from historical data on the unscheduled maintenance performed on a population of repairable equipment over a period of time. Since this type of data is not available when the REAL system is being designed, the estimation of the failure and repair structure must initially be done from design specifications, experience and intuition.

Spare Parts Storage Space: The space available in the locations at which spare parts must be available for the maintenance of the repairable equipment.

Storage Space Cost: The cost of space to store spare parts, repairable equipment, and/or maintenance equipment.

Spare Parts Costs: The procurement cost of each spare part and component to be used in the scheduled or unscheduled maintenance of the repairable equipment in a REAL system.

Spare Part Procurement Lead Time: The length of time, either deterministic or probabilistic, which will elapse between the placing of an order for a spare part, and the arrival, from the supplier, of that order at its stockage site.

Transportation Times: This is a rather broad class of possible system parameters which deal with the various possible transportation activities which may occur in an MREAL system. When scheduled or unscheduled maintenance must be performed on a unit of repairable equipment, the unit must often be transported some distance to the repair facility. When one repair location needs a particular spare part which is only available at some other location, this part must be transported some distance. If the MREAL system allows the sharing of repair facilities or mechanics between REAL systems, then there may be requirements for the transportation of units of equipment or mechanics between repair facilities. There may be other transportation processes which must be modeled in a particular MREAL system.

Test Set Cost: The cost of a test set for the repairable equipment in a REAL system. There may be several different types of test sets, each with its own cost.

Preventive Maintenance Action Cost: The cost of a particular preventive maintenance action on a unit of repairable equipment in a specific REAL system.

Procurement Budget: The total amount of procurement capital available for all REAL systems in an MREAL system in the current year, and in as many future years as can be reasonably forecasted. These procurement funds may be spent on the repairable equipment, the repair channels, personnel training, spare parts inventories, and inspection and testing equipment.

Operating Budget: The total amount of operating capital available for all REAL systems in an MREAL system in the current year, and in as many future years as can be reasonably forecasted.

These operating funds may be spent on scheduled and unscheduled maintenance actions, salaries and benefits, and energy and other consumable supplies for the repairable equipment.

Cost of Equipment Shortages and Degradation: The cost, in terms of money or some other measure, of an outright shortage of a unit of equipment. This also includes the costs which are incurred whenever a REAL system must operate with degraded equipment, i.e. a unit or units which are partially operational.

Some System Performance Measures

Before it is possible to control or manage any system, some measure or measures of effectiveness must be defined. These system performance measures are necessary to judge the effectiveness of the management decisions made for the system. There are many measures of effectiveness which have been used to judge portions of REAL and MREAL systems. Blanchard [1981] and Rau [1970] provide definitions and descriptions of most of the measures of effectiveness commonly used in the logistics field.

While this dissertation uses both the phrase "measure of effectiveness" and the phrase "system performance measure," the latter is the better of the two. It gives a stronger sense that the measure should have meaning for the system as a whole. It makes little sense to judge the performance of a REAL system by, for example, the reliability of the repairable equipment because this neglects the contributions to overall system performance made by the repair channels, the number of spare units of repairable equipment procured, the maintainability characteristics of the equipment, etc.

The following section describes some possible system performance measures for MREAL systems. An attempt has been made to present these measures from the holistic, systems perspective.

System Cost: First and foremost among all system performance measure is the economic one: cost.[22] System costs may be defined in many different ways. Life cycle cost is one the most popular measures of system cost. Life cycle costing recognizes the time value of money by using one of the cash flow analysis techniques of engineering economy[23] to estimate the total system life cycle cost for the repairable equipment and its associated logistics. The most important characteristic of a system cost performance measure is that it represent all of the significant costs (and revenues, as applicable) involved in the ownership of a REAL system over its lifetime. Such costs may include the cost of the repairable equipment, the spare parts, the personnel, training, repair facilities, transportation, etc.

Budgetary Measures: Since most MREAL systems exist in organizations with limited financial resources, one important measure of system performance is whether or not the system can be owned and managed within these financial constraints. It is very easy to do the initial acquisition of the repairable equipment and some of the associated logistic support activities, only to discover that completing the acquisition of the system will exceed the available financial resources. This may force the system manager to adopt a suboptimal system design, which could end up costing the organization much more in the long run.

Manning: Most MREAL systems require a certain number of people to operate and manage the system. The system manager or managers, equipment operators, mechanics, and logistics personnel are frequently needed. While manning can often be combined with system cost measures by costing each individual required, some MREAL systems may exist in organizations which are under manpower limitations. Thus the performance of a particular system design may have to be measured with respect to manning requirements.

---

[22] Or net profit or loss, if the MREAL system belongs to a commercial organization.

[23] Also known as economic analysis, or sometimes as present worth analysis.

System Equipment Availability: A number of system performance measures fall in this category. Most of these measure attempt in some way to deal with the stochastic nature of the equipment failure and repair processes. Some, such as the traditional inherent, achieved, and operational availabilities described by Blanchard [1981], represent characteristics which are inherent to the repairable equipment, but without regard for the demand upon the REAL system. Other measures, such as expected shortages, described by Moore and Fabrycky [1985], examine the ability of a system design to meet the demand. This represents the characteristics inherent to the repairable equipment, the system design, and the demand itself. The advantage of such a measure (over the traditional measures of availability) is that it better represents the occasional queuing behavior which leaves a REAL system very short of equipment. The disadvantage is that it requires the demand to be known or estimated.

Another measure of system availability described by Moore and Fabrycky [1985] is the probability of catastrophic system failure. This measure represents the probability of experiencing more than a certain large number of shortages. This measure even more directly represents the behavior of the tail of the probability distribution on shortages, which is a measure of system availability.

Sherbrooke [1968] describes another example of a system availability measure in his paper on the US Air Force's METRIC logistics model. The objective function of this model minimizes the expected number of spare part backorders.

Objectives of the Dissertation

The first objective of this dissertation is to define a class of systems called Multiple Repairable Equipment and Logistic (MREAL) systems. This system class will be defined in such a way as to be representative of those systems involving populations of repairable equipment owned by individuals, business and government organizations. The definition will show that a significant characteristic of MREAL systems is the interaction between equipment design characteristics and the logistic support requirement of the system.

The second objective of this dissertation is to show that the study of MREAL systems can provide models with significant potential for providing information which may lead to improved economic performance of systems of repairable equipment and their associated logistic support activities.

The dissertation's third objective is to show that the study of MREAL systems contains elements of previous work in inventory systems, queuing systems, logistics systems, and in reliability, availability and maintainability (RAM). This objective is accomplished in part by the creation of a logical classification hierarchy for MREAL systems in which the significant components of such a system are described.

The fourth objective of this dissertation is to show that a holistic model of an MREAL system can be constructed and an estimate of the optimal solution for the model found at a reasonable cost.[24] The optimization makes use of OR methodologies such as discrete nonlinear programming, stochastic processes and engineering economy. This dissertation accomplishes this objective by creating such a model and an associated optimization technique at a cost which is less than $10,000.

The final objective of this dissertation is to describe and enumerate recommendations for future research on MREAL systems. It is hoped that these recommendations will provide worthwhile and accurate guidance for the future improvement of mathematical representations of MREAL systems.

---

[24] Models of MREAL systems attempt to capture the essence of the decision problems faced by the system manager in designing, procuring, supporting, and managing the system. The "optimal solution" is thus the set of values of the decision variables which yield the best performance of the system. System performance is, of course, judged by the system manager. It should be noted that models of complex systems such as this have a tendency to help the system manager better understand the behavior of the system, and to help to refine the system manager's concept of system performance. Thus, such models are not intended to simply mimic the intuitive decision making processes in the system manager's mind, but rather they are intended to capture the essence of reality as it pertains to the actual MREAL system, thereby improving the system manager's ability to manage the system.

While it is possible that MREAL systems can be optimized using existing, mathematical models, it will be difficult to demonstrate this until holistic models are available as performance benchmarks. The specific model developed in this dissertation, MREAL1, does not claim to be the ultimate holistic model of the general MREAL system. Instead, it is intended as a step in the direction toward the creation of such models using the principles of operations research.

A Look Ahead

In Chapter II, a proposed classification hierarchy for MREAL systems is given. The hierarchy is intended as a guide for modeler's of MREAL systems, as a catalyst for discussion about the structure of real MREAL systems, and to classify the kinds of modeling work which has been done on REAL systems.

Chapter III reviews a cross section of the literature pertaining to REAL systems. It also includes the small amount of published research which views REAL systems from the holistic viewpoint. The material in Chapter IV describes a specific model, called MREAL1, of an MREAL system. The structure of this model is extracted from the structure of the classification hierarchy, thus demonstrating the hierarchy's usefulness. This chapter also contains a description of the specific test problem for which the optimization model was created. Chapter V contains a description of the literature of discrete nonlinear programming. It also contains a description of the test problem enumerator program and the test problem optimization program. In Chapter VI the results and conclusions from the modeling and optimization process are discussed. The final chapter, Chapter VII, contains the recommendations for future research.

Appendix A contains a glossary of terms and symbols dealing with MREAL system models and their optimization. Appendix B contains the FORTRAN code for the MREAL1 Enumerator.

Appendix C contains the FORTRAN code for the MREAL1 Optimizer. Samples of the output of both programs are contained in Appendices D and E respectively.

Chapter II

A CLASSIFICATION HIERARCHY FOR MREAL SYSTEMS

The realization that nearly all logistics modeling has proceeded from some particular functional logistic viewpoint, with the result that current models and their respective solutions may be sub-optimal with respect to the overall MREAL system, lead to the consideration of the MREAL system in its entirety and to an attempt to develop a complete representation of the associated problems. The result of this effort is a classification hierarchy which suggests a holistic structure to the MREAL system. It encompasses, as subproblems, much of the existing logistics modeling work, and many of the functional areas of logistics.

No claim is made that this hierarchy is the ultimate description of the structure of an MREAL system. Instead, this hierarchy is a first attempt to specifically identify and describe this structure. It is hoped that the hierarchy will lead to better and more useful models by:

1. encouraging model creators to think about structural elements of the MREAL system which are missing from their models, and how these elements may affect their model;

2. making the model user and the model creator more aware of the potential interactions between structural elements of MREAL systems;

3. helping the model user understand the place a specific model has in the overall structure of an MREAL system, and to help show how a specific model may contribute to the overall ILS planning process for the MREAL system.

A diagram of the classification hierarchy for MREAL systems is shown in Figure 2 on page 28. The hierarchy consists of five levels of systems and problems. The diagram shows the general relationships between these systems and problems. While the diagram clearly reminds the reader of the type and structure of problems which must be solved for MREAL systems, it does not clearly show one important item: time. The diagram could be considered to be an infinitesimal time slice of an MREAL system, if it were possible to draw the diagram with time being the third coordinate axis, coming out of the paper. What is important to note when viewing the diagram is that models of MREAL systems must eventually take time dynamics into account.

Level One

Models which deal with the optimal allocation of resources for collections of REAL systems comprise Level One of the hierarchy. The key issue to be resolved at this level concerns the fact that different REAL systems satisfy demands which may differ in importance. In light of these differences, a good MREAL system model should help manage the resource allocations which must be made between the REAL systems in the MREAL system. Such limited resources may for example include procurement budgets, operating budgets, personnel or real property.

Level Two

The second level of the hierarchy consists of the individual REAL systems, each representing a different major end item of repairable equipment. Each has a demand determined by the mission of the organization and environment in which the equipment functions. At this level, models must

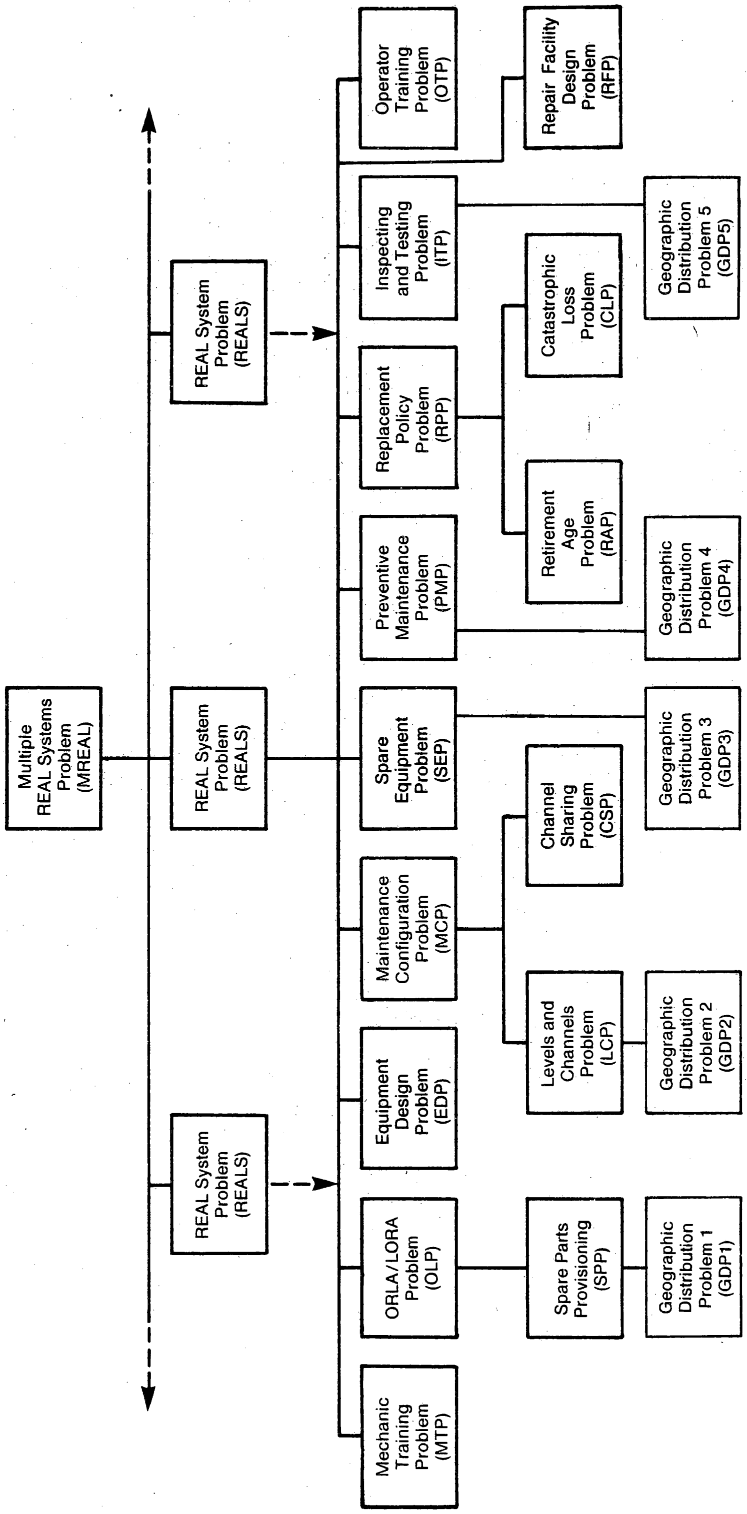

Figure 2. Classification Hierarchy for MREAL Systems

bring together all of the problems and concerns from the levels below, so that the overall REAL system may be optimized, or so that the REAL system's contributions to the MREAL system model can be represented.

Level Three

A holistic REAL system model integrates all elements of the REAL system which can affect performance and which cannot be modeled separately from the rest of the system. Such a model is an important tool for the allocation of resources within a REAL system, between such elements as spare end items, maintenance channels, physical design, and operator training. There are ten such major elements to be considered at this level. All of these elements must be considered in the ILS planning process for a REAL system. These ten elements are defined below:

Mechanic Training Problem (MTP): Determine the type of training for mechanics, as it pertains to the effect of training on repair time probability distributions.

Optimal Level of Repair/Level of Repair Analysis Problem (OLP): Determine the maintenance level at which each type of equipment failure will be repaired; determine the location at which each type of failure will be repaired.

Equipment Design Problem (EDP): Determine the design, design specification, and/or procurement source for the repairable equipment which yields the best performance for the overall REAL system (or MREAL system).

Maintenance Configuration Problem (MCP): Determine how the maintenance facilities in the REAL system will be configured.

Spare Equipment Problem (SEP): Determine the number of units of repairable, prime mission equipment (including spares) to be procured to meet the demand; determine when this equipment will be procured, i.e. determine the procurement profile.

Preventive Maintenance Policy (PMP): Determine what preventive maintenance will be performed on the repairable equipment; determine the frequency of such maintenance.

Replacement Policy Problem (RPP): Determine the circumstances under which repairable equipment will be replaced.

Inspection and Testing Policy (ITP): Determine the types of inspection and testing to be performed; determine the frequency with which this inspection and testing will be performed; determine what, if any, inspection and testing will be performed during scheduled preventive maintenance.

Repair Facility Design Problem (RFD): Determine the design of tools, repair stands, test and diagnosis sets, parts storage racks, tool racks, and recordkeeping procedures as they may affect equipment repair time distributions; determine the repair facility workstation layout; determine the level of resources to spend on the research and development process for new designs of repairable equipment in the REAL system.

Operator Training Problem (OTP): Determine the level of training required for operators of repairable equipment especially with respect to the equipment design, the built-in operating envelope limit devices, and inspection and test equipment.

Level Four

The fourth level of the hierarchy contains subelements from several Level Three elements. These subelements are either problems whose solutions depend upon the policies established for their respective Level Three element, or they are significant problems which must be solved before the policies for their respective Level Three element can be established. The Level Four subelements are:

Spare Parts Provisioning (SPP): Determine the initial stockage quantities, the reorder rules, the replenishment quantities, and the stockage locations for each spare part; determine the quantities to stock at each repair channel of each maintenance level.[25]

Levels and Channels Problem (LCP): Determine the number of levels of maintenance (organizational, intermediate, depot, etc.) to operate in the MREAL system; determine the number of repair channels to operate at each maintenance level; determine the number of mechanics to assign to each repair channel at each maintenance level for the REAL system; determine the queuing and service disciplines to be used in each repair channel.

Channel Sharing Problem (CSP): Determine the extent to and conditions under which equipment from each REAL system will be maintained at the repair channels of the other REAL systems.[26]

Retirement Age Problem (RAP): Determine the criteria (calendar age, operating hours, etc.) to be used for retiring repairable equipment; determine when to routinely retire repairable equipment.

Catastrophic Loss Problem (CLP): Determine which types of catastrophic damage will lead to repair of the equipment and which types will lead to replacement.

---

[25] Note that some readers may take issue with putting the initial provisioning and the reorder/replenishment activities together in the same subelement. While traditionally these two management activities have been performed using different sets of input information, they are both inherently the same kind of activity. Thus their inclusion together.

[26] The repair channels for different REAL systems are often located in the same facilities and buildings.

Level Five

The fifth and final level of the hierarchy contains the elements of the REAL system which deal with location problems. Five of the elements of Level Three and Level Four have significant subproblems concerning the geographic location of equipment and/or facilities. The Level Five elements are:

Geographic Distribution Problem 1 (GDP1): Determine the stockage locations of spare and repair parts, for each type of part needed in the REAL system.

Geographic Distribution Problem 2 (GDP2): Determine the geographic location of each repair channel at each level of maintenance in the REAL system.

Geographic Distribution Problem 3 (GDP3): Determine the geographic distribution of the repairable equipment itself, especially the spare or maintenance float units.

Geographic Distribution Problem 4 (GDP4): Determine the locations at which each type of preventive maintenance will be performed on the repairable equipment.

Geographic Distribution Problem 5 (GDP5): Determine the locations at which each type of inspection and each type of testing will be performed on the repairable equipment.

The hierarchy is a conceptual aid intended to help identify the significant policies which affect the cost and logistic performance of an MREAL system in the face of limited resources. Thus those elements which affect life cycle costs and those elements which affect system dependability and availability must be included. It should be noted, however, that the classification hierarchy is a generic one for the general MREAL system. Actual MREAL systems may have somewhat different structural elements. It is also possible that the author has inadvertently omitted some important element or other of the generic MREAL system.

The classification hierarchy should assist the REAL and MREAL system modeler in visualizing the structure of the system being modeled. It should also help generate questions which should be asked when trying to understand the interactions of a particular system. For instance, the Mechanic Training Problem (MTP) leads one to think about such questions as: What impact does a mechanic's workload have on time to repair? How many mechanics should be assigned to a repair channel? What impact does a change in the level of a mechanic's training have on time to repair? How do we model the trade-off between training level and repairable equipment design?[27] How does the Preventive Maintenance Problem interact with Mechanic Training Problem?

Similarly, the Inspection and Testing Problem (ITP) element encourages thinking about such questions as: What routine tests should be performed? How often? At what maintenance level? What action should be taken for each test outcome? What effect will the inspection and testing policies have on system availability? On the Mechanic Training Problem?

Each of the elements and subelements of the hierarchy should generate questions like these: What policies should be set for this element? How will these policies affect MREAL system costs, availabilities and dependabilities? How will these policies affect the other elements of the system structure? The holistic models being developed for MREAL systems are the tools intended to help answer these questions.

The examination of MREAL system structure via the hierarchy also illustrates one very important point: holistic MREAL system models can serve as benchmarks to help determine the adequacy of existing non-holistic models. It is a distinct possibility that the suboptimal solutions currently obtained from existing models are very close to the true optimal solutions. If this is so, then it may be possible to dispense with the formulation and optimization of large, expensive holistic models, in favor of the less expensive, piecemeal models currently in use. However, it will be difficult, if

[27] It seems reasonable that changes in the design of the repairable equipment can lead to: a) changes in the level of training required for a mechanic; b) changes in the repair time distributions. Some equipment is just easier and faster to repair than others.

not impossible, to determine whether or not this is true unless holistic models are constructed to serve as benchmarks. Without such holistic models as benchmarks, we may never be quite sure how well existing models are performing.

Chapter III

LITERATURE REVIEW

The complex nature of the MREAL system has impeded development of holistic models which encompass the policy decisions required for appropriate system management. While no truly holistic MREAL system model has been found in the literature, a number of partial models of REAL systems (as opposed to MREAL systems) have appeared in the literature. A partial REAL system model is one which does not encompass all of the components of a REAL system but does have a holistic aspect, in that it includes three or more significant components of the REAL system.

Of particular note in the review of the literature is the paper by Nahmias [1981]. In this paper, Nahmias reviews a large number of modeling techniques related to REAL systems. He divides the models into two broad categories: models based on techniques from classical inventory theory; and models based upon queueing theory. In the former group lie models primarily aimed at solving the spare parts provisioning problem and its associated geographic distribution problem. These models also sometimes touch upon the spare equipment problem in that they may try to determine optimal equipment population sizes given known repair capacities. Nahmias describes work by Sherbrooke [1968] (the METRIC model), Sherbrooke [1971], Muckstadt [1973], and Porteus and Lansdowne [1974]

In the category of queueing based models, Nahmias describes work by Mirasol [1964], Gross, Kahn, and Marsh [1977], Gross and Ince [1978, 1981], and Lureau [1974]. Nahmias points out that the queueing approaches have two important advantages over the models based upon classical inventory theory: they do not have to assume that there are an infinite number of servers; and they may treat the number of servers as a decision variable.

Geisler and Murrie [1979] also provide a very useful review of models related to partial REAL systems. Their paper is limited, however, to a survey of military models for aircraft logistics. They identify three categories of models: design models; spares allocation models; and base-aircraft operations models. The design models deal with aircraft design characteristics such as component reliability, built-in redundancy and component utilization and importance. The spares allocation models deal with the spare parts provisioning problem and its associated geographic distribution problem. The base-aircraft operations models include aspects of the spare parts provisioning problem, the maintenance capacity problem, the equipment design problem, and the spare equipment problem.

Partial REAL System Models

In Hart [1971], and Fabrycky and Hart [1975] a model was developed and optimized which represented a single REAL system by the total number of units of repairable equipment, N , the number of repair channels, M, (a single level of maintenance was assumed), and the retirement age, R, of the equipment. The time to repair and the time to failure were both assumed to be exponentially distributed. Mean failure and repair rates were allowed to change with equipment age. The stochastic behavior was then approximated using the finite queuing model and an average of the mean times between failure (MTBFs) and an average of the mean times to repair (MTTRs). The model minimizes expected annual equivalent costs of prime equipment, repair facilities and shortages. The model was applied to several different designs of equipment. Hinger [1973] extended the Hart-Fabrycky model to include two levels of maintenance and constraints on population size,

N, and retirement age, R. The optimization technique for both models was the sectioning search of Friedman and Savage [1947].

Mackliet [1984] captured the basic ideas of the Hart-Fabrycky model in a Monte Carlo simulation. While this simulation model does not include an optimization procedure, it does allow for representation of the transient behavior of the REAL system. Mackliet calls his model the Availability Simulation (ASIM). ASIM computes point and confidence interval estimators for inherent availability when given values for the decision variables in the model. These decision variables include: repairable equipment population size as a function of time; the number of repair channels; mean time between failure (MTBF) and mean time to repair (MTTR) profiles; and, indirectly, retirement age.

ASIM will handle a variety of probability distributions for the breakdown, and repair processes. It will also model the transient REAL System behavior due to the catastrophic failure of equipment or the changing of the population size. However, it cannot optimize over the decision variables. It can only provide good estimates of one REAL system performance measure (inherent availability) given a set of values for the decision variables.

In 1984, Moore enhanced the Hart-Fabrycky model by showing that an exact, steady state probability distribution for the number of shortages could be derived. This distribution was then used to compute the steady state expected number of shortages and the probability of one or more shortages.[28] Moore applied the finite queuing model as described in the following paragraph.

The Hart-Fabrycky model assumes that there exists a population of N units of repairable equipment, and M identical repair channels to repair the equipment when it fails. The population consists of R classes of equipment, all with the same number of units of equipment. The classes are similar in that they have exponentially distributed failure times, exponentially distributed times to

[28] Fabrycky, W.J., Malmborg, C.J., Moore, T.P. and Brammer,K.W., "The Repairable Equipment Population System Demonstrator," User's Guide and program for IBM-PC, VPI&SU Report for US Navy Contract N00039-84-C-0346T, 1984.

repair, and they can all perform the activities necessary to meet the demand. The classes are dissimilar in that they have slightly different values for the parameter of the exponential distributions (MTBF and MTTR). There is a demand for the equipment, d. Of course $N \geq d$ and $1 \leq M \leq N$. The Hart-Fabrycky model assumes that the finite queuing model, which requires that all N units of repairable equipment be identical, can sufficiently approximate the stochastic behavior of the situation described above. Let

$$\hat{\lambda} \equiv \frac{1}{\sum_{i=1}^{R} \frac{MTBF_i}{R}}$$

and

$$\hat{\mu} \equiv \frac{1}{\sum_{i=1}^{R} \frac{MTTR_i}{R}}$$

where $MTBF_i$ is the mean time between failure for the i th class of equipment and $MTTR_i$ is the mean time to repair for the ith class of equipment. $\hat{\lambda}$ is the time averages failure rate, and $\hat{\mu}$ is the time averaged repair rate. Note that $M\hat{\mu}$ is the maximum possible repair rate for the system. When all units of equipment are working the arrival rate of failed equipment at the repair facility is $N\hat{\lambda}$. When exactly n units of equipment are in a failed status, this arrival rate becomes $(N - n)\hat{\lambda}$. We define P_n to be the steady state probability that exactly n units are in a failed status. From the Birth-Death stochastic process, in the steady state, P_n must satisfy the following equations:

$$
\begin{aligned}
N\hat{\lambda}P_0 &= \hat{\mu}P_1 \\
N\hat{\lambda}P_0 + 2\hat{\mu}P_2 &= (\hat{\mu} + (N-1)\hat{\lambda})P_1 \\
(N-1)\hat{\lambda}P_1 + 3\hat{\mu}P_3 &= (2\hat{\mu} + (N-2)\hat{\lambda})P_2 \\
. &= . \\
. &= . \\
. &= . \\
(N-M+2)\hat{\lambda}P_{M-2} + M\hat{\mu}P_M &= [(M-1)\hat{\mu} + (N-M+1)\hat{\lambda}]P_{M-1} \\
. &= . \\
. &= . \\
. &= . \\
2\hat{\lambda}P_{N-2} + M\hat{\mu}P_N &= (M\hat{\mu} + \hat{\lambda})P_{n-1} \\
\hat{\lambda}P_{N-1} &= M\hat{\mu}P_N
\end{aligned}
$$

and

$$\sum_{n=0}^{N} P_n = 1$$

Setting $\rho \equiv \left[\frac{\hat{\lambda}}{\hat{\mu}}\right]$ and solving for P_0 we obtain:

$$P_0 = \frac{1}{\displaystyle\sum_{k=0}^{M} \frac{N!\rho^k}{(N-k)!\,k!} + \sum_{k=M+1}^{N} \frac{N!\rho^k}{(N-k)!\,M!\,M^{(k-M)}}}$$

Substituting P_0 back into the previous equations yields the following expressions:

$$P_n = \frac{\dfrac{N!\,\rho^n}{(N-n)!\,n!}}{\displaystyle\sum_{k=0}^{M} \frac{N!\rho^k}{(N-k)!\,k!} + \sum_{k=M+1}^{N} \frac{N!\rho^k}{(N-k)!\,M!\,M^{(k-M)}}}$$

for $n \in \{0,1,2, \ldots, M\}$ and

$$P_n = \frac{\dfrac{N!\,\rho^n}{(N-n)!\,M!\,M^{n-M}}}{\sum_{k=0}^{M} \dfrac{N!\rho^k}{(N-k)!\,k!} + \sum_{k=M+1}^{N} \dfrac{N!\rho^k}{(N-k)!\,M!\,M^{(k-M)}}}$$

for $n \in \{M+1, M+2, \dots, N\}$.

This derivation of $\{P_0, P_1, \dots, P_N\}$ is a straightforward application of the finite source (machine repair) model with multiple service channels described by Palm [1947]. However, this probability distribution is for the random variable F, the number of units in failed status. The Hart-Fabrycky model requires the probability distribution on the random variable S. This random variable represents the number of units of repairable equipment which the REAL system is short of.

A shortage does not occur until $F = n > N - d$. Consequently, the steady state probability that $S = 0$ is given by:

$$\mathrm{P}(S = 0) = \sum_{n=0}^{N-d} P_n$$

The remainder of the probability distribution on S is given by:

$$\mathrm{P}(S = k) = P_{N-d+k}$$

The desired steady state expected number of unit shortages, $\mathrm{E}(S)$, is given by:

$$\mathrm{E}(S) = \sum_{k=0}^{d} k\, P_{N-d+k}$$

And, the desired steady state probability of having 1 or more shortages is given by

$$\mathrm{P}(S \geq 1) = \sum_{k=1}^{d} P_{N-d+k}$$

Fabrycky, Malmborg, Moore and Brammer [1984] created an interactive program for the IBM personal computer using the enhanced version of the Hart-Fabrycky model described above. The program allows a user to quickly obtain REAL system total expected annual equivalent cost for user-given choices of equipment population size, N, repair channels, M, and retirement age, R, and system parameters such as estimated unit cost, per unit-year shortage cost, MTBF-age profile, and MTTR-age profile, etc. Since the behavior of REAL systems is not intuitively obvious to the casual observer, the software has significant potential as an educational tool for managers and owners of REAL systems. It can also be used to compare alternate equipment designs (with respect to N, M, and R), and to illustrate the trade-offs between design and logistics.

Geisler and Murrie [1979] describe a model called the Logistics Composite Model (LCOM). This model is a very large Monte Carlo simulation which examines the operation of a single Air Force base. It simulates the interactions between maintenance, supply, and aircraft operations. LCOM can be tailored to a particular Air Force base and includes the capability of simulating failures and repair activities on aircraft components down to the level of those components which can be removed by first level maintenance personnel on the flight line. LCOM produces information on a large number of performance measures including equipment utilization, manpower utilization, maintenance activity, and mission performance from the viewpoint of aircraft availability. This is one of the more complete partial REAL system models found in the search of the literature. It should be noted, however, that no optimization is performed by the model.

Models of Maintenance Configuration and Spare Equipment

Of considerable interest to the logistics community are models which address the trade-off between adding repair facilities and adding spare equipment (maintenance float) to a REAL system. The difficulty in developing such models arises from the stochastic nature of the interactions between repair facilities, spare equipment and cost performance in a REAL system. In his book, Rau [1970] describes the basic probabilistic and stochastic tools for analyzing the failure of repairable and non-repairable equipment, and the repair of repairable equipment. Only in the last chapter does he discuss the queuing theory necessary to begin the analysis of simple REAL systems.

Perhaps the earliest significant work on the stochastic behavior of REAL systems was performed by Mirasol [1964] in his Ph.D. thesis. The steady state behavior of a REAL system is examined using a multi-stage circular queueing network which represents the passage of units from the maintenance float inventory, to operationally ready (in use) status, to failed (awaiting transportation, in transit, awaiting parts, awaiting repair, or in repair), to repaired (awaiting transportation or in transit). The model concentrates on representing the number of spare units of equipment, the number of repair facilities and the provisioning of spare parts.

Mirasol uses a stage of parallel queues in the circular network to represent the failure of different types of parts. Each queue represents the supply channel for a specific repair part (or assembly). Arrival at a particular "awaiting spare parts" queue is governed by a multinomial distribution. To make the analysis tractable, Mirasol assumes that all random processes are Poisson (exponential time to failure and exponential time to repair), and that the repair processes are identical for each type of part failure. He defines an availability measure called strategic availability, the product of the system unavailability rate and the mean duration of the system unavailability.

The optimization procedure used by Mirasol is best characterized as an intelligent search, with the objective of obtaining a strategic availability versus total system cost curve. To obtain this result, Mirasol assumes that strategic availability is a monotone function of cost. He points out that the assumptions made are quite restrictive and sometimes unrealistic. Mirasol suggests the use of simulation as a way to build a more realistic model.

Mirasol's methodology is too restrictive to be of great use on real REAL systems. However, his efforts are a significant first exploration of the use of queueing networks to represent the stochastic behavior of REAL systems.

In 1977, Gross, Kahn, and Marsh formulated a model of the REAL system which involved the number of repair channels, M, as a function of time, and the number of spare units of repairable equipment (which equals the population, N, minus the demand), as a function of time. The model allows demand to change with time. The objective function of the Gross-Kahn-Marsh model is simpler than in the models previously mentioned. Salvage values and most operating costs are neglected. Shortage costs are also left out of the objective function. Instead, an availability constraint is given for each time period.

The probabilistic aspect of the REAL system is modeled by Gross, Kahn and Marsh using the finite source queuing approximation. They define the failure rate for each time period in terms of the rate from the previous period, modified by the rate for new units (acquired for the current period) and the rate for units repaired during the last period, assuming that repairs improve failure rates. Gross, Kahn and Marsh require the repair rate to be identical for all units during a particular time period, but allow it to change from period to period.

The heuristic used by Gross, Kahn and Marsh to solve their integer, non-linear stochastic mathematical program involves a rather simple two dimensional step search. The number of repair facilities whose procurement cost is roughly equal to the cost of a unit of spare equipment is found, assuming such spare units are more expensive than repair facilities and the current solution is incremented by either one spare unit or by the equivalent number of repair facilities. The availability which results from each of these two new solutions is computed, with the solution yielding the best availability being selected. This becomes the new current solution. When feasibility is achieved, a smaller, local search is initiated which hopefully locates an cost improved, feasible solution.

The authors also discuss the use of a queueing network to represent removal, transportation and repair times. They show how this model can easily be solved as a series of M/M/j queues, assuming that the population of repairable equipment is infinite. They also give some evidence that using this infinite source model yields a larger number of spares and servers, giving a higher equipment availability in the REAL system, but probably leading to a higher cost.

In Gross and Kahn [1976] and Gross and Ince [1981] investigated the errors which occur as a result of estimating the behavior of the equipment repair queues by using mean failure rates and mean mean times between failures (MTBF's) in the finite source queueing model. They conclude that these errors tend to be small for most realistic equipment repair problems.

While Gross, Kahn and Marsh suggested a heuristic procedure for optimizing their model, Falk and Rappoport [1980] developed an optimization algorithm for the same basic model. This algorithm is based on an integer branch and bound scheme which generates subproblems that can be solved by an enumeration scheme to define the feasible region. Then dynamic programming is applied.

A Model for Maintenance Configuration and Spare Parts Provisioning

Brammer [1985] developed a model for determining the period by period maintenance requirements for a REAL system. The model estimates the number of repair channels, the repair channel staffing and the spare parts demand for the system in each period using a Markov analysis approach. This model forecasts spare parts demand based upon historical spare part failure data.

A Model of Geographic Distribution Problem 3

Porteus and Lansdowne [1974] describe work which is aimed primarily at determining the optimal number of spare units of equipment in the situation where the equipment is distributed between several different locations. Each location has exactly one repair facility (one channel), but the model presumes that repair rates can be controlled (and different) at each location. Thus one of the decision variables deals with maintenance capacity via the presumed ability to allocate maintenance resources so that any MTTR value can be controlled at each location.

The model also allows for multiple types of failures, providing for a multinomial distribution governing failure type, given that a failure has occurred. It is assumed that MTTR can be controlled for each type of failure.

The objective of the model is to minimize the expected number of shortages subject to a budget constraint via the appropriate choice of the number of spares and the setting of MTTR. The advantage of this kind of formulation is that it skirts the problem of measuring the negative impact of a shortage upon the REAL system. The multiple REAL system problem cannot be solved unless the shortage impact can be dealt with. Consequently, the Porteus and Lansdowne model cannot be adapted to the multiple REAL system problem.

Miscellaneous Models

While not specifically addressed to the modeling of REAL systems, Cohen, Pierskalla and Nahmias [1980] describe an inventory theory model which suggests that the applications of inventory theory to REAL systems should be further explored. The model computes optimal order quantities for a periodic review inventory system in which a constant proportion of stock issued to meet demand in each period feeds back into the inventory after a fixed number of periods. In addition, a fraction of the stock in inventory decays each period.

This model has many of the elements necessary for a REAL system model: decayed stock could possibly represent planned retirement of equipment; stock returned to the inventory after a fixed number of periods could be interpreted as equipment undergoing repair in which the repair time is a constant; a second such loop of returning stock could represent equipment which has successfully met a demand and now awaits a new demand; and the stock used could represent the catastrophic loss of units of equipment. There may be merit in future investigation of the application of classical inventory theory to REAL systems.

Since MREAL systems are the repairable item counterpart to the classical inventory system involving consumable items, an examination of some of the classical inventory theory is useful. Fabrycky and Banks [1966], for example, describe an optimization procedure for deterministic procurement-inventory systems with multiple items and multiple procurement sources. This work is of particular interest because it is an example of the use of dynamic programming to solve the multiple item, multiple source (MIMS) inventory problem, and because it uses total system cost as the system performance measure in the objective function.

The model described by Fabrycky and Banks attempts to minimize total procurement, holding and shortage costs, subject to a constraint on total warehouse space. It is this constraint which links the item/source combinations and prevents a solution consisting optimal solutions for each individual item.

In their formulation, the maximum possible number of units of an item which would ever have to be warehoused is denoted by I^* . Fabrycky and Banks find total system cost as a function of the decision variables, Q^* and L^*, and I^*. They then use the calculus to find the values of the decision variables which minimize total system cost. The resulting Q^* and L^* are functions of I^*.

The authors then compute total system cost, and values for Q^* and L^* for all feasible values of I^* for all source/item combinations. This is an exhaustive search of the feasible region for the single item, single source (SISS) and the single item, multiple source (SIMS) problems. This search is aided by treating the integer variables Q^* and L^* as real and using the calculus to obtain the unconstrained minimum as a function of I^*.

Once these computations have been performed for all source/item combinations, the optimal values of Q^*, L^* and total cost have been obtained for every source/item combination, plus for every feasible value of I^*. In reality, fixing values for the decision variables Q^* and L^* leads to a particular I^*. Consequently, different sources for the same item can be compared by examining the cost performance of each source at the same I^* value. This yields an optimal source, Q^* , L^* and total system cost for an item at a fixed I^*. Repeating this procedure for all feasible values of I^* yields what Fabrycky and Banks call the condensed cost functions for each item.

The condensed cost function represents the best possible cost performance for a given item as a function of I^*. The final part of the Fabrycky/Banks procedure uses dynamic programming to find the minimum cost combination of items which meets the warehousing constraint.

One shortcoming of the paper is that it does not show that the total system cost function is convex. In this case, this could be shown by demonstrating that this function has continuous second partial derivatives and that its Hessian matrix is positive definite or semidefinite for all non-negative Q^* and L^*.

An examination of these and other models, and the modeling needs of the logistics community, reveals a number of shortcomings: existing models make numerous assumptions which may not hold for real REAL systems; probabilistic system behavior is modeled approximately; the transient behavior of a system can be modeled, but only at considerable cost and often not in a form which lends itself to optimization (i.e. by simulation); the logistic characteristics of equipment designs are addressed by the models as an afterthought or not at all. It is also clear that many of the existing models were developed within the scope of a single functional area of logistics. According to Stein [1985], this "functional parochialism" is responsible for considerable MREAL system sub-optimization. It appears that "functional parochialism" may have also discouraged the development of truly holistic optimization models

Chapter IV

THE MREAL1 MODEL

The process of creating the MREAL system classification hierarchy leads to the visualization of a number of possible holistic models of an MREAL system. It also clearly shows that a truly holistic model, one encompassing all of the problems and subproblems would be complex indeed. Given that existing models do not come close to encompassing all of the components of an MREAL system, it would be an impossible task to create such a model in this dissertation. Instead, a more conservative approach was adopted. Accordingly four elements and subelements of REAL systems were chosen to be modeled. These REAL system models were then linked appropriately to represent the behavior of an MREAL system. This model is called MREAL1, to indicate that it is a first effort at holistically modeling an MREAL system. The four elements of the REAL system which are represented in model MREAL1 come from:

1. the Spare Equipment Problem (SEP)

2. the Retirement Age Problem (RAP)

3. the Levels and Channels Problem (LCP)

4. the Equipment Design Problem (EDP).

For each REAL system in the MREAL system, the model attempts to answer the following questions simultaneously. These questions are essentially asking about the optimal values of the decision variables.

1. How much capital should be invested in each REAL system?

2. How many units of repairable equipment should be procured in each REAL system?

3. How many repair channels should be provided for each REAL system?

4. At what age should the equipment in each REAL system be routinely retired?

5. From what source should the equipment be procured for each REAL system?[29]

The answers to these questions must be found with respect to some measure or measures of MREAL system performance. Perhaps the two most important such measures of performance are the total cost of the MREAL system, and the system-wide equipment availability achieved in each individual REAL system. However, there are certainly other measures of system performance which are important to a system manager.

This leads to a problem for the modeler. If traditional mathematical programming is used to formulate the model, then only one measure of performance can be directly included in the objective function to be maximized or minimized. Several ways around this problem have been found, each with some disadvantages. One method is to put the "most important" performance measure into the objective function. The remaining performance measures are then relegated to becoming constraints in which some minimum or maximum level of performance is required. The main disadvantage of this method is that some system managers cannot determine a required performance level

[29] Alternately, what MTBF- and MTTR-profiles should be designed into the prime equipment of each REAL system?

for these performance measures. Instead, the system manager demands the best possible performance of these measures, too.

Another method is to find some way to combine performance measures indirectly in a single objective function. This requires some implicit or explicit weighting to represent the relative importance of each performance measure. Since some measures of system performance quantify very different things (such as cost versus availability), some system managers do not feel comfortable with this type of objective function. They feel as though the modeler is "comparing apples to oranges."

A third solution to this problem is to specify a series of goals for each system performance measure.Then one of the versions of goal programming is used to minimize the deviations from the goals, simultaneously or sequentially. The disadvantage of this method is that some measures of performance are antithetical. For example, an MREAL system may be such that any improvement in system availability will increase total system cost.

Measures of System Performance in MREAL1

The MREAL1 model uses a combination of the first two methods described above for dealing with the measurement of system performance. The measures used in this model are the total MREAL system cost, and two measures of REAL system availability: expected system unavailability[30] achieved in the *i*th REAL system using the *j*th equipment design, $S_{UNA_{ij}}$, and steady state cat-

[30] System unavailability in this dissertation refers to the number of units of repairable equipment which a REAL system is short. For example, if a REAL system has a population of 12 units of repairable equipment to meet a demand of 10, and 4 units are in repair, then the REAL system is short 2 units of equipment.

astrophic unavailability[31] in the ith REAL system, using the jth equipment design, $C_{UNA_{ij}}(sh_i)$.[32] Two measures of budgetary performance are also included.

The most important performance measure is considered to be, for the purposes of model MREAL1, total system cost. Consequently, this performance measure forms the basis for the objective function of model MREAL1. However, one of the measures of REAL system availability, $S_{UNA_{ij}}$, is combined with the system cost components when the total cost is computed. This is achieved by computing the expected cost of shortages using the value of $S_{UNA_{ij}}$ obtained from the specific choices of decision variables. In addition both of the system availability performance measures appear in the constraints of the MREAL1 model.

The objective function of MREAL1 contains total annual equivalent costs of:

1. procurement and operation of the repairable equipment

2. procurement and operation of the repair facilities

3. expected number of shortage-years which result system availability which is achieved by a particular choice of decision variable values.

In order to have a fair comparison between procurement costs and operating costs for the repairable equipment, the procurement cost is annualized over the equipment life, R_i, in each REAL system. The time value of money and the estimated salvage value at retirement, $f_{ij}(R_i)$, are used in this computation. $f_{ij}(R_i)$ is computed as a linear function of retirement age. The first portion of the MREAL1 objective function thus accounts for the annualized procurement cost and operating cost for each unit of repairable equipment in each REAL system.

---

31 Catastrophic unavailability refers to the probability of a certain large number of shortages occurring. In model MREAL1, this probability is a steady state probability.

32 Note that Appendix A contains a glossary of mathematical notation and symbols.

The procurement cost of a repair channel in a given REAL system is also annualized so that it can be combined with annual operating costs. It is assumed, however, that the retirement age of a repair channel is not a policy issue, but is set by considerations outside the MREAL system. The operating costs of the repair channels are fixed per channel, per year, but may be different for each REAL system. Repair channel costs are assumed not to vary as the number of units in for repair increases or decreases. In reality, mechanics tend to work faster as the repair load increases. Further, system managers often assign extra mechanics, or authorize overtime when a repair facility is backlogged. The assumption that this will not affect repair channel costs is one which should be relaxed in future models.

The total shortage cost for each REAL system is computed by estimating the mean number of shortage-years experienced per year for a given combination of decision variable values and multiplying by the cost which is incurred by a shortage-year. These costs are then summed over all REAL systems in the MREAL system. The assumption is made that the amount of shortage-years per year occurring in a REAL system has no effect upon the number of shortage-years per year occurring in any other REAL system, i.e. the shortage behaviors of the REAL systems are assumed to be independent.

Model MREAL1 assumes that the equipment in each REAL system fails according to an exponential time to failure distribution. It also assumes that equipment repair time is exponentially distributed. Each unit of equipment, however, may have a different mean time between failure (MTBF), and also a different mean time to repair (MTTR). This characteristic is used to represent the changes in MTBF and MTTR which typically occur as the equipment gets older. The stochastic behavior of the MREAL1 model can be represented by a two-stage cyclic queue with as many classes of customers as there are "age" groups. The MREAL1 model, in particular, designates such a class for each year-group over the life of the equipment in each REAL system. However, little modification of the model is required to achieve the level of detail in which each individual unit of equipment has its own MTBF and MTTR, slightly different from all the other units'.

Gross and Ince [1981] have shown how the stochastic behavior of a two-stage cyclic queue can be modeled under the first-come, first-served queuing discipline assumed by MREAL1. Unfortunately, as the number of "age" groups increases, the resources necessary to perform the required computations increases dramatically, making the use of this particular model impractical for the moment. Gross and Ince show that the classical finite source queuing model can approximate the true steady state probabilities of having $S = k$ shortages for the two-stage cyclic queue under certain circumstances. MREAL1 uses one of these approximations (time averaging) to estimate the probability distribution for the shortages random variable, S.

The are several links between the REAL systems modeled in the MREAL1 model. One such link is the respective value given the costs of shortage-years. The model automatically allocates more resources to force the number of shortage-years lower in those REAL systems with higher shortage-year costs. Another link involves budget limitations on the organization's procurement and operating expenditures. Two constraints are used to represent these limitations.

The first constraint limits the total annual budget for the operating costs of both the repairable equipment and the repair channels, for all the REAL systems. A second constraint represents a separate limitation on the amount spent each year to replace retiring equipment. The model assumes that every REAL system is in a steady state in which a fixed fraction, $\frac{1}{R_i}$ (where R_i is the retirement age of equipment in the ith REAL system) of the repairable equipment is retired every year.

Availability constraints of two types are included for each individual REAL system. One specifies the maximum allowable mean system unavailability, $S_{UNA_{ij}}$, as a percent of demand, in the ith REAL system when equipment design j is chosen. This constraint involves a statistical expectation which depends upon the stochastic behavior of the two-stage cyclic queue. The constraint value obtained for a particular set of decision variable values is the steady state mean number of shortages, in units, divided by the demand. The second constraint specifies a maximum allowable probability of ever exceeding a specified number of shortages, sh_i, in the ith REAL system. Whereas the first constraint pertains to the average stochastic behavior of a REAL system, the second constraint deals with the potential extremes of the stochastic behavior of the system.

It should be noted that these two measures of system unavailability are not traditional measures of unavailability. Such traditional measures are focused upon the unit of repairable equipment rather than the system of equipment. For example, Rau [1970] defines inherent availability as:

$$A_i = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}}$$

These mean times, however, refer to the means of the failure and repair processes of the repairable equipment itself. Thus inherent availability does not take into account the possibility that a population of repairable equipment can, at random times, overload the REAL system's repair facilities. If a REAL system does not have a dedicated repair facility for each unit of repairable equipment, inherent availability will over-estimate the long term fraction of time which the equipment is actually available to perform its mission.

The inaccuracy caused by the definition of inherent availability is well known. To try to overcome this problem, operational availability is defined. For instance, Kline [1982] gives the following definition:

$$A_o = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR} + \text{MLDT}}$$

where MLDT is the mean logistics downtime. MLDT is intended to represent all of the time the repairable equipment is not undergoing repair, but is not available to perform its mission. Blanchard [1986] defines operational availability as:

$$A_o = \frac{\text{MTBF}}{\text{MTBF} + \text{MDT}}$$

where MDT is the maintenance downtime. MDT is intended to represent all of the time the repairable equipment is unavailable to perform it mission. Neither of these definitions exactly represents the stochastic behavior of the REAL system, because they do not precisely reflect the queuing behavior of a REAL system, especially under transient conditions.

Inherent availability and operational availability are, nonetheless, popular measures of the availability performance of a REAL system. This is most appropriate when the exact demand for the repairable equipment cannot be determined to a relatively precise number of units. In this situation, with the demand unknown, the shortage random variable, S, carries no meaning. The system manager is thus forced to try to manage the REAL system so as to maximize operational availability, rather than to minimize shortages.

If, however, the demand for repairable equipment is known, deterministically or stochastically, then it is better to examine availability from the point of view of the REAL system's ability to meet the demand. Graves and Keilson [1983], for example, define system availability as the probability that this demand can be met using a particular set of values of the decision variables for the REAL system. They also define system persistence as the time between system failures, where a system failure is the event that the REAL system cannot meet all of the demand.

When attempting to measure REAL system availability, Graves and Keilson's concept of system availability could lead to a "system operational availability" of the following sort:

$$A_s = \frac{\text{MTBSF}}{\text{MTBSF} + \text{MSDT}}$$

where MTBSF is the mean time between system failures and MSDT is mean system downtime. The system is, of course, only up when enough units of repairable equipment are available to meet all of the system demand. Unfortunately, this definition of system operational availability is less useful than might be expected. The demand upon a REAL system is very often a soft number, in that the mission of a REAL system can usually be accomplished even when the system is short a few units of equipment. More often, as a larger and larger number of shortages occurs, system costs increase gradually, but at an increasing rate. The implication of this is that system "failures" are not so cut and dried. To adequately manage the REAL system, the system manager needs to know something about the probability distribution on shortages[33] and about how costs increase as shortages increase.

The system availability measures used in this dissertation attempt to address these issues. The shortage cost included in the objective function is based on a linearly rising cost of shortages, and uses the exact steady state probability distribution for shortages.[34] The catastrophic system unavailability measure is included as a constraint in recognition of the fact that shortage costs do not usually remain linear as the number of shortages approaches the demand itself. At some point, costs increase dramatically, i.e., the REAL system fails. This constraint is thus an attempt to force the model to deliver a solution in which such a catastrophe is averted.

Assumptions Made in Model MREAL1

The tractability of any mathematical model depends upon the simplifying assumptions made. Since MREAL1 is an initial attempt to create a holistic model of an MREAL system, a fair number of restrictive assumptions have been made to insure that the model could be formulated. These assumptions are listed below:

1. MREAL1 assumes that the MREAL system under study has achieved steady state in both the stochastic sense, and the procurement sense. The system is not undergoing a procurement buildup of the repairable equipment, nor is the system undergoing retirement.

[33] Or, equivalently, the system manager must know something about the distribution on system availability, with respect to the demand.

[34] The number of shortages is equal to the demand minus the number of units available or to zero, whichever is greater.

2. The time between failures is exponentially distributed for all of the repairable equipment in each REAL system in the MREAL system. However, each equipment year group in a particular REAL system may have a mean time between failure different from the mean MTBF, $1/\hat{\lambda}$, for the entire population of equipment in that REAL system. The model assumes that the dispersion of the year group MTBF values about $1/\hat{\lambda}$ is relatively small.

3. The time to repair is exponentially distributed for all of the repairable equipment in each REAL system in the MREAL system. However, each equipment year group in a particular REAL system may have an MTTR which is different from the mean MTTR, $1/\hat{\mu}$, for the entire population of equipment in that REAL system. The model assumes that the dispersion of the year group MTTR values about $1/\hat{\mu}$ for the population is relatively small.

4. It is assumed that the population size ranges involved with each REAL system are small enough to avoid representation difficulties on the computing machine used to solve the model.

5. The finite source queuing model using time averaging is assumed to approximate the behavior of the actual two-stage cyclic queue in each REAL system.

6. All repair channels in a particular REAL system are assumed to be identical.

7. All equipment failures are repaired at the same level of maintenance in a particular REAL system.

8. Administrative and transportation times for failed equipment are assumed to be contained in the time to repair.

9. The first come, first served queuing discipline is assumed to be in effect at all times in every REAL system.

10. No sharing of idle repair channel resources is allowed between REAL systems.

11. The demand upon each REAL system is assumed to be deterministic and known.

12. Shortages occurring in one REAL system have no effect upon the shortages which may occur in any other REAL system.

13. The MTBF and MTTR values for each year group and equipment design in each REAL system are known.

14. All equipment costs are known.

15. The maximum possible retirement age is known for each equipment design in each REAL system.

16. All repair channel costs and the repair channel lives are known.

17. Repair channel costs are estimated so that no cost savings are available from the sharing of physical facilities by the repair channels of two or more REAL systems.

The Multiple REAL Systems Model 1 (MREAL1)

Three kinds of notation appear in the mathematical formulation of MREAL1: decision variables; system parameters; and system performance measures. Decision variables represent system operating policies and are capitalized in the notation. System parameters are costs and other factors which cannot be directly controlled by system owners. System parameters appear in lower case. System performance measures represent the cost and availability quantities which result from the system parameters and policies. Performance measures appear in the mixed case. See the Glossary in Appendix A for the definitions of these symbols.

For a given combination of equipment designs in each REAL system, model MREAL1 formulates the following problem:

$$\text{Minimize} \quad f(\overline{N}, \overline{M}, \overline{R}) = \sum_{i=1}^{n_{reals}} (C_{p_i}(N_i, R_i) + C_{f_i}(M_i) + C_{s_i}(N_i, M_i, R_i))$$

subject to:

$$\sum_{i=1}^{n_{reals}} C_{pr_i} = \sum_{i=1}^{n_{reals}} \left\lceil \frac{N_i}{R_i} \right\rceil p_{ij_i} \le b_{pr}$$

$$\sum_{i=1}^{n_{reals}} C_{op_i} = \sum_{i=1}^{n_{reals}} (N_i\, o_{ij_i} + M_i\, ro_i) \le b_{op}$$

$$S_{UNA\; ij_i}(N_i, M_i, R_i) \le b_{UNA_i} \qquad \text{for all } \; i \in \{1, 2, \dots, n_{reals}\}$$

and

$$C_{UNA\; ij_i}(N_i, M_i, R_i \,|\, sh_i) \le b_{CAT_i} \qquad \text{for all } \; i \in \{1, 2, \dots, n_{reals}\}$$

where C_{p_i}, C_{f_i}, C_{s_i}, $S_{UNA\; ij_i}$ and $C_{UNA\; ij_i}(sh_i)$ are the functions of N_i, M_i, and R_i as described below, j_i is the design chosen in the ith REAL system, and sh_i is the number of shortages considered to be catastrophic in the ith REAL system. The problem is also subject to the following explicit constraints:

$$N_i \ge d_i \quad \text{for all } i \in \{1, 2, \dots, n_{reals}\}$$

$$M_i \ge 1 \quad \text{for all } i \in \{1, 2, \dots, n_{reals}\}$$

$$N_i - M_i \ge 0 \quad \text{for all } i \in \{1, 2, \dots, n_{reals}\}$$

$$R_i \ge 1 \quad \text{for all } i \in \{1, 2, \dots, n_{reals}\}$$

$$R_i \le rmax_{ij_i} \quad \text{for all } i \in \{1, 2, \dots, n_{reals}\}$$

The Equipment Population Cost

The annualized cost of procurement and operations for the repairable equipment in the ith REAL system is given by the following:

$$C_{p_i}(N_i, R_i) = N_i \{(A \mid P, i_{nt}, R_i)(p_{ij_i} - f_{ij_i}(R_i)) + i_{nt} f_{ij_i}(R_i) + o_{ij_i}\}$$

The Repair Facility Cost

The annualized cost of procurement and operations for the repair facilities in the ith REAL system is given by the following:

$$C_{f_i}(M_i) = M_i \{(A \mid P, i_{nt}, rr_i)(rp_i - rf_i) + i_{nt} rf_i + ro_i\}$$

The Shortage Cost

The annual cost of the shortages, $C_{s_i}(N_i, M_i, R_i)$, in the ith REAL system, based on the finite source, multiple server, queuing model discussed previously, is given by:

$$(c_i) \frac{\displaystyle\sum_{k=1}^{M_i + d_i - N_i} \frac{k N_i!\, \rho(R_i)^{(N_i - d_i + k)}}{(d_i - k)!\,(N_i - d_i + k)!} + \sum_{k=\max\{1, M_i + d_i - N_i + 1\}}^{d_i} \frac{k\, N_i!\, \rho(R_i)^{(N_i - d_i + k)}}{(d_i - k)!\, M_i!\, M_i^{(N_i - d_i + k - M_i)}}}{\displaystyle\sum_{k=0}^{M_i} \frac{N_i!\, \rho(R_i)^k}{(N_i - k)!\, k!} + \sum_{k=M_i+1}^{N_i} \frac{N_i!\, \rho(R_i)^k}{(N_i - k)!\, M_i!\, M_i^{(k - M_i)}}}$$

The Annual Replacement Capital Constraint

The annual procurement capital cost represents only the steady state annual requirement for new equipment (replacing retiring equipment). It does not include initial procurement capital for the REAL system. The replacement cost component for the ith REAL system is given by the following equation:

$$C_{pr_i} = \left[\frac{N_i}{R_i}\right] p_{ij_i}$$

The Annual Operating Budget Constraint

The annual operating budget constraint represents the limitation on funds to pay for the operating costs of the repairable equipment and repair facilities in all of the REAL systems in the MREAL1 model. The operating cost component for the ith REAL system is given by the following equation:

$$C_{op_i} = N_i \, o_{ij_i} + M_i \, ro_i$$

The System Unavailability Measure

This performance measure is the fraction of equipment demand which, on the average, will not be satisfied because of shortages. It is the ratio of the expected number of shortages over the expected demand. If the j_ith design is used in the ith REAL system, $S_{UNA_{ij_i}}(N_i, M_i, R_i)$ it is given by the following expression:

$$\left[\frac{1}{d_i}\right]\frac{\displaystyle\sum_{k=1}^{M_i+d_i-N_i}\frac{kN_i!\,\rho(R_i)^{(N_i-d_i+k)}}{(d_i-k)!\,(N_i-d_i+k)!}+\sum_{k=\max\{1,M_i+d_i-N_i+1\}}^{d_i}\frac{k\,N_i!\,\rho(R_i)^{(N_i-d_i+k)}}{(d_i-k)!M_i!M_i^{(N_i-d_i+k-M_i)}}}{\displaystyle\sum_{k=0}^{M_i}\frac{N_i!\,\rho(R_i)^k}{(N_i-k)!\,k!}+\sum_{k=M_i+1}^{N_i}\frac{N_i!\,\rho(R_i)^k}{(N_i-k)!\,M_i!\,M_i^{(k-M_i)}}}$$

The Catastrophic Shortage Probability

This performance measure is the steady state probability that the number of shortages will ever be sh_i or greater, i.e. $P(S \geq sh_i)$. For the ith REAL system and the j_ith design, $C_{UNA\,ij_i}(N_i, M_i, R_i \,|\, sh_i)$ is given by the following expression:

$$\frac{\displaystyle\sum_{k=sh_i}^{M_i+d_i-N_i}\frac{N_i!\,\rho(R_i)^{(N_i-d_i+k)}}{(d_i-k)!\,(N_i-d_i+k)!}+\sum_{k=\max\{M_i+d_i-N_i+1,sh_i\}}^{d_i}\frac{N_i!\,\rho(R_i)^{(N_i-d_i+k)}}{(d_i-k)!\,M_i!\,M_i^{(N_i-d_i+k-M_i)}}}{\displaystyle\sum_{k=0}^{M_i}\frac{N_i!\,\rho(R_i)^k}{(N_i-k)!\,k!}+\sum_{k=M_i+1}^{N_i}\frac{N_i!\,\rho(R_i)^k}{(N_i-k)!\,M_i!\,M_i^{(k-M_i)}}}$$

The MREAL1 Test Problem

The MREAL1 Test Problem is a numerical example of a specific MREAL1 model of a hypothetical Metropolitan Transit Authority (MTA). The purpose of the development and optimization of this test problem is to demonstrate that the MREAL1 model behaves reasonably well (for a small problem), and that the global optimum can be approximated (at least) using the MREAL1 Optimizer.

The Metropolitan Transit Authority (MTA) will soon begin operating a very small bus and subway system in an unnamed small town. The MTA Manager believes that the Authority's route structure requires 10 operating buses, and 15 operating subway railcars. Since the Authority will procure new buses and subway railcars, the Manager has narrowed her procurement choices to two makes of bus and two makes of railcar.

The MTA has decided to operate its own maintenance shop for the buses and railcars. The maintenance shop is to be staffed to perform preventive maintenance, unscheduled maintenance, minor damage repairs and minor rebuilds. Major rebuilds due to wearout are not anticipated. The Manager intends to sell old or worn-out buses and railcars for salvage value. Losses due to catastrophe or accident will be covered by insurance policies carried by the Authority.

The system parameters for the MREAL1 Test Problem are shown in Figure 3 on page 65 through Figure 9 on page 71. These parameter values have been obtained by the MTA Manager from sources such as the Town Council (for budget limitations), the bus and railcar manufacturers, the American Public Transportation Association, and her own revenue computations and estimates.[35]

Figure 3 on page 65 shows the system parameters which apply to the MREAL system level. The system parameters which apply to each REAL system level are shown in Figure 4 on page 66 (for the buses), and in Figure 5 on page 67 (for the subway railcars). The data applicable to individual equipment designs are found in Figure 6 on page 68 and Figure 7 on page 69 (for the buses), and in Figure 8 on page 70 and Figure 9 on page 71 (for the railcars).

The figures do not show a unit salvage value for the repairable equipment, since this is a function of the age at which the equipment is retired.[36] Both the MREAL1 test problem enumerator and the MREAL1 test problem optimizer, to be discussed in Chapter 5, include an equation for computing the unit salvage value. This equation represents straight line depreciation in the value of a unit of

---

35 The system parameters values were actually estimated by the author, except for the bus costs, which were obtained from Connelly [1985].

36 The retirement age is a decision variable and therefore not known at the time the values for the system parameters are estimated.

PARAMETER DESCRIPTION	SYMBOL	VALUE
Number of REAL systems	n_{reals}	2
Annual interest rate	i_{nt}	10%
Annual operating budget	b_{op}	\$3,000,000
Annual procurement budget (for replacing retired equipment)	b_{pr}	\$600,000

Figure 3. MREAL System Level Parameters for the Test Problem.

repairable equipment, from its initial procurement cost, p_{ij}, to zero over a period of $rmax_{ij}$ years.[37] The equation for salvage value is:

$$f_{ij}(R_i) = p_{ij}\left[1 - \frac{R_i}{rmax_{ij}}\right]$$

If the MREAL1 test problem enumerator were to examine all points in the region of explicit feasibility which can be represented on the IBM mainframe computer at Virginia Tech, it would have to examine some 4,138,145,550 points. This assumes that $N_1 \in \{10, \dots, 56\}$,[38] $N_2 \in \{15, \dots, 56\}$,[38] $M_1 \in \{1, \dots, N_1\}$, $M_2 \in \{1, \dots, N_2\}$, $R_1 \in \{1, \dots, 15 \text{ or } 20\}$,[39] and $R_2 \in \{1, \dots, 25 \text{ or } 30\}$.[39] The general equation for the total number of potential solutions associated with a specific design combination is given by

$$\prod_{i=1}^{n_{reals}} rmax_{ij} \prod_{i=1}^{n_{reals}} \left(N_{\max_i} - d_i + 1\right)\binom{N_{\max_i} + d_i}{2}$$

[37] $rmax_{ij}$ represents the maximum possible life of the j th design in the ith REAL system.

[38] 56! is the largest factorial which can be represented on the IBM mainframe computer at Virginia Tech. This represents a severe but not insurmountable programming problem if it becomes necessary to optimize an MREAL1 model involving equipment population sizes exceeding 56.

[39] Depending upon which design combination is in effect.

PARAMETER DESCRIPTION	SYMBOL	VALUE
Demand (in buses)	d_1	10
Number of designs considered	s_1	2
Bus shortage-year cost	c_1	$1,825,000
Repair Facility Data (per channel):		
-procurement cost	rp_1	$300,000
-annual operating cost	ro_1	$90,000
-retirement age	rr_1	15 years
-salvage value	rf_1	$80,000
Availability Constraints:		
-maximum expected shortage (as a percent of demand)	b_{UNA_1}	5%
-catastrophic shortage	sh_1	3 or more
-maximum probability of a catastrophic shortage	b_{CAT_1}	0.0001

Figure 4. REAL System Level Parameters for the Bus System.

PARAMETER DESCRIPTION	SYMBOL	VALUE
Demand (in railcars)	d_2	15
Number of designs considered	s_2	2
Railcar shortage-year cost	c_2	$5,475,000
Repair Facility Data (per channel):		
-procurement cost	rp_2	$500,000
-annual operating cost	ro_2	$130,000
-retirement age	rr_2	15 years
-salvage value	rf_2	$120,000
Availability Constraints:		
-maximum expected shortage (as a percent of demand)	b_{UNA_2}	1%
-catastrophic shortage	sh_2	5 or more
-maximum probability of a catastrophic shortage	b_{CAT_2}	0.00001

Figure 5. REAL System Level Parameters for the Subway System.

PARAMETER DESCRIPTION	SYMBOL	VALUE
Cost of a bus (design 1)	p_{11}	\$140,000
Annual operating cost of a bus	o_{11}	\$100,000
Maximum life (in years)	$rmax_{11}$	15

Age Functions for MTBF ($\lambda_{11}(t)$) and MTTR ($\mu_{11}(t)$):

AGE (years)	MTBF (years)	MTTR (years)
1	0.4	0.010
2	0.8	0.011
3	1.0	0.011
4	1.1	0.012
5	1.1	0.013
6	1.0	0.015
7	0.9	0.015
8	0.8	0.015
9	0.7	0.015
10	0.7	0.015
11	0.7	0.017
12	0.5	0.019
13	0.3	0.020
14	0.3	0.021
15	0.2	0.022

Figure 6. Design Parameters for Bus Design 1.

PARAMETER DESCRIPTION	SYMBOL	VALUE
Cost of a bus (design 2)	p_{12}	\$160,000
Annual operating cost of a bus	o_{12}	\$90,000
Maximum life (in years)	$rmax_{12}$	20

Age Functions for MTBF ($\lambda_{12}(t)$) and MTTR ($\mu_{12}(t)$):

AGE (years)	MTBF (years)	MTTR (years)
1	0.5	0.012
2	0.8	0.011
3	1.0	0.012
4	1.1	0.013
5	1.2	0.013
6	1.2	0.014
7	1.2	0.015
8	1.1	0.015
9	1.0	0.015
10	1.0	0.015
11	0.9	0.015
12	0.8	0.016
13	0.7	0.016
14	0.7	0.017
15	0.5	0.020
16	1.4	0.022
17	0.3	0.023
18	0.05	0.025
19	0.05	0.027
20	0.05	0.029

Figure 7. Design Parameters for Bus Design 2.

PARAMETER DESCRIPTION	SYMBOL	VALUE
Cost of a railcar (design 1)	p_{21}	$350,000
Annual operating cost of a railcar	o_{21}	$70,000
Maximum life (in years)	$rmax_{21}$	25

Age Functions for MTBF ($\lambda_{21}(t)$) and MTTR ($\mu_{21}(t)$):

AGE (years)	MTBF (years)	MTTR (years)
1	0.4	0.004
2	1.0	0.002
3	1.2	0.001
4	1.5	0.001
5	1.5	0.001
6	1.5	0.001
7	1.5	0.001
8	1.5	0.001
9	1.5	0.001
10	1.4	0.002
11	1.4	0.002
12	1.4	0.002
13	1.4	0.002
14	1.4	0.002
15	1.3	0.002
16	1.2	0.004
17	1.2	0.004
18	1.2	0.004
19	1.2	0.004
20	1.0	0.004
21	0.8	0.009
22	0.8	0.009
23	0.8	0.009
24	0.8	0.009
25	0.8	0.009

Figure 8. Design Parameters for Railcar Design 1.

PARAMETER DESCRIPTION	SYMBOL	VALUE
Cost of a railcar (design 2)	p_{22}	$420,000
Annual operating cost of a railcar	o_{22}	$65,000
Maximum life (in years)	$rmax_{22}$	30

Age Functions for MTBF ($\lambda_{22}(t)$) and MTTR ($\mu_{22}(t)$):

AGE (years)	MTBF (years)	MTTR (years)
1	0.5	0.003
2	1.0	0.002
3	1.3	0.001
4	1.6	0.001
5	1.6	0.001
6	1.6	0.001
7	1.6	0.001
8	1.6	0.001
9	1.6	0.001
10	1.6	0.001
11	1.5	0.001
12	1.5	0.001
13	1.5	0.001
14	1.5	0.001
15	1.5	0.001
16	1.4	0.002
17	1.4	0.002
18	1.4	0.002
19	1.4	0.002
20	1.4	0.002
21	1.3	0.003
22	1.3	0.003
23	1.3	0.003
24	1.3	0.003
25	1.3	0.003
26	1.2	0.006
27	1.0	0.006
28	0.9	0.006
29	0.8	0.006
30	0.7	0.006

Figure 9. Design Parameters for Railcar Design 2.

where $N_{\max_i}$ is the integer whose factorial value is the largest which can be represented on the computing hardware to be used. The number of design combinations is given by

$$\prod_{i=1}^{n_{reals}} s_i$$

The total number of possible solutions over all design combinations is thus given by the following:

$$\sum_{j_1=1}^{s_1} \sum_{j_2=1}^{s_2} \cdots \sum_{j_{n_{reals}}=1}^{s_{n_{reals}}} \prod_{i=1}^{n_{reals}} rmax_{ij_i} \prod_{i=1}^{n_{reals}} (N_{\max_i} - d_i + 1)\left(\frac{N_{\max_i} + d_i}{2}\right)$$

In Chapter 6 the results of applying the MREAL1 enumerator program and the MREAL1 optimizer program to this test problem are described. The local minima found for the test problem are shown by design combination, and the sample output runs found in Appendices D and E are discussed. An iteration of the optimizer is traced for one design combination. Finally, the possibilities are discussed for using the MREAL1 model to estimate the savings potential of a preventive maintenance program.

Chapter V

OPTIMIZATION OF THE MREAL1 TEST PROBLEM

Discrete Nonlinear Programming Techniques

The model MREAL1 and the MREAL1 test problem have several discrete decision variables. For example, N_i is the MREAL1 decision variable which represents the number of units of repairable equipment procured for the ith REAL system. The system manager can procure 13 or 14, or some other integral number of, units of repairable equipment, but she cannot procure 13.4 units. The same is true for the decision variable M_i, which represents the number of repair channels operated in the ith REAL system.[40]

The model MREAL1 also makes use of finite queuing theory. In this theory, the probability distribution on shortages is undefined for non-integer values of N_i or M_i. Consequently, the objective function obtained in the model MREAL1 is not defined for non-integer values of N_i or M_i.[41]

---

40 It may be possible to visualize the procurement of a partial repair channel through the implementation of channel sharing, or through the hiring of part-time mechanics. In reality, it is doubtful that this variable could ever be made truly continuous.

41 In a private communication, Dr. Joel Nachlas of the Industrial Engineering and Operations Research Department at Virginia Tech has pointed out that a function, $f_A(\bar{x})$, may exist which has the following properties: $f_A : E^n \rightarrow R^1$, $f_A(\bar{x}) = f(\bar{x})$ for all discrete $\bar{x}$ in E^n_{ef}, and $f_A(\bar{x})$ is continuous over all $\bar{x}$ in E^n_{ef}, where

This means that the optimization of model MREAL1 falls in the class of mathematical programming problems called discrete programming. Unfortunately, the part of the MREAL1 model that which deals with equipment design choices is not even a discrete programming problem. Instead, it is a combinatorial programming problem.

Combinatorial Programming

The combinatorial programming problem arises when a decision variable cannot be measured or valued on an ordinal or cardinal scale. In this situation, the only way a number scale can be applied to represent the decision variables is by using 0-1 indicator variables to indicate the presence or lack of a particular decision variable component. In the MREAL1 model, an ordinal scale has yet to be found upon which the designs can be placed. Instead, a series of 0-1 indicator variables represents the decision to accept or reject each individual design. Derigs [1985] describes the formal way in which a combinatorial programming problem is defined. Given a finite set E (such as the four designs of the MREAL1 test problem, $E = \{Bus_1, Bus_2, Rail_1, Rail_2\}$), the modeler chooses an appropriate family of subsets ζ, contained in 2^E, and a real-valued mapping z on ζ which associates with every $S \in \zeta$ a cost or utility. Then the combinatorial programming problem is to find the S which satisfies

$$\min\{z(S) \mid S \in \zeta\} \text{ or } \max\{z(S) \mid S \in \zeta\}$$

In the MREAL1 test problem, 2^E is the collection of 16 sets, shown below:

$$\{\varphi\},$$

$$\{Bus_1\},\ \{Bus_2\},\ \{Rail_1\},\ \{Rail_2\},$$

E^n_{ef} is the region of explicit feasibility of model MREAL1. If such a function can be constructed, then its continuity may be exploited to some advantage in attempting to locate the global, constrained minimum of f.

$$\{Bus_1, Bus_2\},\ \{Bus_1, Rail_1\},\ \{Bus_1, Rail_2\},$$

$$\{Bus_2, Rail_1\},\ \{Bus_2, Rail_2\},\ \{Rail_1, Rail_2\},$$

$$\{Bus_1, Bus_2, Rail_1\},\ \{Bus_1, Bus_2, Rail_2\},$$

$$\{Bus_1, Rail_1, Rail_2\},\ \{Bus_2, Rail_1, Rail_2\},$$

$$\{Bus_1, Bus_2, Rail_1, Rail_2\}$$

ζ may be chosen, at the modeler's convenience, to be any subset of this collection. For the MREAL1 test problem, a ζ, consisting of the following elements, is of interest:

$$\{Bus_1, Rail_1\},\ \{Bus_1, Rail_2\},$$

$$\{Bus_2, Rail_1\},\ \{Bus_2, Rail_2\}$$

This ζ represents all of the possible design combinations in the MREAL1 test problem. One of the purposes of the MREAL1 test problem optimizer is to find the element $S \in \zeta$ which yields the minimum feasible cost. For the combinatorial programming problem represented by the MREAL1 test problem design combinations, the Optimizer uses a complete factorial search method, discussed below.[42] Due to the complexity of the other parts of the MREAL1 optimization problem, partial enumeration techniques for the combinatorial programming portion, such as branch and bound, were not pursued. See Chapter VII for recommendations for future research in this area.

The model MREAL1 is not only partly discrete and partly combinatorial, but it also is nonlinear in the objective function and most of the constraints. In the past 30 years, no single optimization technique has consistently shown itself to be superior to all others in the solution of discrete non-linear and combinatorial nonlinear programming problems. Consequently, the researcher must

[42] This means that *z(S)* is computed for all *S* in zeta, a complete enumeration of all possible solutions in zeta.

have a wide, detailed knowledge of a large number of varied algorithms and heuristics, to be able to make an intelligent choice of technique for a specific problem. And, once this choice is made, the researcher must be prepared to accept disappointing results.

Search Techniques

Many of the techniques for solving nonlinear programming problems are search techniques. In a search technique, a number of sequential or simultaneous evaluations[43] of the objective function and perhaps the constraint equations, are made. The information obtained by these search point observations is used in some way to obtain or estimate the location of the optimal solution. Brooks [1958] describes four categories of search techniques with applicability to nonlinear discrete and combinatorial programming problems.[44] His categories are:

1. Factorial method: this method involves the examination (enumeration) of some or all of the discrete points in the problem's solution space. Full enumeration of all points is called a complete factorial, and guarantees that the best solution found is the global optimum. Partial enumeration is called a fractional or confounded factorial.

2. Univariate method: in this procedure, the decision variables are placed in some (arbitrary) order. Evaluations of the objective function $f(\bar{x})$ are made while varying the value of the first decision variable and holding the remaining variables fixed. The first decision variable is then fixed at the value which corresponds to the best value of $f(\bar{x})$ which was obtained. This process

[43] Numerous synonymous terms are used in the literature: evaluation, trial, observation, experiment, enumeration, and search point.

[44] Brooks' paper discusses methods of seeking maxima for situations in which a response surface, such as the yield of a chemical reaction, is explored by performing experiments. In these experiments, the researcher has considerable freedom in choosing values for the decision variables. However, the researcher can perform only a limited number of experiments, the value of $f(\bar{x})$ being best obtained by experimentation or observation. In model MREAL1, the value of $f(\bar{x})$ is most easily obtained by straightforward computation from the value of $\bar{x}$, due to the complexity of the function. This corresponds to a search observation.

is repeated for the 2nd, 3rd, etc. decision variables, thus completing one iteration of the univariate method. Iterations are repeated over smaller and smaller regions until the chosen stopping rule takes effect. This search method is also known as a sectioning search and also as an alternating variable search. It is originally due to Friedman and Savage [1947].

3. Steepest ascent method: in this method, evaluations of $f(\overline{x})$ are made in the vicinity of $\overline{x}_B$ and used to estimate the direction of gradient at that point. A search is then conducted in the direction of the gradient, until the best value of $f(\overline{x})$ along that line is found. $\overline{x}_B$ is then reset to correspond to this best value of $f(\overline{x})$ and the entire process is repeated. This method is due to Box and Wilson [1951].

4. Random method: a set of points, $\{\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_k, \ldots, \overline{x}_n\}$ are selected at random. For each point, $f(\overline{x}_k)$ is evaluated. The point yielding the best such value is declared to be an estimate of the optimum.

In the time since Brooks wrote this paper, many more search techniques have been developed for discrete and combinatorial programming. Methods such as Box's [1966] complex method, Rosenbrock's [1960] search technique and the Hooke-Jeeves [1961] method are a few of the recent additions. Several such methods will be discussed in the following sections.

Since the objective function and most of the constraints of model MREAL1 are non-linear, discussions of such non-linear programming techniques as the penalty and barrier methods, augmented Lagrangian penalty function method and the gradient projection method of feasible directions also follow. Finally, the problems of adapting sequential line search techniques to the discrete (integer) nature of MREAL1 will be discussed.

Random Search Methods

Brooks [1958] shows that the number of trials n required in a random search to assure with probability, p, that at least one of the $\bar{x}_k$'s falls in the fraction, a, (of all solutions) which has the best values of the objective function, is given by:

$$n = \frac{\ln(1 - p)}{\ln(1 - a)}$$

The weakness of the plain random search method appears when the solution space is large. As indicated in Chapter IV, the "small" MREAL1 test problem has some 4,138,145,550 potential feasible solutions which can be represented on the IBM mainframe at Virginia Tech. If a random search method were set up with a 95% probability of obtaining one or more of the best 1000 solutions to the MREAL1 test problem, the following equation would give the required sample size n:

$$n = \frac{\ln(1 - 0.95)}{\ln(1 - \frac{1000}{4.138 \times 10^9})}$$

$$n = 12{,}394{,}424$$

Assuming that these 12.4 million evaluations of the constraints and objective function were performed on a computer capable of doing an evaluation in 0.001 seconds, the total time required for this sampling plan would be in excess of 34 hours. Computational experience with a DEC Rainbow 100 personal computer indicates that the estimated lower limit on computational time for one evaluation is approximately 1.41 seconds. It would thus take the DEC Rainbow a minimum of 202.76 days to perform this random search!

Recently Conley [1980, 1981] has described numerous examples of the random search method. Conley calls this method Monte Carlo optimization and argues strongly for it use in practical optimization problems of the integer type. He describes a multistage Monte Carlo integer programming technique in which an initial random search is conducted over the entire solution space. In subsequent stages, random searches are conducted in smaller and smaller regions which are focused around the location of the best solution found in the previous stage. Conley believes that this technique is extremely cost- and time-effective for problems with 100 or fewer variables, but he offers no specific proof of this. He does, however, discuss 7 such sample problems which he has solved using this technique.[45]

One significant disadvantage of random search techniques is that the resulting solution may be very far from the global optimum, z^*. Derigs [1985] describes how a confidence interval for z^* may be computed from random search type information for a combinatorial programming problem, using the Weibull distribution. Derigs discusses how this statistical estimation procedure may be used to evaluate how "good" an approximate solution to a combinatorial programming problem is. He references related works which deal with the application of statistical techniques to nonlinear programming problems. These techniques are also used to generate statistical lower bounds when solving a combinatorial optimization problem with the branch and bound method.

Enumerative Search Methods

If a discrete programming problem is sufficiently constrained, it will have a finite number of feasible solutions. In this situation, it may be possible to locate the global optimum by computing the value of the objective function for every discrete point in the feasible region. This procedure works for combinatorial programming problems, too, if the feasible region is small enough and/or there is sufficient, appropriate computing power available.

---

[45] Conley, W., Computer Optimization: A Simplified Approach (New York: Petrocelli Books, 1981), pp. 213-239.

For the purposes of this research, the combinatorial programming problem posed by the choices of equipment designs in model MREAL1 is solved by complete enumeration, i.e. $z(S)$ is computed for all $S \in \zeta$ It should be noted that this limits the size of the MREAL1 model which can be solved. The specific maximum number of designs and REAL systems which can be handled using complete enumeration depend upon the availability and cost of computing resources, and the potential worth of locating an approximately optimal solution to the MREAL1 problem.

The factorial method mentioned by Brooks [1958] is a type of enumeration, in that it sets up a grid of discrete points in the solution space, and evaluates the value of the objective function at each of these discrete points. This may be done for either a discrete or continuous mathematical programming problem. If the problem is discrete, then this amounts to a complete enumeration and is guaranteed to find the optimal solution. If the problem is continuous, the procedure is does not guarantee that the optimal solution will be found. However, for appropriately well behaved objective functions, and a sufficiently small d, the distance between grid points, the factorial method will usually be no further from the optimal solution than $d\sqrt{n}$, where n is the number of decision variables.

Conley [1980] devotes two chapters of his book, Computer Optimization Techniques, to the complete enumeration of integer programming problems. He almost exclusively presents examples of integer programming problems of up to eight variables, the FORTRAN code used to perform the complete enumeration, and the final solution. He does point out that this technique is practical for many decision makers with suitably small problems. The enumeration method is far easier to understand than nearly all other optimization techniques for integer programming problems. And the computing power for small problems is becoming widely available in the form of the personal computer.

Steepest Descent Search Techniques

Also known as gradient based search techniques, their development began in the middle of the 19th century when Cauchy proposed the method of steepest descent. This method determines the direction of steepest descent, $\bar{d}$, from the current point, $\bar{x}^{(k)}$, then proceeds in this direction, searching along the line established by $\bar{d}$ and $\bar{x}^{(k)}$. The direction, $\bar{d}$, is determined by a first order[46] approximation to $f(\bar{x})$.[47] If we let $\mathrm{del}_{\bar{x}} f(\bar{x})$ represent the gradient function of f(x bar) with respect to $\bar{x}$, then the first order direction of steepest descent at $\bar{x}^{(k)}$ is given by:

$$\frac{-\ \mathrm{del}_{\bar{x}} f(\bar{x}^{(k)})}{\|\mathrm{del}_{\bar{x}} f(\bar{x}^{(k)})\|_2}$$

Bazaraa and Shetty [1979] point out that the method of steepest descent usually performs well in the early stages of a search. Unfortunately, as the search approaches a local minimum or the global minimum, this method tends to take smaller and smaller zigzag steps towards the minimum. The other difficulty with the steepest descent method is that it requires $\mathrm{del}_{\bar{x}} f(\bar{x})$.

In Brooks' [1958] discussion of the steepest descent method, he is referring to the situation in which $\mathrm{del}_{\bar{x}} f(\bar{x})$ is unknown because the analytic representation of $f(\bar{x})$ is unknown. This problem is overcome by using finite difference approximations. Gill and Murray [1974] and Stewart [1967] define forward, backward, and central difference approximations to $\mathrm{del}_{\bar{x}} f(\bar{x})$. The ith component of the forward difference approximation to the gradient is given by the following expression:

$$\frac{f(\bar{x} + \Delta\, \bar{e}^{(i)}) - f(\bar{x})}{\Delta}$$

46 linear approximation.

47 A modification of the method of steepest descent, Newton's method, uses first and second order approximations to $f(\bar{x})$.

where Δ is a small positive number. The ith component of the backward difference approximation to the gradient is given by the following expression:

$$\frac{f(\bar{x}) - f(\bar{x} - \Delta\, \bar{e}^{(i)})}{\Delta}$$

The ith component of the central difference approximation to the gradient is given by:

$$\frac{f(\bar{x} + \Delta\, \bar{e}^{(i)}) - f(\bar{x} - \Delta\, \bar{e}^{(i)})}{2\,\Delta}$$

The finite difference approximation method is potentially useful when starting a search from a point which is not near a minimum, and when the gradient function is not available. The only requirement is that the function $f(\bar{x})$ be computable. The objective function of model MREAL1 meets this requirement. Also, its gradient function is not available. The MREAL1 optimizer program therefore makes use of the finite difference approximation to the gradient to determine the initial search direction. However, it is used only for the first iteration. Other search techniques are used in subsequent iterations.

Newton's method has been successfully applied as a gradient based search technique in multi-dimensional nonlinear unconstrained minimization problems. Newton's method uses a quadratic approximation to $f(\bar{x})$ at $\bar{x}^{(k)}$ to determine the search direction, d. Unfortunately, this approximation requires the Hessian matrix of f, which is clearly not available for model MREAL1. The approximation also requires the gradient of f. Adapting Newton's method to the MREAL1 model would imply that f was being approximated using an approximation of the gradient of f and an approximation of the Hessian of f. This amount of approximation to would probably not perform well, so Newton's method was not chosen for the MREAL1 optimizer.

Box [1954] discusses the use of a polynomial approximations to $f(\bar{x})$ to find a local maximum of a response surface. This response surface is a function, $f(\bar{x})$, whose algebraic form is not known, but which can be evaluated by experimental (or mathematical) methods at many or all values of

$\overline{x} \in E^n$. In Box's method he suggests that the method of steepest descent, in ascent form for his problems, be used first, until it makes no further progress improving $f(\overline{x})$. This represents the approximation of $f(\overline{x})$ with a first order polynomial (a hyper-plane). Box suggests the use of a second order polynomial approximation as the next step in the minimization procedure. This second order approximation is used to determine whether or not a local maximum has been found, and if not, to continue the search. Box suggests increasing the order of the polynomial approximation as necessary until a local or the global maximum has been located. In Box's paper, the coefficients of the polynomial approximations are obtained by performing a least squares fit of the polynomial form to an appropriate number of evaluations of $f(\overline{x})$

Rosenbrock's Search Technique

There are a rather large number of techniques for the unconstrained minimization of a multi-dimensional function, $f(\overline{x})$. According to Bazaraa and Shetty [1979], the method of Rosenbrock [1960] and the method of Zangwill [1967] are considered quite efficient. Rosenbrock's original method consists of the following steps:

1. Initialization step. Set $k = 0$. Choose a point $x^{(k)}$. Set $\overline{d}_i = \overline{e}_i$ for all $i \in \{1, 2, \dots, n\}$.

2. Sectioning search step. In Rosenbrock's original method, the directions, $\overline{d}_i$, are searched one at a time using fixed steps. After all directions have been searched, the process is repeated with appropriately modified and/or accelerated step sizes for each direction, if the previous sectioning iteration produced an improvement in $f(\overline{x})$. This step is repeated until it fails to produce an improvement in $f(\overline{x})$.

3. Gram-Schmidt orthogonalization step. In this step, it is presumed that the sectioning search step has moved the search to a point on a resolution ridge (in maximization problems) or a

resolution valley (in minimization problems).[48] Rosenbrock attempts to escape from this situation by rotating the directions, $\bar{d}_i$, so that one of them may approximately coincide with the direction of the ridge crest (in maximization problems) or the valley bottom (in minimization problems). The procedure used to rotate the directions is Gram-Schmidt orthogonalization. The direction of the resolution ridge or valley is not determined directly. Instead, the coordinate rotation is based upon the direction established between $\bar{x}^{(k)}$ and the best point known when orthogonalization was begun. When it has been completed, $\bar{x}^{(k+1)}$ is set equal to the best point found so far. Return to the sectioning search step.

4. Termination criteria. Terminate the algorithm in step 3 if $\|\bar{x}^{(k+1)} - \bar{x}^{(k)}\|_2$ is less than some small positive scalar.

A logical modification of Rosenbrock's method is to replace the discrete step search (with acceleration) by a line search. This modification was reported by Swann [1964]. This line search solves the following minimization problem sequentially, for each $i \in \{1, 2, \dots, n\}$.

$$\underset{\Delta}{\text{Minimize}} \ f(\bar{x}_{(i)}^{(k)} + \Delta\, \bar{d}_i)$$

Each time this minimization is performed $\bar{x}_{(i)}^{(k)}$ is replaced by the result of the current minimization and $\bar{d}_i$ is replaced with $\bar{d}_{i+1}$. The MREAL1 optimizer uses Rosenbrock's method, with a line search modified for the discrete nature of model MREAL1, to perform the imbedded unconstrained minimizations.

[48] A resolution ridge refers to a special situation in a search procedure which uses discrete steps. In this situation, the search may stop part of the way up a narrow, steep sided ridge whose direction does not coincide with any of the directions, $\bar{d}_i$. This occurs when the discrete search step sizes are such that the function evaluations around the ridge point "fall" off the ridge in all directions, plus or minus $\bar{d}_i$.

Univariate Minimization

There are a number of univariate minimization methods. These methods typically assume that some large interval [a, b] of the domain of a univariate function is available in which the minimum is known to lie. These methods then attempt to progressively reduce the size of the interval, making sure it still contains the minimum, until it is "small enough". Gill and Murray [1981], Box, Davies and Swann [1969], and others describe the three most prominent interval type univariate searches: the Fibonacci method, the golden section method, and polynomial interpolation. Of these three methods, Fibonacci's is the most efficient if judged by reduction in interval size for a fixed number of iterations of the method.

The Fibonacci method was chosen for use in the MREAL1 optimizer because it adapted very nicely to the integer nature of the problem. For the opposite reason, the golden section method was not chosen. In the MREAL1 optimizer, a large number of discrete line searches must be performed. While these line searches are not, strictly speaking, univariate minimizations,[49] they are looking for the point giving the minimum function value among the discrete points nearest a specific line (direction) through the solution space. This is very similar to a univariate minimization.

The polynomial interpolation method works well on smooth, continuous functions. Since the functions in the MREAL1 model are discrete, this method was not chosen. Note, however, that Chapter 7 recommends that a version of this method be explicitly investigated for possible use in future optimization procedures for MREAL systems as its performance compared to the Fibonacci method is not known.

---

[49] Because they may involve changes in many decision variables.

Constrained Nonlinear Optimization

There are a large number of techniques for constrained nonlinear optimization, as can easily be seen in texts such as Bazaraa and Shetty [1979], Beightler, Phillips, and Wilde [1979], Beveridge and Schechter [1970], Box, Davies and Swann [1969], Fiacco and McCormick [1968], Fletcher [1969], Gill, Murray, and Wright [1981], Greig [1980], Himmelblau [1972], Rao [1979], Rau [1970], Wismer and Chattergy [1978], and Zangwill [1969]. While some such techniques have shown their superiority over other methods for problems with specific structures, none have been shown to be better than all other techniques for all constrained nonlinear problems.

Some of these techniques involve searches of the feasible region using various schemes to maintain feasibility in the face of nonlinear constraints. Some are limited to nonlinear objective functions and linear constraints. These, of course, take advantage of of the linearity of the constraints to assist in maintaining feasibility during the search. There are other methods which involve the transformation of the constrained nonlinear problem into a sequence of unconstrained nonlinear problems. Fiacco and McCormick [1968] call these methods sequential unconstrained minimization. The following sections will describe some of these methods, since the MREAL1 optimizer makes use of one such method.

Gill, Murray and Wright [1981] describe the history of the penalty and barrier function methods often used in sequential unconstrained minimization. They point out that penalty and barrier function methods have a long history of use in solving engineering problems, particularly quadratic and absolute value penalty functions. In the penalty function method, an auxiliary function (also known as a merit function) is created from the objective function and the constraint functions. The idea is to allow a sequence of unconstrained minimizations to be performed on the auxiliary function such that the solution found at the end of a sufficient number of such minimizations will correspond to the solution to the constrained problem. The trick is to create an auxiliary function which will ultimately force feasibility. This is done by composing a function of the objective function and a penalty term which is zero for feasible solutions, and increasingly large for infeasible solutions as they move further and further from the feasible region. If a quadratic penalty term is used, the auxiliary function typically takes the form:

$$f(\overline{x}) + \frac{1}{2} \sum_{i \in I} \sigma_i \{g_i(\overline{x})\}^2$$

where $f(\overline{x})$ is the objective function, $g_i(\overline{x})$ is the ith constraint of m constraints, I is the set of violated constraints, and σ_i is the ith penalty parameter, a positive number.

If an absolute value penalty term is used, the auxiliary function typically takes the form:

$$f(\overline{x}) + \sum_{i \in I} \sigma_i \, | g_i(\overline{x}) |$$

where $f(\overline{x})$ is the objective function, $g_i(\overline{x})$ is the ith constraint of m constraints, and σ_i is the ith penalty parameter, a positive number.

An advantage of the penalty function method is that it does not require a starting feasible solution, since the auxiliary function can represent infeasible solutions. This also leads to the possibility of a program failure if, for instance, it leads to taking the square root of a negative number. The choice of initial penalty parameter is also important, in that too large a value may cause ill-conditioning.

The barrier function method requires an initial feasible solution. It proceeds through a sequence of unconstrained minimizations of the auxiliary function, but in this case the auxiliary function contains a barrier term which tends to be small except when solutions approach the constraints (infeasibility). Two typical barrier terms are the inverse and the logarithmic. The inverse barrier function, credited to Carroll [1961], typically takes the form:

$$f(\overline{x}) + \sigma \sum_{i=1}^{m} \frac{1}{g_i(\overline{x})}$$

where $f(\bar{x})$ is the objective function, $g_i(\bar{x})$ is the ith constraint of m constraints, and σ is the penalty parameter, a positive number.

The logarithmic barrier function, credited to Frisch [1955], typically takes the form:

$$f(\bar{x}) - \sigma \sum_{i=1}^{m} \ln(g_i(\bar{x}))$$

where $f(\bar{x})$ is the objective function, $g_i(\bar{x})$ is the ith constraint of m constraints, and σ is the ith penalty parameter, a positive number. The main disadvantage of the barrier method is the requirement for a starting feasible solution. For this reason, the barrier method was not used in the MREAL1 optimizer.

Augmented Lagrangian Penalty Function Methods

Haarhoff and Buys [1970], Hestenes [1969], and Powell [1969] first proposed the use of an augmented Lagrangian (ALAG) penalty function for nonlinear programming[50] at about the same time. They proposed its use as an auxiliary function in the sequence of unconstrained minimizations used to solve a nonlinear program subject to equality constraints. This form of auxiliary function was developed using the quadratic penalty term mentioned previously, and was shown to overcome the computational problems (ill conditioning) of the plain quadratic penalty function method.

The ALAG penalty function method was extended from Powell's work by Fletcher [1975] in order to deal with inequality constraints. Balachandran [1984] indicates that many researchers in nonlinear programming consider the ALAG penalty function technique to be the "most effective and efficient transformation method for solving the nonlinear program."[51] Both Balachandran [1984] and

[50] Also known as the method of multipliers.

[51] S. Balachandran, "Dynamic Scaling of Nonlinear Programs and Its Implementation Within the Augmented Lagrangian Penalty Function Technique" (Doctoral Dissertation, VPI&SU, 1984), p. 17.

Gill, Murray, and Wright [1981] give excellent summaries of the historical development of the augmented Lagrangian penalty function method. They also describe the most popular versions of the method. The ALAG penalty function method used in the MREAL1 optimizer is adapted from Balachadran [1984] It is discussed in the section of this chapter describing the outer loop of the MREAL1 test problem optimizer.

The MREAL1 Test Problem Enumerator

The MREAL1 Test Problem Enumerator is a software package which computes the values of the objective and constraint functions for selected points within the region of explicit feasibility for the MREAL1 Test Problem. The program was written in the FORTRAN-77 programming language, using the IBM VS FORTRAN compiler. Of the points (values of the decision variables) which are enumerated in a particular run, the program finds and records two special ones: a) the point which yields the lowest cost, regardless of feasibility, and b) the point which yields the lowest feasible cost. The Enumerator was developed for two purposes: 1) to assist in the development of subroutines to be used in the MREAL1 Test Problem Optimizer, and 2) to provide a means for determining whether or not a particular solution is a N_2 discrete local minimum.

The MREAL1 test problem enumerator can be used for four distinct purposes:

1. To determine whether or not a particular point is a N_2 discrete local minimum.

2. To follow the path of steepest descent (in the discrete sense) from a starting point to a discrete local minimum.

3. To find the point with the least cost and the point with the the least feasible cost in a particular portion of the region of explicit feasibility.

4. Given sufficient computing time, the Enumerator can find the global optimum for the MREAL1 Test Problem.

In order for the Enumerator to be used to follow the path of steepest descent, a series of enumerations are performed. From a feasible starting point $\bar{x}_b$ (chosen arbitrarily or by some heuristic), the Enumerator is used to check whether or not $\bar{x}_b$ is a N_2 discrete local minimum, i.e. whether or not $f(\bar{x}_b) \leq f(\bar{x})$ for all feasible $\bar{x}$ such that $|x_{b_i} - x_i| \leq 1$ for all $i \in \{1, \ldots, n\}$.[52] If so, a discrete local minimum has been found.

Otherwise, the process is repeated after setting $\bar{x}_b$ equal to the best feasible point found. This represents a step in the local direction of steepest descent. This step has a maximum length of $\sqrt{n}$ where n is the dimensionality of the problem. For a design combination in the MREAL1 test problem, $n = 6$. For the MREAL1 model in general, a design combination results in a problem with n equal to three times the number of REAL systems.

These steps are repeated until a local minimum is found. Note that measures must be taken to prevent infinite cycling through a series of two or more adjacent feasible points with equal objective function values. (Adjacent in the sense that $|x_{b_i} - x_i| \leq 1$ for all $i \in \{1, \ldots, n\}$.)

The tasks of finding the best and best feasible points in all or part of the region of explicit feasibility are performed by the Enumerator when the DO-loops in MAIN are set to the enumerate the desired region. As has been discussed previously, however, the amount of time required to enumerate a sizable portion of the problem for a realistic system may be unacceptably long. Appendix B shows the FORTRAN code for the Enumerator as it is used to perform the check for a local minimum. The functions of the main program and each of the subroutines in the Enumerator are described in the following sections.

---

[52] There are $3^n - 1$ such points adjacent to $\bar{x}_b$.

Main Program

The Main Program controls the selection of points to be enumerated, and computes the integer values of the decision variables for these points. It also sets the design combination flags to reflect current choices of design in each REAL System. It contains logic which allows the user to choose between a regional type of enumeration and an adjacent point type of enumeration. However, this logic is not user friendly, but rather is intended to reduce the number of programming changes which would otherwise be required to switch between the two types of enumeration.

Once a particular point $\bar{x}$ has been selected by the DO-loops for enumeration, the Main Program calls Subroutine COMP which handles the computation of the objective and constraint function values for $\bar{x}$. The Main Program also handles the initial and final printer output for a particular run, including the printing of information about the best, and best feasible, points found.

Block Data Input Subprogram

This Block Data subprogram contains the cost and constraint data for the MREAL1 Test Problem. It also contains the ρ-profile for each possible equipment design in each REAL system. The annual interest rate is included in this subroutine, but the capital recovery factors (the $A \mid P$'s) corresponding to this interest rate are not. They are contained in a separate Block Data subprogram, called APE.

Subroutine COMP

Subroutine COMP makes all of the calls which are necessary to compute the value of the objective function and the values of all the constraint functions for a given point $\bar{x}$. Because of the nature of the imbedded finite source queuing model in MREAL1, a normalization factor must be calculated prior to any computations involving the finite source queuing model. This is performed by Subroutine NORMAL.

The objective function value for the point $\bar{x}$ is computed by COMP via a call to subroutine OBJ. The annual replacement procurement capital constraint is computed by subtracting the sum of the values obtained from function CPR_i from the budget amount b_{pr} . The annual operating budget constraint is calculated by subtracting the sum of the values obtained from function COP_i from the budget amount b_{op}.

The constraints on equipment unavailability are computed separately for each REAL system, i, and are calculated in terms of the expected percent of the demand, d_i, which will not be met in the enumerated solution, $\bar{x}$. This statistical expectation is computed by function $SUNA_i$, and then subtracted from the constraint amount, b_{UNA_i}.

The constraints on the probability of catastrophic equipment unavailability are also computed separately for each REAL system, i . The function $CUNA_i$ computes this probability for a particular REAL system, and this value is then subtracted from the maximum allowable probability, b_{CAT_i}.

Subroutine COMP also performs several other functions. It records a flag value in FLAG1 which represents the constraint number of the last violated constraint for the point $\bar{x}$. This value is passed to MAIN via the COMMON block labeled NLP. COMP also saves $\bar{x}$ if it has either the best cost found so far, or if it is the best feasible point found so far. When this occurs the corresponding constraint and objective function values are also saved, along with the corresponding design combination. Finally Subroutine COMP prints the coordinates of $\bar{x}$, the design combination, the value of the objective function, the value of FLAG1, and the values of the constraint functions.

Subroutine OBJ

This subroutine computes the total system cost at $\bar{x}$ for both of the REAL systems in MREAL1. This total system cost includes the annual equivalent cost in each REAL system for all of the equipment (taken over the lifetime specified by $\bar{x}$), an annual operating cost for each equipment population, an annual equivalent cost for the number of repair channels specified in $\bar{x}$, an annual cost of operating the repair channels, and the annual expected cost of equipment shortages.

Subroutine NORMAL

This subroutine computes the finite source queuing model normalization factor for the enumerated point $\bar{x}$. Since this is potentially the sum of some very small numbers, the individual components of the normalization factor are computed in double precision. This subroutine computes the normalization factor for each REAL system. The computations in NORMAL and several of the function subroutines require factorials. Since NORMAL and these function subroutines are called many times in the course of a given run, considerable computational effort would be required to compute the factorial values each time they are needed. Consequently a double precision function subroutine which contains the 57 factorial values that the IBM mainframe can store (from 0! to 56!), is included in the Enumerator.

Function CPR

This function computes the annual cost of replacement capital for an individual REAL system. This value is computed for the *j*th design of the *i*th REAL system from the following equation:

$$CPR_i = \frac{N_i}{R_i} \times p_{ij}$$

Function COP

This function computes the annual operating cost for both the equipment and repair channels in a particular REAL system. This value is computed for the jth design in the i th REAL system from the following equation:

$$COP_i = (N_i \times o_{ij}) + (M_i \times ro_i)$$

Function CUNA

This function computes the scaled value of the catastrophic unavailability constraint for a particular REAL system. CUNA is scaled using a factor of $r_i = 1.0 \times 10^{10}$ for $i = \{1,2\}$. The scale factor insures that the value of this constraint function has an order of magnitude similar to that of the objective function and the other constraints. While this scale factor is not critical for the functioning of the MREAL1 Test Problem Enumerator, it is critical for the augmented Lagrangian penalty function used in the Optimizer. The value of CUNA is computed for the ith REAL system using the following:

$$(r_i)\frac{\displaystyle\sum_{k=sh_i}^{M_i + d_i - N_i} \frac{N_i!\,\rho(R_i)^{(N_i - d_i + k)}}{(d_i - k)!\,(N_i - d_i + k)!} + \sum_{k=\max\{M_i + d_i - N_i + 1,\, sh_i\}}^{d_i} \frac{N_i!\,\rho(R_i)^{(N_i - d_i + k)}}{(d_i - k)!\,M_i!\,M_i^{(N_i - d_i + k - M_i)}}}{\displaystyle\sum_{k=0}^{M_i} \frac{N_i!\,\rho(R_i)^k}{(N_i - k)!\,k!} + \sum_{k=M_i+1}^{N_i} \frac{N_i!\,\rho(R_i)^k}{(N_i - k)!\,M_i!\,M_i^{(k - M_i)}}}$$

Function SUNA

This function computes the scaled value of the unavailability constraint for a particular REAL system. SUNA is scaled using a factor of $r_i = 1.0 \times 10^9$ for $i = \{1,2\}$. The scale factor gives this constraint function an order of magnitude similar to that of the objective function and the other constraints. The value of SUNA is computed for the ith REAL system using the following:

$$\left[\frac{r_i}{d_i}\right] \frac{\displaystyle\sum_{k=1}^{M_i + d_i - N_i} \frac{k N_i!\, \rho(R_i)^{(N_i - d_i + k)}}{(d_i - k)!\,(N_i - d_i + k)!} + \sum_{k=\max\{1,\, M_i + d_i - N_i + 1\}}^{d_i} \frac{k\, N_i!\, \rho(R_i)^{(N_i - d_i + k)}}{(d_i - k)!\, M_i!\, M_i^{(N_i - d_i + k - M_i)}}}{\displaystyle\sum_{k=0}^{M_i} \frac{N_i!\, \rho(R_i)^k}{(N_i - k)!\, k!} + \sum_{k=M_i+1}^{N_i} \frac{N_i!\, \rho(R_i)^k}{(N_i - k)!\, M_i!\, M_i^{(k - M_i)}}}$$

Function SUNA1

This function computes the unscaled value of the unavailability constraint for a particular REAL system. It computes the steady state, expected number of shortages of equipment units for the REAL system. This value is required by subroutine OBJ. SUNA1 is computed for the ith REAL system using the following:

$$\frac{\displaystyle\sum_{k=1}^{M_i + d_i - N_i} \frac{k N_i!\, \rho(R_i)^{(N_i - d_i + k)}}{(d_i - k)!\,(N_i - d_i + k)!} + \sum_{k=\max\{1,\, M_i + d_i - N_i + 1\}}^{d_i} \frac{k\, N_i!\, \rho(R_i)^{(N_i - d_i + k)}}{(d_i - k)!\, M_i!\, M_i^{(N_i - d_i + k - M_i)}}}{\displaystyle\sum_{k=0}^{M_i} \frac{N_i!\, \rho(R_i)^k}{(N_i - k)!\, k!} + \sum_{k=M_i+1}^{N_i} \frac{N_i!\, \rho(R_i)^k}{(N_i - k)!\, M_i!\, M_i^{(k - M_i)}}}$$

Double Precision Function NFACT

This function returns the value of N! for values of $N \in \{0, \ldots, 56\}$. Numbers larger than 56! (7.237 E+75) cannot be stored by the FORTRAN compiler on the IBM mainframe computer system at Virginia Tech.

The MREAL1 Test Problem Optimizer

MREAL1 Optimizer Program Structure

The MREAL1 Optimizer is a computer program built specifically to find the optimal solution to the MREAL1 Test Problem. The construction of the Optimizer demonstrates that a model of the type and complexity of the MREAL1 Test Problem can be built and optimized. Whether or not such a model accurately represents real-world systems can only be determined by future testing of model results against actual system performance. The Optimizer is written in FORTRAN-77 and consists of approximately 3300 lines of code, plus calls to intrinsic FORTRAN functions and International Mathematics Subroutine Library (IMSL) subroutines.

To perform the optimization, the program makes use of partial enumeration, difference equation approximations to the gradient, the augmented Lagrangian penalty function technique, Rosenbrock's search technique, discrete Fibonacci line searches, the projected gradient algorithm and a heuristic starting point selection procedure, all adapted to the discrete nature of the MREAL1 model. The solution which results from the use of the Optimizer is not guaranteed to be the global optimum. It may not even be a discrete local minimum. The program contains four main nested loops:

1. The design enumeration loop.

2. The starting point selection loop.

3. The outer loop of the augmented Lagrangian penalty function algorithm.

4. The inner loop of the augmented Lagrangian penalty function algorithm.[53]

The Design Enumeration Loop.

The design enumeration loop is actually a nested series of n_{reals} loops where n_{reals} is the number of REAL systems in the MREAL system being optimized. In the case of the Optimizer for the MREAL1 Test Problem, $n_{reals} = 2$. The ith loop iterates from 1 to s_i, where s_i is the number of design choices in the i th REAL system. As a result, the Optimizer performs an optimization for each of the $\prod_{i=1}^{n_{reals}} s_i$ possible design combinations. Each of these optimizations represents the solution to a multiple equipment, single design (MESD) problem. For the MREAL1 Test Problem, four such MESD problems make up the entire multiple equipment, multiple design (MEMD) problem.

The Starting Point Selection Loop.

The starting point selection loop represents a heuristic for selecting starting points. The first such point is chosen by setting each equipment population size, N_i, equal to the demand, d_i plus a small number of units of equipment, such as 1 or 2. The number of repair channels, M_i, is set equal to a small number, such as 1, 2, or 3. Finally, the retirement age, R_i, is set by finding the integer nearest $0.75 \times rmax_{ij}$.

---

[53] In this loop an unconstrained minimization is performed on the augmented Lagrangian penalty function.

Once the initial search point has been chosen, it is passed to the Outer Loop which then proceeds to find a local minimum for the current design combination. The resulting solution is then used to generate two new starting search points. The Outer Loop is then called upon twice more to find local minima. The best of the resulting local minima is used to generate two more starting points.

When neither of the latest two starting points leads to a local minimum which is strictly better than the current best, the heuristic terminates for the current design combination. The steps in the Starting Point Selection Loop are thus:

1. Initialization step. The initial starting point is chosen by a heuristic which attempts to chose a solution that meets the budgetary constraints of the MREAL1 test problem. Each equipment population is given 1 spare unit of equipment and two repair channels. The retirement age is set at three quarters of the maximum possible retirement age. This will hopefully place the initial solution on the high side of the flat part of the ρ profile, before the negative effects of old age have set in, but still old enough to appropriately amortize the original investment in the repairable equipment. The initial starting point is computed by the following rules:

$$N_i = d_i + 1 \text{ for all } i \in \{1, 2\}$$

$$M_i = 2 \text{ for all } i \in \{1, 2\}$$

$$R_i = [0.75 \times rmax_{ij}]_I \text{ for all } i \in \{1, 2\}$$

   where $[\ \ ]_I$ represents the nearest integer.

2. Call the Outer Loop. This step returns with a local minimum $\bar{x}^*$ for the current design combination.

3. Starting point generation step. This step generates two new starting points from the local minimum $\bar{x}^*$ found in the previous step. The first set of heuristic rules shown below generates a point with a larger equipment population size but a smaller number of repair channels than

the solution $\bar{x}^*$, in the hope that this trade-off may lead to an improved solution. Also, since fewer repair channels are used in the first new solution, the retirement age is decreased to make the population, on the average, more reliable and easier to fix. The second set of rules shown below generates a new starting point by trying a decreased equipment population size and an increased number of repair channels. The retirement age is also increased since the increased number of repair channels means increased repair capacity which may be able to handle the decrease in population reliability. Here are the rules for generating these two starting points:

$$N_i = N_i^* + 2$$

$$M_i = \max\{1, M_i^* - 2\}$$

$$R_i = \max\{1, [0.7 \times R_i^*]_I\}$$

for all $i \in \{1, 2\}$ and where $[\ \ ]_I$ means nearest integer.

and

$$N_i = \max\{d_i, N_i^* - 1\}$$

$$M_i = M_i^* + 1$$

$$R_i = \min\{rmax_{ij}, [1.3 \times R_i^*]_I\}$$

for all $i \in \{1, 2\}$ and where $[\ \ ]_I$ means nearest integer.

4. Call the Outer Loop twice. This step returns two, possibly identical local minimums, $\bar{x}_1^*$ and $\bar{x}_2^*$ for the current design combination.

5. Stopping rule. If $z^* \leq z_1^*$ and $z^* \leq z_2^*$ then return control to the design enumeration loop. Otherwise return to step 3 with $\bar{x}_1^*$ or $\bar{x}_2^*$, whichever yields $\min\{z_1^*, z_2^*\}$.

The Outer Loop

Once a particular starting search point has been chosen, the Outer Loop initializes and controls the updating of the Lagrange multipliers and penalty parameters which are used by the search procedure in the Inner Loop. The Outer Loop is controlled primarily from subroutine DUNLOP. It consists of the following steps:

1. Initialization step. Set $k = 0$. Set the Lagrange multipliers to zero, i.e. $\bar{\lambda}^{(k)} = \bar{0}$. Set the penalty parameters to one, i.e. $\bar{\sigma}^{(k)} = \bar{1}$. Let $\bar{x}^{(k)}$ represent the starting search point.

2. Inner Loop step. Call the Inner Loop to minimize the augmented Lagrangian penalty function with respect to $\bar{x}$, starting from the point $\bar{x}^{(k)}$. This step thus solves the following problem in the $(k + 1)$st iteration:

$$\underset{\bar{x}}{\text{Minimize}} \quad L_a(\bar{x}, \bar{\lambda}^{(k)}, \bar{\sigma}^{(k)})$$

   where:

$$L_a(\bar{x}, \bar{\lambda}^{(k)}, \bar{\sigma}^{(k)}) = f(\bar{x}) + \frac{1}{2} \sum_{i=1}^{m} \sigma_i \left(\min\{g_i(\bar{x}) - \frac{\lambda_i}{\sigma_i}, 0\}\right)^2$$

   and $g_i(\bar{x}) \geq 0$ is the ith constraint.

3. Stopping Condition Check. When the Inner Loop fails to make further improvements in $L_a(\bar{x}, \bar{\lambda}^{(k)}, \bar{\sigma}^{(k)})$, then the current best point, $\bar{x}^{(k)}$, is checked to see if it may be a feasible discrete local minimum of the original objective function. This is accomplished by subroutine LOCMIN. If $\bar{x}^{(k)}$ appears to be a discrete local minimum, then the Outer Loop is terminated and control is passed back to the Starting Point Selection Loop. If $\bar{x}^{(k)}$ is infeasible, or a better adjacent feasible solution is found then the Outer Loop proceeds to step 4.

4. Lagrange multiplier update step. If $\max_i |\min\{g_i(\bar{x}^{(k)}), 0\}| < \max_i |\min\{g_i(\bar{x}^{(k-1)}), 0\}|$ then update the Lagrange multipliers using the Powell-Hestenes updating formula given below:

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} - \min\{\sigma_i^{(k)} g_i(\bar{x}^{(k)}), \lambda_i^{(k)}\} \quad \text{for all } i \in \{1, \dots, m\}$$

Otherwise set $\bar{\lambda}^{(k+1)} = \bar{\lambda}^{(k)}$.

5. Penalty parameter update step. If $\max_i |\min\{g_i(\bar{x}^{(k)}), 0\}| \geq \max_i |\min\{g_i(\bar{x}^{(k-1)}), 0\}|$ then update the ith penalty parameter by the following formula

$$\text{If} \quad -\min\{g_i(\bar{x}^{(k)}), 0\} \geq \max_i |\min\{g_i(\bar{x}^{(k-1)}), 0\}|$$

$$\text{Then set } \sigma_i^{(k+1)} = 10\,\sigma_i^{(k)}$$

Otherwise set $\sigma_i^{(k+1)} = \sigma_i^{(k)}$ and return to step 2.

The Inner Loop

The Inner Loop performs an unconstrained minimization of the specific augmented Lagrangian penalty function established by the Outer Loop. This loop is controlled by subroutine INNER. It begins by using either a forward or a backward difference equation about the start point $\bar{x}^{(k)}$ to approximate the gradient at $\bar{x}^{(k)}$. A Fibonacci line search is then performed in the direction $-\text{del}_{\bar{x}} L_a(\bar{x}^{(k)}, \bar{\lambda}^{(k)}, \bar{\sigma}^{(k)})$. The Fibonacci search stops at the first minimum it encounters in this direction.

From this point onward, the program uses Rosenbrock's search procedure, adapted for the discrete case. The Inner Loop returns control to the Outer Loop as soon as Rosenbrock's procedure stalls. Since the Inner Loop step in the Outer Loop is immediately followed by the local minima check in subroutine LOCMIN, it is not critical that the Inner Loop have a very accurate stopping rule. Consequently, the Inner Loop stopping rule is the stalling of Rosenbrock's procedure.

The steps of the Inner Loop are described below.

1. Initial step. Approximate $\text{del}_{\bar{x}}\, L_a(\bar{x}^{(k)}, \bar{\lambda}^{(k)}, \bar{\sigma}^{(k)})$ using a forward or a backward difference equation, whichever yields the greatest improvement in the ALAG penalty function, to approximate each component. If, for a particular component of $\text{del}_{\bar{x}}\, L_a$, neither the backward nor forward difference equation gives an improved value of the ALAG penalty function, then that specific component of $\text{del}_{\bar{x}}\, L_a$ is set equal to zero. Explicit feasibility is maintained by logic coded in this section of the program. This step is concluded after computing

$$\text{del}_{\bar{x}}\, \hat{L}_a(\bar{x}^{(k)}) = -\frac{\text{del}_{\bar{x}}\, L_a(\bar{x}^{(k)}, \bar{\lambda}^{(k)}, \bar{\sigma}^{(k)})}{\|\text{del}_{\bar{x}}\, L_a(\bar{x}^{(k)}, \bar{\lambda}^{(k)}, \bar{\sigma}^{(k)})\|_2}$$

This is the normalized, approximate direction of steepest descent from $\bar{x}^{(k)}$.

2. Fibonacci line search step. Since the points in the region of explicit feasibility make up a regular lattice, a given direction such as $\text{del}_{\bar{x}}\, \hat{L}_a(\bar{x}^{(k)})$ starting at $\bar{x}^{(k)}$ will not necessarily pass through any other point in the region. Consequently, it is necessary to examine the points of the lattice which lie closest to the line through $\bar{x}^{(k)}$ in the direction $\text{del}_{\bar{x}}\, \hat{L}_a(\bar{x}^{(k)})$. Fox and Liebman [1981] have pointed out that such points are given by the following:

$$\bar{x} = \left[\bar{x}^{(k)} + j\, \frac{\text{del}_{\bar{x}}\, \hat{L}_a(\bar{x}^{(k)})}{\|\text{del}_{\bar{x}}\, \hat{L}_a(\bar{x}^{(k)})\|_\infty} \right]_I$$

where $j \in I$ and $[\quad]_I$ represents the nearest integer.

If we restrict j to the Fibonacci sequence, $F_k \in \{1,2,3,5,8,13,21,\ldots\}$, and use this equation to generate a sequence of discrete search points, we obtain the nice result that points established by any three consecutive Fibonacci numbers define an interval to which a perfect discrete Fibonacci interval search can be applied. This is shown in the following discussion, which is simplified by assuming that:

$$\bar{x}^{(k)} = 0 \qquad \text{and}$$

$$\frac{\text{del}_{\bar{x}} \hat{L}_a(\bar{x}^{(k)})}{\| \text{del}_{\bar{x}} \hat{L}_a(\bar{x}^{(k)}) \|_\infty} = 1$$

This means that the $\bar{x}$'s will be the Fibonacci numbers.

Suppose we have points which were generated by three consecutive Fibonacci numbers: F_n , F_{n+1}, and F_{n+2}. These $\bar{x}$'s will be referred to as F_n, F_{n+1} , and F_{n+2}. The interval of uncertainty defined by F_n and F_{n+2} has length F_{n+1} since, by definition, $F_{n+2} = F_n + F_{n+1}$. F_{n+1} is one of the two points to be searched in this interval, the other being generated by $2F_n$. Depending upon the value of the ALAG penalty function at these points, one of following three intervals of uncertainty will become the new interval of uncertainty: (F_n, $2F_n$), (F_{n+1}, F_{n+2}), or (F_{n+1}, 2 F_n). The length of interval (F_n, $2F_n$), is given by:

$$2F_n - F_n = F_n,$$

a Fibonacci number. The length of the interval (F_{n+1}, F_{n+2}) is given by:

$$\begin{aligned} F_{n+2} - F_{n+1} &= F_{n+1} + F_n - F_{n+1} \\ &= F_n \end{aligned}$$

also a Fibonacci number. Finally, the length of the interval (F_{n+1}, $2F_n$) is given by:

$$\begin{aligned} 2F_n - F_{n+1} &= 2F_n - (F_n + F_{n-1}) \\ &= F_n - F_{n-1} \\ &= F_{n-2} \end{aligned}$$

which, too, is a Fibonacci number. Since all possible intervals of uncertainty resulting from this search have length equal to a Fibonacci number, by induction all subsequent search intervals will have a length equal to a Fibonacci number. This continues until an interval of uncertainty of length 1 is obtained. At this time the search ends since the basic ALAG function is defined only for discrete (integer) values of the decision variables.

The Fibonacci line search actually is begun by first proceeding away from $\bar{x}^{(k)}$ in the direction defined in step 1 of the Inner Loop. Ever larger steps are taken as defined by the Fibonacci sequence, as long as the ALAG penalty function continues to improve. As soon as the improvement ceases, an interval search of the interval established by the last 3 Fibonacci numbers is performed. This Fibonacci interval search is conducted to locate the point with the smallest value of the ALAG penalty function between the points represented by F_{k-2} and F_k. Let the resulting point be represented by $\bar{x}_{(j)}{}^{(k)}$.

3. Rosenbrock's search step. Once $\bar{x}_{(j)}{}^{(k)}$ has been reached, it could be possible to repeat the gradient approximation procedure. The gradient approximation, however, is computed using difference equations with respect to the principal coordinate axes. This could result in the procedure stalling at the bottom of a resolution valley whose principal axis does not lie parallel to any of the principal coordinate axes. To avoid this problem, Rosenbrock's procedure is used. Rosenbrock's technique consists of a sequence of iterations of the following two steps:

   a. Gram-Schmidt orthogonalization substep. A Gram-Schmidt orthogonalization is used to rotate the search axes so that one of them lies parallel to the direction $\bar{x}^{(k)} - \bar{x}_{(j)}{}^{(k)}$

   b. Sectioning search substep. A sectioning (univariate) search is conducted on the rotated coordinate axes. The author has modified the line search of Rosenbrock in order to adapt it to the discrete nature of the problem. The modification consists of the discrete Fibonacci line search described previously.

4. Termination step. Let the solution which results when the discrete Fibonacci line search has been completed on all the rotated coordinate axis be represented by $\bar{x}_{(j+1)}{}^{(k)}$. If $\bar{x}_{(j+1)}{}^{(k)} = \bar{x}_{(j)}{}^{(k)}$ then control is returned to the Outer Loop. Otherwise return to step 3.

MREAL1 Optimizer Program Components

Main Program

The Main program contains a short glossary, the complete variable declarations for the Optimizer, and three labeled COMMON blocks. Labeled COMMON block NLP contains the variables which must be passed between the subroutines involved in the four nested loops of the Optimizer. Labeled COMMON block OTHER contains miscellaneous and add-on variables. Labeled COMMON block FINAL contains the common variables required by MAIN to keep track of the best overall solution obtained over all design combinations.

The Main program writes the output header, controls the Design Enumeration Loop and controls the Starting Point Selection Loop. When all four nested program loops have been completed, the Main program calls for final output of the results from subroutine MANOUT.

Block Data Input Subprogram

The Block Data Input subprogram contains the system parameter values for the MREAL1 Test Problem. It also contains the ρ-profile for each possible equipment design in each REAL system. The annual interest rate is included in this subroutine, but the capital recovery factors (the $A \mid P'$s) corresponding to this interest rate are not. They are contained in a separate Block Data subprogram, called APE.

Subroutine DUNLOP

Subroutine DUNLOP handles the subroutine calls which control the Outer Loop. These calls include subroutine INNER, which controls the Inner Loop, subroutine LOCMIN, which checks the stopping criteria, and subroutine OUTER, which updates the Lagrange multipliers and penalty parameters. As a safety measure, subroutine DUNLOP also contains a check to terminate the Outer Loop after a fixed number of iterations. This subroutine also writes a header for the results produced by the Outer Loop.

Subroutine RESULT

Subroutine RESULT writes the final result for each design combination examined under the control of the Design Enumeration Loop. It also checks to see if the best solution for the current design combination is better than the best solution found for all previous design combinations. If so, RESULT records this solution in the variables in the labeled COMMON block FINAL. RESULT is called by MAIN upon completion of the Outer Loop.

Subroutine MANOUT

Subroutine MANOUT is called by MAIN to write the final, global results from the Optimizer. This subroutine writes a summary of the result for each design combination, then it writes the overall solution, followed by a detailed recapitulation of the overall solution. MANOUT is called by MAIN after all loops have been completed.

Subroutine INIT

Subroutine INIT generates the first start point of the Starting Point Selection Loop. It also sets the initial values for the Lagrange multipliers and penalty parameters. Finally, INIT checks the feasibility of the first start point and initializes the value of the largest constraint violation. INIT is called by MAIN and writes the coordinates of the first start point.

Subroutine INIT1

Subroutine INIT1 generates the "upper" start points for the Starting Point Selection Loop and writes the coordinates of these points. It also sets initial values for the Lagrange multipliers and penalty parameters. Finally, INIT1 checks the feasibility of the first start point and initializes the value of the largest constraint violation. INIT1 is called by MAIN.

Subroutine INIT2

Subroutine INIT2 generates the "lower" start points for the Starting Point Selection Loop and writes the coordinates of these points. It also sets initial values for the Lagrange multipliers and penalty parameters. Finally, INIT2 checks the feasibility of the first start point and initializes the value of the largest constraint violation. INIT2 is called by MAIN.

Subroutine INNER

Subroutine INNER performs the unconstrained minimization of the augmented Lagrangian penalty function set up by the Outer Loop. The subroutine terminates when two subsequent applications of Rosenbrock's search technique produce the same solution. The details of the operation of subroutine INNER are contained in the description of the Inner Loop given previously. INNER is called by subroutine DUNLOP.

Subroutine LOCMIN

This subroutine contains the stopping rule for the Outer Loop. It is entered when Rosenbrock's search procedure (in subroutine INNER) stalls. Called from subroutine DUNLOP, LOCMIN makes an approximate determination as to whether or not the current solution, $\bar{x}^{(k)}$, is a discrete local minimum. This determination is approximate because LOCMIN does not evaluate the constraints and objective function values of all the points in some pre-defined neighborhood of $\bar{x}^{(k)}$. Instead, LOCMIN attempts to locate an improved feasible solution by moving some distance from $\bar{x}^{(k)}$ using the gradient projection method of feasible directions. If such an improved solution is not found, $\bar{x}^{(k)}$ is declared to be a local minimum.

To determine for certain that $\bar{x}^{(k)}$ is a discrete local minimum requires that $f(\bar{x}^{(k)}) \leq f(\bar{x})$ for all $\bar{x} \in N(\bar{x}^{(k)})$ where $N(\bar{x}^{(k)})$ is a neighborhood of $\bar{x}^{(k)}$. While the FORTRAN code which must be written to accomplish such a determination is easy to write, it can rapidly result in a program with an enormous run time. Even small neighborhoods such as $N_2(\bar{x}^{(k)})$ can result in an extraordinarily large number of evaluations in a multi-dimensional problem.

Suppose we choose a neighborhood, $N_2(\bar{x}^{(k)}) \equiv \{\bar{x} \mid |x_i - x_i^{(k)}| \leq 1 \text{ for all } i \in [1,2,\ldots,12]\}$.[54] This neighborhood represents a 12 dimensional hypercube with an edge length of 2, centered on $\bar{x}^{(k)}$. Then the number of points contained in $N_2(\bar{x}^{(k)})$ is given by $3^{12} - 1 = 531{,}440$. This represents an enormous amount of computation.

To keep the computation time down to a reasonable amount for realistic problems, LOCMIN uses an adaptation of the gradient projection method of Rosen as the stopping rule for the outer loop.

[54] An MREAL1 type optimization involving four REAL systems would fit this situation.

This procedure requires much less computing time than the complete enumeration of all points in a hypercube neighborhood. Unfortunately, it also means that we are not guaranteed that $\bar{x}^{(k)}$ is a discrete local minimum as defined above.

The adaptation of Rosen's gradient projection method starts from the point $\bar{x}^{(k)}$ and projects a modified version of the negative gradient of the objective function so that, hopefully, it improves while feasibility is maintained. In LOCMIN, the first step taken is to perform a search along the negative gradient of the objective function without regard to feasibility. This search is limited to a number of adjacent integer points along the search direction equal to the dimensionality of the problem plus three.[55] If an improved feasible point is found, LOCMIN returns to DUNLOP with the new, improved point. Otherwise, a new search direction from $\bar{x}^{(k)}$ is computed so that feasibility is maintained.

To accomplish this LOCMIN first determines which constraints are binding in the direction of the negative gradient of the objective function. The appropriate gradient of each of the binding constraints is computed and placed as a row in matrix M. Next, the projection matrix, P, is computed from the following:

$$P = I - M^t(MM^t)^{-1}M$$

Finally, the projected gradient direction, d_k, is computed from:

$$d_{(k)} = -P\, \mathrm{del}_{\bar{x}} f(\bar{x}^{(k)})$$

where f is the objective function.

Since there are nonlinear constraints in the problem, feasibility may not be maintained for any significant distance from $\bar{x}^{(k)}$ in the direction $d_{(k)}$. However, this direction is a logical choice for

[55] This number was chosen arbitrarily to ensure that the search goes beyond the immediate local neighborhood of $\bar{x}^{(k)}$.

searching for better feasible points from $\bar{x}^{(k)}$. LOCMIN therefore performs a search in the direction $d_{(k)}$. This search moves outward from $\bar{x}^{(k)}$ and checks a specific number of the integer valued points nearest $d_{(k)}$. The number of points searched is equal to the dimensionality of the problem plus three. If an improved feasible point is not found, LOCMIN returns to DUNLOP with the presumption that $\bar{x}^{(k)}$ is a local minimum.

It should be noted that by the time a point $\bar{x}^{(k)}$ is sent to LOCMIN to check the stopping rule, both the gradient approximation procedure and Rosenbrock's method have failed to improve upon it. If LOCMIN, too, cannot find an improved feasible solution, then there is a good chance that $\bar{x}^{(k)}$ is indeed a discrete local minimum. The results of running the MREAL1 Test Problem through the Optimizer bear this out. The Optimizer located 36 points in the MREAL1 Test Problem which LOCMIN could not improve upon. Thirty-one of these points turned out to be discrete local minima with respect to a hypercube with an edge length of 2 and centered on the point.

Subroutine OUTER

This subroutine first calls subroutine CONSTR so that the maximum constraint violation can be determined. It then proceeds to update the Lagrange multipliers and penalty parameters for the augmented Lagrangian penalty function, as part of the Outer Loop. Finally it updates the iteration counter for the Outer Loop and records the current maximum constraint violation for use during the next call to subroutine OUTER. This subroutine is called by DUNLOP.

Subroutine CONSTR

Given a point $\bar{x}$, subroutine CONSTR computes the slack/violation for each constraint at that point and places the values in an array named GEE. CONSTR also determines which constraint has the greatest violation (if any) and places the violation value in MAXG1, and the corresponding constraint number in FLAG1. This subroutine is called from LOCMIN and OUTER. CONSTR makes use of subprograms NORMAL, CPR, COP, SUNA and CUNA. These subprograms are described in the section on the MREAL1 Test Problem Enumerator.

Subroutine CNSTR1

Given a point $\bar{x}$, subroutine CNSTR1 computes the slack/violation for each constraint at that point and places the values in an array named GEE. CNSTR1 also determines which constraints are violated and places a 1 in the corresponding locations in the array FLAGC after having first initialized all elements of FLAGC to zero. This subroutine is called from ALAG. CNSTR1 makes use of subprograms NORMAL, CPR, COP, SUNA and CUNA.

Subroutine XPLSIT

Since the model MREAL1 is undefined for certain values of the decision variables, such as $N_i < d_i$, the Optimizer must insure that all search points it generates remain within the region of explicit feasibility. Otherwise, the program may terminate with errors or unpredictable results if a point outside this region is sent to subroutine NORMAL, CONSTR, CNSTR1, ALAG or OBJ. When XPLSIT is sent a point $\bar{x}$, which lies outside the region of explicit feasibility, it returns the point in the region which is closest to $\bar{x}$. All searches are thus forced to remain within the region of explicit feasibility. XPLSIT is called from subroutines LOCMIN, INNER and CGRAD.

Subroutine XPLST1

This subroutine performs the same function as subroutine XPLSIT. In addition, it sets the variable FLAG to 1 whenever XPLST1 is sent a point which lies outside the region of explicit feasibility. XPLST1 is called from subroutine LOCMIN.

Subroutine XPLST2

XPLST2 does not reset $\bar{x}$ if it lies outside the region of explicit feasibility. It does, however, place a 1 in the appropriate element of the array FLAGE for each violation of the explicit constraints it encounters. XPLST2 is called from subroutine LOCMIN.

Subroutine XPLST3

This subroutine forces explicit feasibility. Before doing so, however, it computes a penalty amount of the form:

$$\sum_{i=1}^{12} r_i \; V_i^2$$

where r_i is a scale factor for the constraint and V_i is the constraint violation of the ith constraint. XPLST3 is called from subroutine ALAG1.

Subroutine ALAG

ALAG computes the value of the augmented Lagrangian penalty function for the Inner Loop. It calls subroutine CNSTR1 to obtain the implicit constraint violations. (Explicit feasibility is forced before ALAG is called.) ALAG is called from subroutine INNER. This subroutine also writes a line of information at the end of each call, so that the progress of the search procedures can be monitored.

Subroutine ALAG1

ALAG1 computes the value of the augmented Lagrangian penalty function for the Inner Loop. It includes a penalty for violating the explicit constraints, before forcing explicit feasibility. This prevents the search procedures in subroutine INNER from trying to go very far from the region of explicit feasibility. ALAG1 calls subroutine CNSTR1 to obtain the implicit constraint violations and XPLST3 to obtain the explicit constraint violations. ALAG1 is called from subroutine INNER. This subroutine also writes a line of information at the end of each call, so that the progress of the search procedures can be monitored.

Subroutine OBJ

OBJ computes the actual value of the objective function, $f(\overline{x})$ for a point $\overline{x}$, in the region of explicit feasibility. It is called from subroutine LOCMIN.

Subroutine CGRAD

Subroutine CGRAD computes the gradient of a selected implicit constraint. A computed GOTO statement at the beginning of the subroutine determines which constraint's gradient will be computed. The gradients of two of the constraints are computed analytically, while the remaining four constraints must have their gradients approximated. Once a constraint gradient is computed, CGRAD normalizes the components. The constraint gradients are required by the gradient projection technique contained in subroutine LOCMIN.

Subroutine ECGRAD

This subroutine computes the gradient of a selected explicit constraint. A computed GOTO statement at the beginning of the subroutine determines which explicit constraint's gradient will be computed. All of these gradients are computed analytically. They are required by the gradient projection technique contained in subroutine LOCMIN.

Subroutine NORMAL

This subroutine serves the same purpose and operates in the same way as subroutine NORMAL in the MREAL1 Test Problem Enumerator.

Functions NFACT, CPR, COP, CUNA, SUNA, and SUNA1

These functions serve the same purpose and operate in the same was as the functions of the same name in the MREAL1 Test Problem Enumerator.

Chapter VI

RESULTS AND CONCLUSIONS

This dissertation has described a class of systems which involve the design, procurement and support of collections of repairable equipment in order to meet various demands. Several examples of this class of system were described. An indication of their importance was also given. An orderly hierarchy of the elements of such systems was described. Finally, the definitions of the various possible decision variables, system parameters and measures of system effectiveness were discussed.

Based upon these definitions, a primitive model of such a system was formulated. This model, called MREAL1, attempts to capture the essential behavior of several important components of the MREAL system, in a holistic sense. The resulting model is primarily combinatorial and discrete. It has a nonlinear objective function and several nonlinear constraints.

In order to investigate the usefulness of this model, several research activities were undertaken. The structure of, and data for, a small, hypothetical MREAL system were created. The MREAL1 model was then developed for this hypothetical system. Finally, two different optimization techniques were used to solve the MREAL1 model of the hypothetical system.

Before developing the optimization techniques for the hypothetical system, however, it was necessary to determine the nature of the objective function. The nature of this function materially affects both the choice and the effectiveness of the possible optimization techniques. Since the objective function of the MREAL1 model, by definition, exists only for integer values of the decision variables, there are a number of complications. First, there may be a large number of discrete local minima, depending upon how the local neighborhood is defined. Secondly, it becomes more difficult to make use of mathematical properties like convexity to assist the optimization process. Thirdly, many of the well developed tools used in the optimization of continuous mathematical programs do not adapt well to the nonlinear, integer case.

Behavior of the Objective Function in MREAL1

Pertinent to the problem of finding an optimal solution to the problem posed in the formulation of the MREAL1 model is a determination of the behavior of the objective function. An objective function possessing many local minima different from the global minimum is much more difficult to deal with than one with a single, global minimum.

Since the graph of a function conveys a great deal of information, simplified three dimensional graphs were made of the objective function of the MREAL test problem. The three dimensional plotting technique available from the Statistical Analysis System (SAS) was employed.

Using the numerical data for the bus design number 1 of the MREAL1 test problem, three different plots were created by the following procedure. Two of the three decision variables for the bus portion of the hypothetical MREAL system (one REAL system) were chosen as independent variables. The value of the third decision variable was held fixed. The expected total REAL system cost was then computed over a range of values chosen for each of the independent variables. Plotting these values yields a picture of a slice of the multi-dimensional hypersurface described by the objective function of the REAL system modeled in MREAL1. Note that the SAS plotting routine performs a linear interpolation for the value of the objective function in between values of the independent variables actually passed to the plotting routine. Each of these plots is described in Figure 10 on page 117 below.

Plot Name	Independent Variables	Range of Values	Fixed Variable	Value of Fixed Variable
MR	Number of repair channels	1-10	Population size	10
	Retirement age in years	1-15		
NR	Population size	10-20	Number of repair channels	1
	Retirement age in years	1-15		
MN	Number of repair channels	1-N	Retirement age	15 years
	Population size	10-20		

Figure 10. Descriptions of Three Plots of a Portion of the MREAL1 Test Problem

From looking at these plots, it appears that the objective function is fairly well behaved for the bus REAL system. In fact, it appears as though a convex function might result if an appropriate continuous valued function were fit to the integer valued objective function. Such a continuous function would, however, have to avoid deviating very much from linearity between adjacent integer points.

Validation of the MREAL1 Enumerator and Optimizer

The validation of the MREAL1 model itself awaits future testing by potential users or researchers. It was not intended to be a stunningly accurate representation of the behavior of an actual MREAL system. Rather, it was intended to demonstrate the possibility of building and solving a holistic model of an MREAL system. Consequently, the validation of the model from the viewpoint of its accuracy representing reality is not performed in this dissertation.

Other kinds of validation were, however, performed. It is important to be certain that the enumerator and the optimizer compute values of the objective and constraint functions in the same

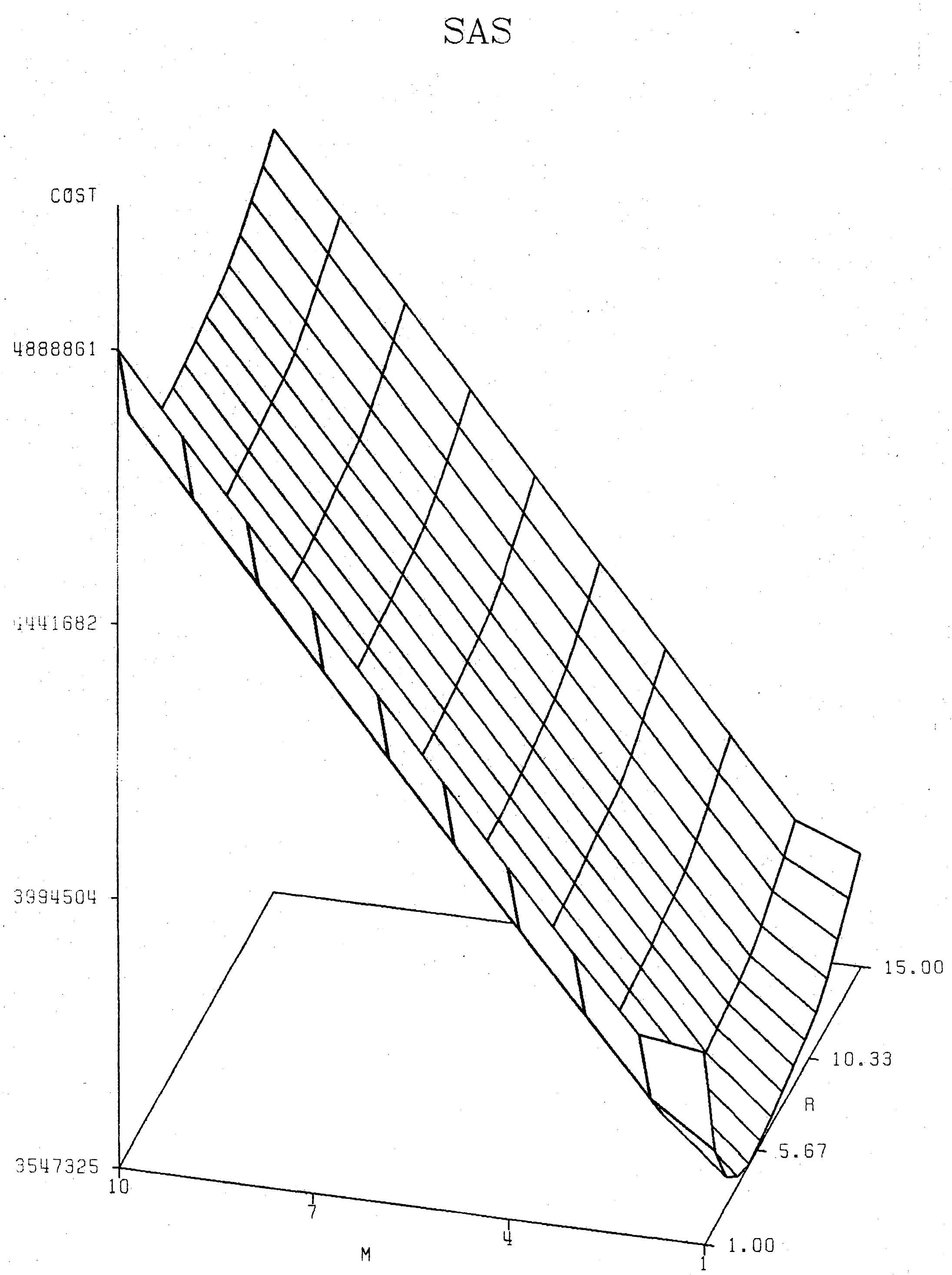

Figure 11. 3 Dimensional Plot of Repair Channels, M, and Retirement Age, R, vs. Total System Cost.

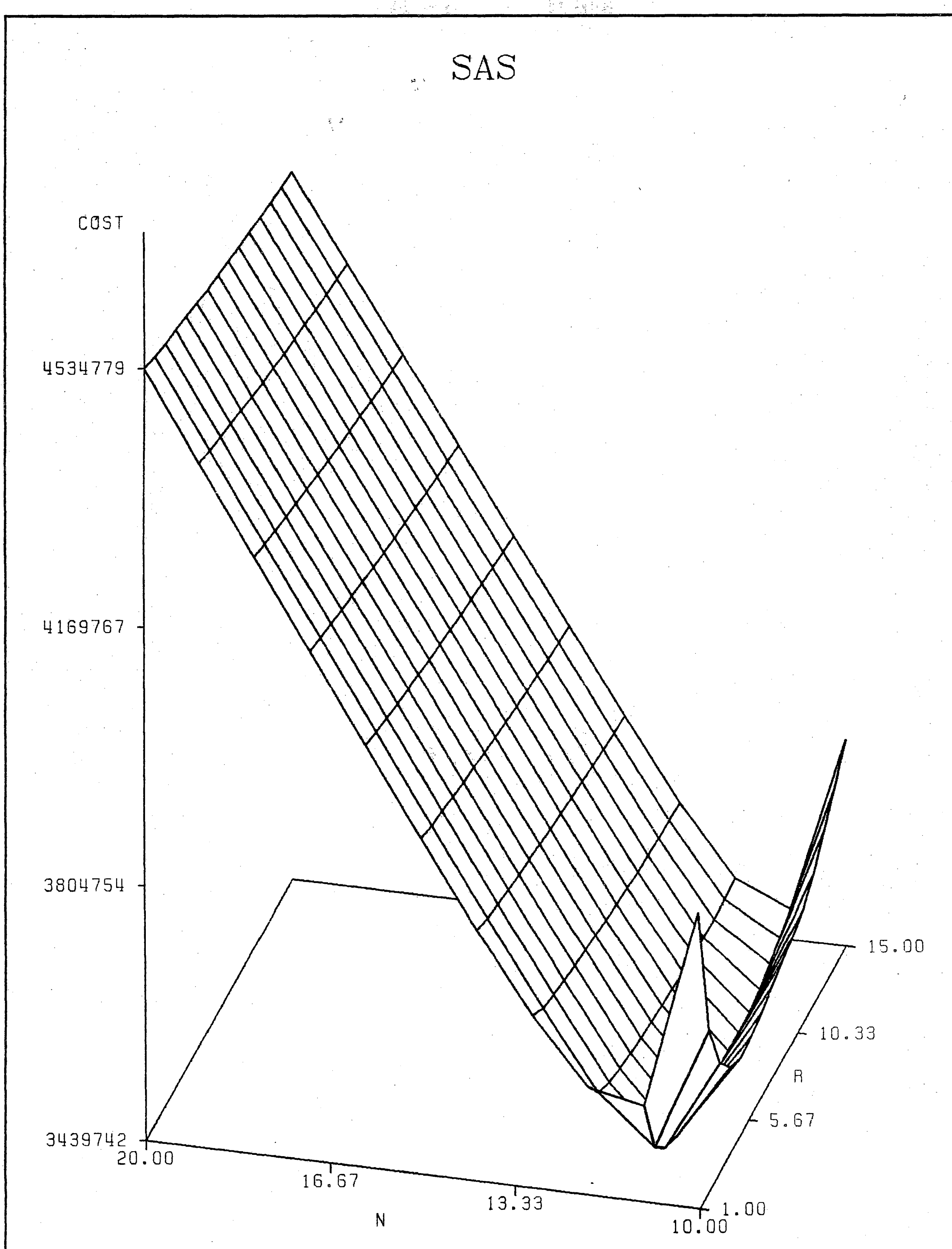

Figure 12. 3 Dimensional Plot of Population Size, N, and Retirement Age, R, vs. Total System Cost.

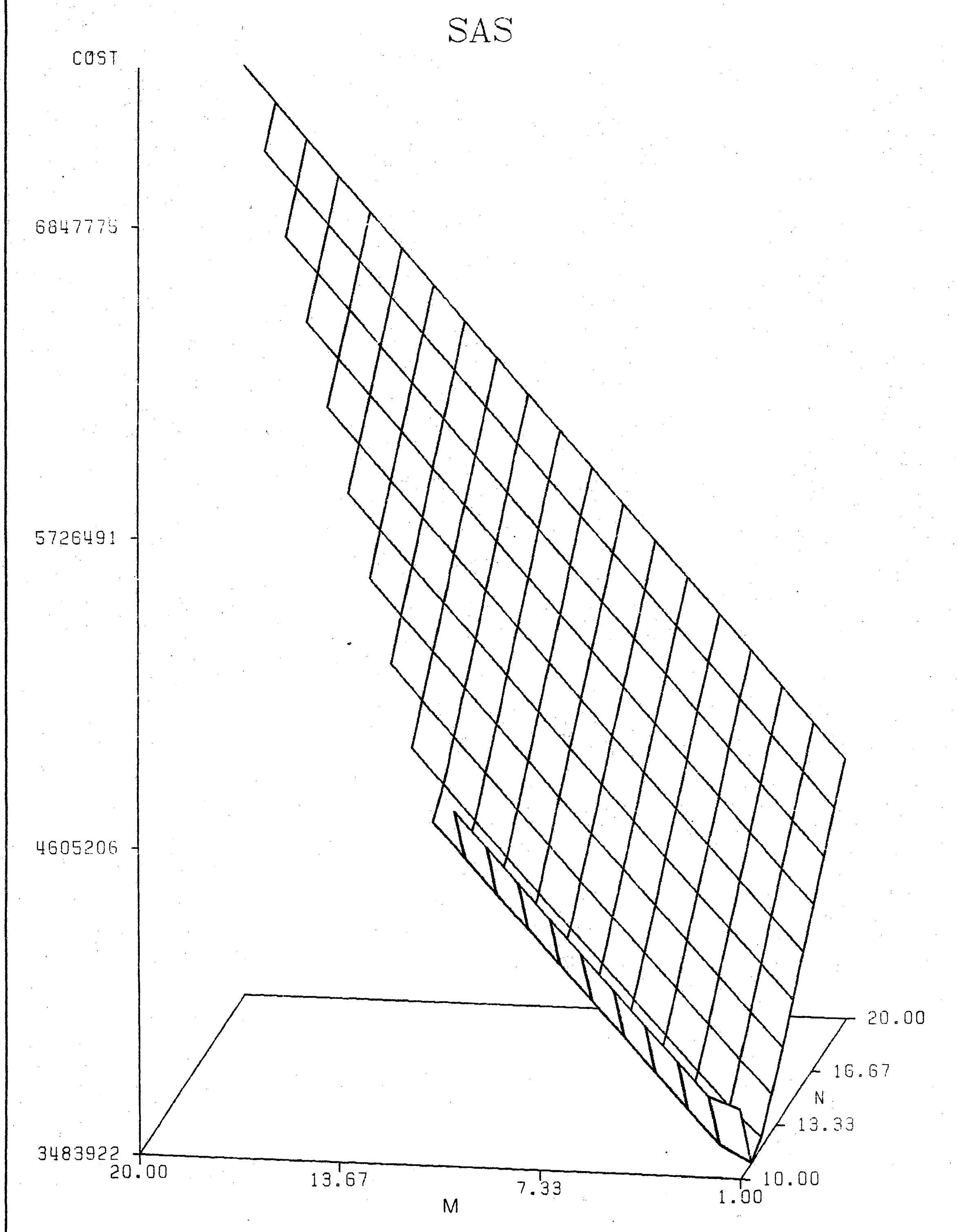

Figure 13. 3 Dimensional Plot of Repair Channels, M, and Population Size, N, vs. Total System Cost.

way. This validation was performed simply by comparing a samples of enumerator output to samples of optimizer output. Examples of both are contained in Appendices D and E, respectively.

The MREAL1 enumerator was also compared to an earlier model which used a similar objective function (see Fabrycky, Malmborg, Moore and Brammer [1984]). This model, the Repairable Equipment Population System Demonstrator (REPS Demonstrator), performs the calculations needed to obtain values for the objective function and the expected shortage function in a single equipment, single design (SESD) system.[56]

The REPS demonstrator was given a set of input data equivalent to the data used in the bus REAL system of the MREAL1 test problem. A series of possible solutions was input, and the resulting values of the objective function and expected shortage function were obtained.

The REPS demonstrator requires that a specific salvage value be input corresponding to the choice of retirement age. Since the MREAL1 enumerator computes the salvage value based on retirement age and does so according to a linear depreciation in value to zero over the maximum possible life, it was necessary to ensure that the appropriate salvage value was input to the REPS demonstrator whenever a solution with a different retirement age was input. It was also necessary to insure that the MTTR- and MTBF-profiles used in the REPS demonstrator corresponded to the data in the MREAL1 enumerator.

The MREAL1 enumerator was modified to do the computations and create the output for the single equipment, single design (SESD) system. It was then run for the same set of possible solutions that was used in the REPS demonstrator. Figure 14 on page 122 shows the results. These results correspond exactly with the results from the REPS demonstrator.

There is an important secondary result of this last validation process. It shows that the MREAL1 model, originally written for a multiple equipment, multiple design (MEMD) problem, can be used

---

[56] This is a single REAL system, with one choice of repairable equipment design.

```
****************************************************************
*                                                              *
*             MREAL1 ENUMERATOR FOR SESD/SEMD PROBLEMS         *
*                                                              *
*             COPYRIGHT 1986 - THOMAS PRESTON MOORE            *
*                                                              *
*             WRITTEN IN FORTRAN 77 (IBM VS-FORTRAN)           *
*                                                              *
*                   LINE PRINTER OUTPUT FORMAT                 *
*                                                              *
****************************************************************
```

FROM ENUMSESD: INITIAL BASE POINT: 11 2 11
ALL POINTS ADJACENT TO THIS BASE POINT ARE ENUMERATED BELOW:

DECISION VARIABLES				SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES					
N1	M1	R1	DESIGN	TOTAL COST	FLAG1	APRC	AOPC	ASUNA	ACUNA
10	1	10	1	.1647353E+07	5	460000.	1910000.	.323634E+08	-.245858E+08
10	1	11	1	.1659878E+07	5	472727.	1910000.	.315039E+08	-.279486E+08
10	1	12	1	.1683692E+07	5	483333.	1910000.	.300332E+08	11342499E+08
10	2	10	1	.1732646E+07	5	460000.	1820000.	.346445E+08	-.510645E+07
10	2	11	1	.1741420E+07	5	472727.	1820000.	.339906E+08	-.589782E+07
10	2	12	1	.1758521E+07	5	483333.	1820000.	.328877E+08	-.737697E+07
10	3	10	1	.1858437E+07	5	460000.	1730000.	.347066E+08	-.299633E+07
10	3	11	1	.1867064E+07	5	472727.	1730000.	.340607E+08	-.351063E+07
10	3	12	1	.1883888E+07	5	483333.	1730000.	.329730E+08	-.447046E+07
11	1	10	1	.1499338E+07	5	446000.	1810000.	.470413E+08	-.329225E+07
11	1	11	1	.1500869E+07	5	460000.	1810000.	.467669E+08	-.406014E+07
11	1	12	1	.1506544E+07	5	471667.	1810000.	.462734E+08	-.557897E+07
11	2	10	1	.1595833E+07	0	446000.	1720000.	.487086E+08	0.486455E+06
11	2	11	1	.1594398E+07	0	460000.	1720000.	.485968E+08	0.395417E+06
11	2	12	1	.1594698E+07	0	471667.	1720000.	.483978E+08	0.215702E+06
11	3	10	1	.1721650E+07	0	446000.	1630000.	.487693E+08	0.775924E+06
11	3	11	1	.1720067E+07	0	460000.	1630000.	.486656E+08	0.736411E+06
11	3	12	1	.1720089E+07	0	471667.	1630000.	.484818E+08	0.658513E+06
12	1	10	1	.1575076E+07	0	432000.	1710000.	.494587E+08	0.214728E+06
12	1	11	1	.1572653E+07	0	447273.	1710000.	.493837E+08	0.354311E+05
12	1	12	1	.1571614E+07	5	460000.	1710000.	.492416E+08	-.338956E+06
12	2	10	1	.1694284E+07	0	432000.	1620000.	.498815E+08	0.952886E+06
12	2	11	1	.1690778E+07	0	447273.	1620000.	.498658E+08	0.942193E+06
12	2	12	1	.1687683E+07	0	460000.	1620000.	.498364E+08	0.919896E+06
12	3	10	1	.1820419E+07	0	432000.	1530000.	.499247E+08	0.986294E+06
12	3	11	1	.1816806E+07	0	447273.	1530000.	.499149E+08	0.983196E+06
12	3	12	1	.1813507E+07	0	460000.	1530000.	.498966E+08	0.976745E+06

N1	M1	R1	DESIGN	BEST COST	APRC	AOPC	ASUNA	ACUNA
11	1	10	1	1499338	446000	1810000	.470413E+08	-.329225E+07

N1	M1	R1	DESIGN	BEST FEASIBLE COST	APRC	AOPC	ASUNA	ACUNA
12	1	11	1	1572653	447273	1710000	.493837E+08	0.354311E+05

Figure 14. Validation Results from the SESD/SEMD Enumerator.

to model a SESD problem. For example, to use the MREAL1 optimizer on a SESD problem, the variable representing the number of REAL systems to be modeled (NREALS) would be set equal to one, and the variable representing the number of equipment designs (S(1)) would be set equal to one.

The MREAL1 model can also handle the single equipment, multiple design (SEMD) problem. In this case, the variable NREALS in the optimizer would be set equal to one, while S(1) would be set greater than one and equal to the number of candidate equipment designs in the REAL system. The same modifications to the MREAL1 enumerator which enabled it to handle the SESD problem allow it to handle a SEMD problem. For the multiple equipment, single design (MESD) problem, the variable NREALS in the optimizer would be set equal to the number of REAL systems in the problem. The variable S(i) would be set equal to one for each of the REAL systems.

Use of the MREAL1 Enumerator

The MREAL1 enumerator served several functions besides the validation of the optimizer's ability to compute values of the objective and and constraints functions. It served to determine if the optimizer was locating true N_2 discrete local minima. In this way, the enumerator was extremely helpful in determining how well the various nonlinear programming techniques were working in the optimizer.

The enumerator also served to provide a capability to do a manual steepest descent type of search. In this mode, the enumerator takes a specified starting point and examines all points in the adjacent N_2 neighborhood. If a better feasible point is found, it becomes the new specified starting point. The process is repeated until a point is found to be an N_2 discrete local minimum.

Results from the MREAL1 Enumerator and Optimizer

Both the optimization technique and the enumeration technique were used to locate feasible solutions to the MREAL1 test problem. Between the two, thirty one feasible N_2 discrete local minima were found. These points and their respective expected total annual equivalent costs are shown in Figure 15 on page 125. The reader should note that there may be more feasible discrete local minima in the MREAL1 test problem than are shown in Figure 15 on page 125. This is due to the fact that neither the enumerator or the optimizer were used to explore the entire feasible region of the MREAL1 test problem and is also an artifact of the discrete nature of the MREAL1 model. Only those portions of the feasible region which appeared to lead to lower objective function values were investigated. Since this objective function appears to be well behaved, it seems unlikely that there will be a feasible local minima better than those shown, yet having adjacent points whose objective function values are much higher than those shown.

The MREAL1 optimizer's stopping routine[57] also identified three additional points as feasible discrete local minima. Since the use of the gradient projection method as a stopping procedure does not guarantee that the resulting point will be a discrete local minimum, these three points turned out not to be such. All three were feasible, but there was at least one better feasible point in the N_2 neighborhood of each. These three points are listed in Figure 16 on page 126.

[57] The stopping routine uses the failure of the gradient projection method to find an improved feasible solution as the termination criteria.

Bus Design 1 and Railcar Design 1:

N_1	M_1	R_1	N_2	M_2	R_2	TOTAL COST
12	1	11	16	1	25	$3,513,817
12	1	11	15	1	14	$3,544,812
13	1	14	16	1	25	$3,616,621
13	1	14	15	1	14	$3,647,616

Bus Design 1 and Railcar Design 2:

N_1	M_1	R_1	N_2	M_2	R_2	TOTAL COST
12	1	11	16	1	30	$3,521,813
12	1	11	15	1	23	$3,521,979
12	1	11	15	1	25	$3,523,101
12	1	11	15	1	20	$3,526,148
12	1	11	15	1	18	$3,527,292
13	1	14	16	1	30	$3,624,617
13	1	14	15	1	25	$3,625,905
13	1	14	15	1	20	$3,628,952
13	1	14	15	1	18	$3,630,096

Bus Design 2 and Railcar Design 1:

N_1	M_1	R_1	N_2	M_2	R_2	TOTAL COST
12	1	15	16	1	25	$3,396,242 **
13	1	18	16	1	25	$3,492,733
12	1	15	15	1	14	$3,427,237
11	1	14	16	1	25	$3,427,346
12	2	20	16	1	25	$3,508,167
14	1	20	16	1	25	$3,594,789

Bus Design 2 and Railcar Design 2:

N_1	M_1	R_1	N_2	M_2	R_2	TOTAL COST
12	1	15	16	1	30	$3,404,238
12	1	15	15	1	23	$3,404,404
12	1	15	15	1	25	$3,405,526
12	1	15	15	1	20	$3,408,573
12	1	15	15	1	18	$3,409,717
12	1	15	15	1	15	$3,417,569
13	1	18	15	1	25	$3,502,017
13	1	18	16	1	30	$3,500,729
13	1	18	15	1	20	$3,505,064
13	1	18	15	1	18	$3,506,208
12	2	20	16	1	30	$3,516,163

Figure 15. Feasible Discrete Local Minima of the MREAL1 Test Problem

Bus Design 1 and Railcar Design 1:

N_1	M_1	R_1	N_2	M_2	R_2	TOTAL COST
11	2	11	16	1	25	\$3,535,562
12	2	15	16	1	25	\$3,621,940

Bus Design 2 and Railcar Design 1:

N_1	M_1	R_1	N_2	M_2	R_2	TOTAL COST
11	2	14	16	1	25	\$3,427,346

Figure 16. False Feasible Discrete Local Minima from the MREAL1 Optimizer

Sample MREAL1 Enumerator Output

Appendix D contains a sample enumeration output for the MREAL1 test problem. The first page of the output contains header information which consists of a banner and the coordinates of the specified initial point. In this case, the initial point represents the choice of 12 buses, 2 bus repair facilities, bus retirement at 20 years, 16 railcars, 1 railcar repair facility, and railcar retirement at 30 years. The enumerator, in this case, has computed the values of all system performance measures and constraint functions for all points in the N_2 local neighborhood of the specified initial point.

The listing of results follows the header information. Each column of the table of results is labeled as follows:

N1 The number of buses in the bus system.

M1 The number of bus repair facilities.

R1 The age at which buses will be retired from service.

N2 The number of railcars in the railcar system.

M2 The number of railcar repair facilities.

R2	The age at which railcars will be retired from service.
DESIGN(1)	The number of the bus design currently under consideration. In this case, it is bus design number 2.
DESIGN(2)	The number of the railcar design currently under consideration. In this case, it is railcar design number 2.
TOTAL COST	Expected total annualized MREAL system cost.
FLAG1	A feasibility indicator. 0 indicates feasibility. 1-5 indicate infeasibility due to, at least, the constraint indicated by the number.
APRC	The total annual cost of procurement of buses and railcars to replace retiring buses and railcars.
AOPC	The total annual cost of operating the buses, railcars, and their associated repair facilities.
ASUNA(1)	The scaled value of the slack in the system unavailability constraint for the bus system. A negative value indicates infeasibility.
ASUNA(2)	The scaled value of the slack in the system unavailability constraint for the railcar system. A negative value indicates infeasibility.
ACUNA(1)	The scaled value of the slack in the catastrophic unavailability constraint for the bus system. A negative value indicates infeasibility.
ACUNA(2)	The scaled value of the slack in the catastrophic unavailability constraint for the railcar system. A negative value indicates infeasibility.

Following the table of results is a recapitulation indicating which point had the best cost, regardless of feasibility, and which point had the best feasible cost.

Sample MREAL1 Optimizer Output

Appendix E contains a sample optimization output for the MREAL1 test problem. The first page of the output contains the banner. It is followed by a line showing the current design combination under consideration. This line is repeated every time a new starting point is chosen by the starting point selection loop. Following this line are three lines showing the current starting point,[58] the initial penalty parameters, and the initial Lagrange multipliers, respectively.

The next section of the sample output traces the search procedures contained in the optimizer. There are two header lines which are very similar to the header lines used in the MREAL1 enumerator. The decision variables are in a slightly different order, and, instead of FLAG1, there is a column titled FLAGC. This column contains six 0-1 indicator values corresponding to each of the six constraints. A zero indicates feasibility, while a one indicates infeasibility.

The next 43 lines of output trace the search conducted by the Inner Loop as it performs the unconstrained optimization of the augmented Lagrangian penalty function using Rosenbrock's search procedure and integer Fibonacci line searches. Following these lines is a message line which indicates that Rosenbrock's search procedure has stalled and that the optimizer will go to subroutine LOCMIN after displaying the current best solution. The following 7 lines display the activity occurring in LOCMIN using Rosen's gradient projection method as a stopping rule. After displaying 7 more search points at the top of the second page of Appendix E, LOCMIN gives up. It has failed to locate an improved feasible solution. Therefore, it is assumed that a feasible discrete local minimum has been found.

On the second page of Appendix E the start of a search from a second starting point is found. Note that the initial base point line is labeled "FROM INIT1". INIT1 generates the "upper" search point

[58] Labelled "initial base point".

in the starting point selection loop. On the third page of the appendix is the third point generated by the starting point selection loop. This point is generated by INIT2, the "lower" starting point generator.

Following the information about the INIT1 base point is the output from the Inner Loop. After 45 search points, Rosenbrock's procedure stalls and subroutine LOCMIN is called. Note that, this time, LOCMIN returns immediately with return code 3. This indicates that LOCMIN found an improved feasible solution immediately. Subroutine OUTER is called to update the penalty parameters and the Lagrange multipliers and the optimizer returns once again to the Inner Loop.

After the optimizer has performed these computations for all possible design combinations, a summary table is printed on the last page of the output. The most important portions of this table are reproduced in Figure 17 on page 130. Note that the best solution found is significantly less costly than two of the design combinations, but is fairly close in value to the cost for bus design 2 and railcar design 2 (design combination 2-2). This suggests that the system managers for the Metropolitan Transit Authority should concentrate their decision making efforts on design combinations 2-1 and 2-2.

Modeling Preventive Maintenance Using MREAL1

While the preventive maintenance problem (PMP) is not explicitly modeled in model MREAL1, it is possible to use MREAL1 to represent part of the decision problem posed by PMP. MREAL1 can be used to evaluate a specific proposal for a preventive maintenance program, if two things are known: the estimated annual preventive maintenance cost per unit of repairable equipment; and the estimated improvement in the MTBF profile to be obtained from the preventive maintenance. The following example makes use of the MREAL1 test problem to illustrate this point.

Optimal Solution for each Design Combination:

Bus Design	Railcar Design	N1	M1	R1	N2	M2	R2	Total Cost
1	1	12	1	11	16	1	25	$3,513,818
1	2	12	1	11	16	1	30	$3,521,814
2	1	12	1	15	16	1	25	$3,396,242
2	2	12	1	15	15	1	23	$3,404,404

The Optimal Solution to Problem MREAL1 (Test) is:

Bus Design	Railcar Design	N1	M1	R1	N2	M2	R2	Total Cost
2	1	12	1	15	16	1	25	$3,396,242

Detailed Summary of the Optimal Solution:

System	Annual Replacement Capital Cost	Annual Operating Cost	Expected Number of Shortages	Probability of N or More Shortages	N
Bus:	$128,000.00	$1,170,000.00	0.005507	0.00008433	3
Railcar:	$224,000.00	$1,250,000.00	0.002285	0.00000000	5
Total:	$352,000.00	$2,420,000.00			
Budget:	$600,000.00	$3,000,000.00			

Figure 17. Summary of Results from the MREAL1 Optimizer.

Suppose that the Metropolitan Transit Authority has a specific preventive maintenance program in mind for each design of bus and each design of railcar. Suppose the cost of preventive maintenance is as shown for each design, in Figure 18 on page 134, Figure 19 on page 135, Figure 20 on page 136, and Figure 21 on page 137, respectively. In this case the cost is included in the annual operating cost of the repairable equipment. In some situations, it might be more appropriate to include the total annual cost of preventive maintenance in the repair facility operating cost. However, since preventive maintenance costs tend to increase with increasing equipment population size, these costs were included in the equipment operating cost instead.

The Metropolitan Transit Authority should consider running a preventive maintenance program only if it will produce some benefits. One quantifiable benefit of a preventive maintenance program is an improvement in the mean time between failure (MTBF) profile. Preventive maintenance should produce some sort of increase in MTBF values across most of the possible life of the equipment. For bus design 1, a 30% increase was assumed. For bus design 2, a 25% increase was assumed. For railcar design 1, a 20% MTBF improvement was assumed. And for railcar design 2, 15% was assumed.[59] These improvements are reflected in the MTBF profiles shown in Figure 18 on page 134, Figure 19 on page 135, Figure 20 on page 136, and Figure 21 on page 137. Note that no change in the MTTR profiles is shown. Preventive maintenance has less effect upon MTTR than upon MTBF. For the purposes of this example, it was assumed that preventive maintenance has a negligible effect upon the MTTR profile.

These changes, in MTBF profile and in the annual repairable equipment operating cost, were made in the version of the MREAL1 optimizer which produced the sample output in appendix E. The optimizer was then run to see if it could identify a feasible solution to the preventive maintenance problem which would lead to a lower expected total annualized system cost than was obtained without the preventive maintenance program. The results are shown in Figure 22 on page 138.

---

[59] Of course, in a real problem, these improvements might be obtained from historical preventive maintenance and MTBF profile data for similar repairable equipment.

In this case, the optimizer could not find a better solution using preventive maintenance. However, given the uncertainty in the various parameter estimates used in the MREAL1 test problem, this cost does not appear to be significantly higher than the best cost found for the MREAL1 test problem without preventive maintenance.

Summary

This dissertation has defined a broad class of systems which are in several respects similar to classical inventory systems. A hierarchy of some (there may be more) of the important components of this multiple repairable equipment and logistic (MREAL) system was discussed. Some of the interactions between these components were also discussed.

The important idea of attempting to create holistic models of Multiple REAL systems was described. In keeping with this idea, a model called MREAL1 was formulated from a subset of the components of the MREAL model. An attempt was made to include components from all levels of the hierarchy, and to model components which traditionally have not been modelled together.

It was demonstrated that a model such as MREAL1 can be created. Further, it was shown that reasonable solutions could be obtained at a relatively low cost by adapting combinations of existing nonlinear optimization methods to the problem. These adaptations included the use of discrete Fibonacci line searches, combining the steepest descent method with Rosenbrock's search technique, and the use of the gradient projection method of Rosen as a final stopping rule. The model MREAL1 was also adapted to solve a preventive maintenance problem, thus effectively extending the model to additional components of the MREAL system hierarchy.

The following chapter describes some of the work left undone. It indicates a number of areas which should be investigated in pursuit of more effective and useful models of MREAL systems. Among these areas are improvements in the representation of REAL system queuing behavior, representation of non-steady state conditions, investigation of alternative optimization methods, and the design and execution of experiments with actual MREAL systems.

PARAMETER	SYMBOL	VALUE
Annual cost of preventive maintenance per bus	--	$2,000
Annual operating cost of a bus including preventive maintenance.	o_{11}	$102,000

Age Functions for MTBF ($\lambda_{11}(t)$) and MTTR ($\mu_{11}(t)$):

AGE (years)	MTBF (years)	MTTR (years)
1	0.52	0.010
2	1.04	0.011
3	1.30	0.011
4	1.43	0.012
5	1.43	0.013
6	1.30	0.015
7	1.17	0.015
8	1.04	0.015
9	0.91	0.015
10	0.91	0.015
11	0.91	0.017
12	0.65	0.019
13	0.39	0.020
14	0.39	0.021
15	0.26	0.022

Figure 18. Design Parameters for Bus Design 1 with Preventive Maintenance.

PARAMETER	SYMBOL	VALUE
Annual cost of preventive maintenance per bus	--	$1,000
Annual operating cost of a bus including preventive maintenance.	o_{12}	$91,000

Age Functions for MTBF ($\lambda_{12}(t)$) and MTTR ($\mu_{12}(t)$):

AGE (years)	MTBF (years)	MTTR (years)
1	0.625	0.012
2	1.000	0.011
3	1.250	0.012
4	1.375	0.013
5	1.499999	0.013
6	1.499999	0.014
7	1.499999	0.015
8	1.375	0.015
9	1.250	0.015
10	1.250	0.015
11	1.124999	0.015
12	1.000	0.016
13	0.875	0.016
14	0.875	0.017
15	0.625	0.020
16	1.749999	0.022
17	0.375	0.023
18	0.0625	0.025
19	0.0625	0.027
20	0.0625	0.029

Figure 19. Design Parameters for Bus Design 2 with Preventive Maintenance.

PARAMETER	SYMBOL	VALUE
Annual cost of preventive maintenance per railcar	--	$2,000
Annual operating cost of a railcar including preventive maintenance.	o_{21}	$72,000

Age Functions for MTBF ($\lambda_{21}(t)$) and MTTR ($\mu_{21}(t)$):

AGE (years)	MTBF (years)	MTTR (years)
1	0.48	0.004
2	1.2	0.002
3	1.439999	0.001
4	1.799999	0.001
5	1.799999	0.001
6	1.799999	0.001
7	1.799999	0.001
8	1.799999	0.001
9	1.799999	0.001
10	1.679999	0.002
11	1.679999	0.002
12	1.679999	0.002
13	1.679999	0.002
14	1.679999	0.002
15	1.559999	0.002
16	1.439999	0.004
17	1.439999	0.004
18	1.439999	0.004
19	1.439999	0.004
20	1.2	0.004
21	0.96	0.009
22	0.96	0.009
23	0.96	0.009
24	0.96	0.009
25	0.96	0.009

Figure 20. Design Parameters for Railcar Design 1 with Preventive Maintenance.

PARAMETER	SYMBOL	VALUE
Annual cost of preventive maintenance per railcar	--	\$500
Annual operating cost of a railcar including preventive maintenance.	o_{22}	\$65,500

Age Functions for MTBF ($\lambda_{22}(t)$) and MTTR ($\mu_{22}(t)$):

AGE (years)	MTBF (years)	MTTR (years)
1	0.575	0.003
2	1.15	0.002
3	1.494999	0.001
4	1.839999	0.001
5	1.839999	0.001
6	1.839999	0.001
7	1.839999	0.001
8	1.839999	0.001
9	1.839999	0.001
10	1.839999	0.001
11	1.724999	0.001
12	1.724999	0.001
13	1.724999	0.001
14	1.724999	0.001
15	1.724999	0.001
16	1.609999	0.002
17	1.609999	0.002
18	1.609999	0.002
19	1.609999	0.002
20	1.609999	0.002
21	1.494999	0.003
22	1.494999	0.003
23	1.494999	0.003
24	1.494999	0.003
25	1.494999	0.003
26	1.379999	0.006
27	1.15	0.006
28	1.034999	0.006
29	0.92	0.006
30	0.805	0.006

Figure 21. Design Parameters for Railcar Design 2 with Preventive Maintenance.

Optimal Solution for each Design Combination:

Bus Design	Railcar Design	N1	M1	R1	N2	M2	R2	Total Cost
1	1	12	1	14	16	1	25	$3,559,345
1	2	12	1	14	15	1	25	$3,531,790
2	1	12	1	18	15	1	15	$3,445,398
2	2	12	1	18	15	1	25	$3,404,410

The Optimal Solution to Problem MREAL1 (Test) is:

Bus Design	Railcar Design	N1	M1	R1	N2	M2	R2	Total Cost
2	2	12	1	18	15	1	25	$3,404,410

Detailed Summary of the Optimal Solution:

System	Annual Replacement Capital Cost	Annual Operating Cost	Expected Number of Shortages	Probability of N or More Shortages	N
Bus:	$106,666.62	$1,182,000.00	0.004828	0.00006548	3
Railcar:	$252,000.00	$1,112,500.00	0.016086	0.00000000	5
Total:	$358,666.62	$2,294,500.00			
Budget:	$600,000.00	$3,000,000.00			

Figure 22. Results from MREAL1 Optimizer with Preventive Maintenance Alternative.

Chapter VII

FURTHER RESEARCH DIRECTIONS

It is abundantly clear that the MREAL system is a significant and very expensive component of nearly every business organization and government agency. Effective management and control of MREAL systems by decision makers is thus an important objective for these organizations. Since the primary purpose of the field of Industrial Engineering and Operations Research involves the application of the scientific method and of mathematical techniques to the problems of human decision makers, the study of MREAL systems should be of considerable interest.

As this dissertation has shown, however, MREAL systems are extremely complex. These systems do not lend themselves to the elegant or simple application of mathematical models. Consequently, the development of models which handle these complexities has been slow. This dissertation is a small contribution towards this development. Much more research and experimentation must be done before useful, user-friendly decision support systems will be widely available to managers of MREAL systems.

The work needed to develop these user-friendly decision support systems falls in four broad areas. The first involves the development of new and/or improved mathematical models which yield useful and realistic predictions of the behavior of real MREAL systems. The second area involves the creation of the mathematical techniques and computer algorithms necessary to solve these models in a rapid, cost effective manner. Thirdly, a significant amount of experimentation and testing needs to be performed with the models and solution techniques. Since MREAL systems are so complex, an important purpose of this experimentation would be to compare model predictions to the behavior of actual MREAL systems. Finally, and perhaps most importantly, suitable ways of implementing these models and techniques in decision support systems must be found. The following sections of this chapter discuss some of the specific problems in each of these areas.

Some Important Modeling Needs

The classification hierarchy discussed in Chapter III and in Moore and Fabrycky [1985] is one possible, systematic structure for the modeling of MREAL systems. A comparison of the structure contained in existing models such as MREAL1 or METRIC (Sherbrooke [1968] and Muckstadt [1973]) quickly reveals where additional modeling effort may be required. However, a number of specific issues should be mentioned.

1. The bane of probabilists and operations researchers everywhere, transient system behavior is very difficult to model in complex systems. Since MREAL systems have a number of imbedded stochastic components, an important unanswered question concerns the degree to which such transient behavior can be simplified in or ignored by models of MREAL systems, without significantly degrading the accuracy of the model's predictions.In addition, more computable models of transient stochastic behavior are needed, in order to handle those situations where transient behavior cannot be ignored.

2. Since there will be MREAL systems in which transient stochastic behavior dominates, some future models must realistically capture this behavior in their mathematical formulation. For these models to be of practical use, probabilists must find mathematical formulations for the transient stochastic behavior of REAL systems which are more computable and tractable than existing models.

3. Since model MREAL1 makes use of the finite source queuing model, the stochastic behavior of the MREAL system depends upon $\rho(R_i)$. As $\rho(R_i) = \hat{\lambda}(R_i)/\hat{\mu}(R_i)$, an infinite number of combinations of $\hat{\lambda}(R_i)$ and $\hat{\mu}(R_i)$ can therefore yield the same stochastic behavior. For example a $\hat{\lambda}(R_i)$ of 2 and $\hat{\mu}(R_i)$ of 1 yields the same stochastic behavior as a $\hat{\mu}(R_i)$ 2 and a $\hat{\lambda}(R_i)$ of 4. In order to obtain improvements in the queuing behavior of the system, $\rho(R_i)$ must be made smaller. This may be accomplished to a considerable extent in two different ways:

   a. By reducing $\hat{\lambda}(R_i)$, i.e. by increasing MTBF.

   b. By increasing $\hat{\mu}(R_i)$, that is, by decreasing the MTTR.

   Historically, most REAL system owners have chosen to try to increase MTBF (improve reliability) by changing the design of the repairable equipment. The decision to increase MTBF rather than decrease MTTR (improve maintainability) has been primarily based upon the reasoning that it is better to make the repairable equipment more reliable than it is to make it easier to fix. Certainly the soldiers who operate repairable combat equipment would prefer reliability over maintainability. There lives depend more directly upon the reliability than upon the maintainability.

   This method of choosing how to spend an equipment design budgets may lead to the best choice. However, it is also possible that design budgets would be better spent improving maintainability, once some specified minimum reliability has been achieved. A future research direction would be to examine the relevant costs and benefits of improving reliability, and to compare these to the costs and benefits of improving maintainability. It might turn out that the best way to spend a large portion of the procurement budget for a repairable equipment is on a very good level of maintainability, i.e. to minimize MTTR. This research direction takes on special significance in light of the recent efforts of the military to design repairable equipment to meet a specified procurement cost. In this situation, a decision must be made, consciously or by default, about the "amount" of reliability and the "amount" of maintainability to purchase.

4. MREAL systems are made up of different populations of repairable equipment. Within a population of equipment, no two units have exactly identical stochastic behavior. A unit of repairable equipment is made of components, some repairable, some replaceable. No two identical components have exactly identical stochastic characteristics. No wonder probabilists have frequent nightmares. The result of this situation is that many assumptions (which differ from reality to some degree) must be made about these stochastic characteristics to obtain reasonably computable steady state models. New, more realistic models of the stochastic components of MREAL systems need to be developed. Gross and Ince [1981] is an example of this type of research.

5. Few MREAL systems have truly time invariant demands for each type of equipment. Instead, the demand for a specific equipment may behave stochastically, may behave deterministically, but change at known points in time, or it may behave fuzzily, i.e. the MREAL system manager does not know exactly what the demand is. The modeling of a military MREAL system under the transition from peacetime to wartime would have to include the modeling of a significant, nearly instantaneous change in equipment demand. Future models must find ways to represent this behavior, or else demonstrate that doing so will not improve the accuracy of model predictions.

6. Model MREAL1 assumes that the MREAL system is always in the steady state with respect to equipment procurement. This is an assumption which is usually violated by actual MREAL systems. Future models need to represent, and even aid the decision maker in selecting, the equipment procurement and disposal profiles. Should block procurement be used, or should the units be procured a few at a time over many time periods? Mackliet's [1984] work is one approach toward dealing with equipment procurement and disposal profiles.

7. In the real world, the maintenance channels in many MREAL systems are not strictly segregated by equipment type. The consequent channel sharing may have a significant effect upon the stochastic components of the system. Baskett et al. [1975] treat a simplified version of this situation by modeling a queue with different classes of customers served by a single server who performs simultaneous service.

8. Most large organizations staff their maintenance channels according to some type of echelon scheme in which the channels are set up in classes, with each class performing different kinds of maintenance. Hinger [1973] addressed a single REAL system with two classes of maintenance channels. He did not, however, provide a model which would assist in determining how many classes of maintenance channel to establish. Rather the model only helps to determine how many channels to operate in each class. Both determinations are useful products of the MREAL systems modeling process. Clearly more research must be performed to model multiple classes of maintenance channels.

9. The MREAL1 model assumes that changes in the stochastic failure and stochastic repair behavior of an item of equipment can be entirely attributed to the calendar age of the equipment. It is obvious that this does not always hold. The ρ-profile for a unit of equipment may more appropriately be a function of operating time, service (on/off) cycles, or work (production) rates. Or it may be a function of the time a unit may periodically spend in some kind of storage. Here is yet another additional research need.

10. A very serious shortcoming of the MREAL1 model is the need to explicitly enumerate each design combination and perform an optimization on it. It may turn out that sets of possible equipment designs can be ordered in some reasonable way. For example, an ordering according to unit procurement cost, or an ordering by some summary measure of the ρ -profile could be done. Such an ordering might allow the design choices to be treated in the MREAL system model more like the other decision variables. This would result in significant savings of computation time.

11. In some REAL systems, the units of equipment may spend a significant fraction of their life under some type of warranty. Since this will affect the way in which maintenance is performed, a corresponding model change is required.

12. Some consideration may, in the future, need to be given to additional measures of system effectiveness. Most existing models measure MREAL or REAL system effectiveness in terms of cost, availability and/or budget. Future decision makers may wish to include other measures of system effectiveness. Some possibilities include energy consumption (not measured strictly as a dollar cost), environmental impact, disposability and labor intensity.[60]

13. Most work which models the stochastic behavior of a maintenance channel makes the assumption that this behavior depends upon the design of the unit of equipment. Work in the area of human factors provides ample evidence that this is only partly true. The other factor which significantly affects time to repair is the design of the maintenance channel itself. Good workstation layout, effectively designed tools, and simplified administrative procedures can significantly decrease mean time to repair, and can also affect the repair time distribution. The joint impacts of equipment and maintenance channel designs on repair time distributions must be studied further so that these impacts can be appropriately handled in future MREAL system models.

14. For many MREAL system managers, the manner in which MREAL1 handles shortage costs would be unacceptable. The difficulty arises when the modeler demands that the manager provide a dollar value on the cost of a unit-day of shortage. In some MREAL systems, a unit-day of shortage causes little added dollar cost, but creates considerable human inconvenience, delay, or perhaps even death. These are shortage "costs" which are very difficult to measure in dollars. For this reason shortages are frequently treated as system constraints so that the manager can specify an acceptable level of shortages without having to address the

[60] Labor intensity of a REAL system is a measure of the quantity, and skills, of the humans required to operate the repairable equipment and logistic support in the system.

costing problem. Additional research is needed to deal with this problem. One line of research ought to examine how non-dollar shortage costs can be treated as part of the objective function, perhaps through improved data collection on non-dollar shortage costs or through goal programming formulations in which dollar and non-dollar objective are modeled. Another line of research might examine the possibility of modeling earning functions instead of penalizing shortages. In such an objective function, the system is rewarded for the amount of equipment which can be made available.

15. Models of MREAL systems, by their nature, contain uncertainty. This uncertainty stems from the stochastic components of the system, from the error introduced by the estimation of costs and other system parameters, and from the fact that these models represent a simplified version of reality. The MREAL1 model operates strictly on point estimates of costs and system parameters. The optimized model produces point estimates of costs and unavailabilities. It would be useful if future models of MREAL systems were able to estimate the variability of costs and unavailabilities. This would allow the construction of interval estimates of costs and of other measures of system effectiveness.

16. Effective modeling of MREAL systems cannot proceed without effective communications. Such communications cannot occur unless the modelers are using a common language. Since the modeling of MREAL systems is primarily mathematical, this common language must include the definitions of terms and the definitions of symbols.

    The human desire for the ersatz immortality achieved by having one's definitions accepted in general use is understandable. But must this be achieved at the price of forcing all modelers to learn new names for old ideas over and over again? The development of models of MREAL systems, indeed the concept of an MREAL system is in a very early and formative stage. Now is the ideal time to attempt to bring the current researchers into agreement on term definitions, symbol definitions and perhaps basic problem (not model) structure. Perhaps it would be appropriate to create a committee (not another one!) with representation from (and responsible to) the major professional societies with interest in MREAL systems. The goal of such a committee would be to achieve agreement on the names of terms and their definitions, symbols and their definitions, to sell these terms and symbols to the participating societies, and to produce and publish the dictionary needed to back up this effort.

Some Needs for Alternate Optimization Techniques

Existing and future models of MREAL systems will be useless unless effective, relatively inexpensive techniques exist to solve these models. Since MREAL1 is, and many other models will be, discrete and nonlinear, and since no single discrete, nonlinear optimization technique has risen above all the others, the effectiveness of such techniques must be compared, perhaps for each MREAL system model. The following list shows some discrete nonlinear programming methods which ought to be examined for effectiveness and/or applicability to models similar to MREAL1.

1. The random search methods (see Brooks [1958], Conley [1980], Conley [1981] and Friedman and Savage [1947]).

2. Confidence limit stopping techniques (see Derigs [1985] and McRoberts [1971]).

3. Simplex/complex optimization (see Box [1966], Fox and Liebman [1981], Glankwahmdee, Liebman and Hogg [1979] and Nelder and Mead [1965]).

4. Other sequential unconstrained minimization techniques and improvements of the augmented Lagrangian penalty function method.

5. The applicability of goal programming to the combining of measures of system performance in the objective functions of future models.

6. The use of existing optimization techniques for multiple equipment, single design (MESD) problems to obtain lower bounds to be used to fathom branches in a branch and bound partial enumeration of the combinatorial programming portion of the MREAL system problem.

7. The applicability of the most recent branch and bound and dynamic programming techniques should be periodically examined for use on the combinatorial programming portion of the MREAL system problem.

In addition to examining alternative discrete nonlinear programming techniques, three other issues which affecting modeling and optimization should be investigated. The first of these issues concerns the possibility of imbedding fast Monte Carlo simulations into search type optimizations. If this could be accomplished, then more realistic representations of the stochastic components of an MREAL system could be modeled. The challenge of this potential technique is to make computer code for the simulations sufficiently rapid, as the simulation would be called frequently.

The second issue deals with the limitations on numerical accuracy when a computer is used to optimize an MREAL model which contains an imbedded finite queuing model. Current and foreseeable computing machines have upper limits on the size of the numbers which can be stored in them. This limit could easily be exceeded when modeling real MREAL systems. While this problem can be solved by extensive additional coding and attendant increases in computing costs, it may be better to model an approximation to the finite queuing model using the infinite queuing model.

The last issue is a result of the rapidly decreasing cost of binary computational power, and the rapidly increasing computational speeds. As these speeds increase, and costs decrease, a point will be reached at which total enumeration of all possible solutions will be possible for real size MREAL systems, and may be relatively inexpensive. The advantages of using complete enumeration include:

1. the determination of an exact global optimum.

2. the need for fewer simplifying assumptions.

3. easier computer programming.

An important future research effort would be to estimate the cost of using complete enumeration as an optimization technique minus the value of the improvement in the optimal solution.[61] A comparison could then be made with the costs of existing algorithmic optimization techniques.

Experimentation and Testing

First and foremost among the recommendations for further research is the need to verify the accuracy and usefulness of MREAL system models such as MREAL1. This should be done by comparing model predictions against reality. A suitable set of organizations would be chosen as test benches. The model to be tested would be implemented at the organization for a period of years, and its performance monitored. It might also be possible to locate sufficient historical data from an organization to provide a means for comparing model predictions with actual performance after the fact.

A second class of experiments involves the comparison of existing models with each other. The system parameters for each model would necessarily have to be identical, although one model might choose to ignore some information which the other made use of. For this comparison to be useful, however, at least some output information must be common to both models.

[61] Since existing solution techniques for optimizing models of MREAL systems only approximate the optimal solution, the cost of switching to a complete enumeration technique should be examined in light of the potential cost savings which may result from obtaining an exact optimal solution.

A similar class of experiments needs to be performed to compare the performance of alternative optimization schemes. This type of experiment must be conducted using the same model with each optimization method.

Another important result of the comparison of reality with a particular model would be a determination concerning the separability of various components of the model. The current practice for dealing with MREAL systems in most businesses and organizations is to separate the various components of the system and attempt to optimize each by itself. Maintenance is optimized separately from the spare parts inventory. Each REAL system is optimized separately, etc. It is entirely possible that these separate optimizations produce a globally optimal solution for the MREAL system. To determine whether or not this is true, we must have valid results from global MREAL system models to compare with the results from the separate models.

Finally the acceptance by system managers of the predictions from MREAL system models is of critical importance. The building of MREAL system models and optimization techniques will be a strictly academic endeavor if the models are not useful to MREAL system managers and owners. The development and implementation of such models must ultimately include mechanisms for obtaining feedback on acceptance and usefulness from the users.

Development of MREAL Systems Decision Support

Models of MREAL systems will not make decisions. They can only provide a human system manager with insights which can be used to aid decision making. Few such decision makers are willing to learn the intricacies of building and optimizing such a model. They demand a decision support system which is very user friendly. The current MREAL1 model, for example, is far from being user friendly or a complete decision support system. The following list shows some of the issues which must be addressed on the way to creating a decision support system.

1. User friendly system parameter input interfaces must be designed.

2. Appropriate methods of displaying output information must be developed.

3. Response times (and costs) for the decision support system must be appropriate for the organization and the MREAL system.

4. The ways in which the user may intervene in the activities of the decision support system must be carefully designed and user friendly. Such intervention might include sensitivity analysis and the evaluation (by the model) of specific suggestions from the user.

5. Consideration might be given to the incorporation in the decision support system of techniques from decision analysis and artificial intelligence.

A large number of recommendations and suggestions for further research have been presented in this chapter. These recommendations have dealt with the general areas of modeling improvement, optimization improvements, experimentation and the creation of suitable decision support systems. As there is more research work here than could be performed by several investigators over an entire career, it is hoped that this dissertation, and these ideas, will interest additional investigators in studying Multiple Repairable Equipment and Logistic Systems.

REFERENCES

1. Ascher, H., "Development of Systems Reliability Models from Basic Physical/Mathematical Concepts", *Generic Techniques in Systems Reliability Assessment - Proceedings*, NATO Advanced Study Institute Program, Noordhoff-Leyden (1973):11-28.

2. Balachandran, S., "Dynamic Scaling of Nonlinear Programs and Its Implementation Within the Augmented Lagrangian Penalty Function Technique" (Doctoral Dissertation, VPI&SU, 1984).

3. Bamforth, T., "Integrated Logistics Support: Lessons Learned of a Naval Surface Weapons System" (unpublished paper, Department of the Navy, 1984).

4. Barzily, Z. and Gross, D., "Transient Solutions for Repairable Item Provisioning" (Technical Paper Serial T-390, Program in Logistics, The George Washington University, 1979).

5. Baskett, F., Chandy, K.M., Muntz, R.R. and Palacios, F.G., "Open, Closed, and Mixed Networks of Queues with Different Classes of Customers," *Journal of the Association of Computing Machinery* 22 (1975):248-260, cited in Gross and Ince [58].

6. Bazaraa, M.S. and Shetty, C.M., *Nonlinear Programming: Theory and Algorithms*, New York, John Wiley & Sons, 1979.

7. Beightler, C.S., Phillips, D.T. and Wilde, D.J., *Foundations of Optimization*, Englewood Cliffs, NJ, Prentice-Hall, Inc., 1979.

8. Berlucchi, E.A., "LOR... The Maintenance Engineer's Tool for Reducing Logistics Costs", *Proceedings*, 17th Annual International Logistics Symposium, Boston, MA, August, 1982.

9. Bertsekas, D.P., "Multiplier Methods: A Survey", *Automatica*, 12 (1976):133-145.

10. Beveridge, G.S.G. and Schechter, R.S., *Optimization: Theory and Practice*, New York, McGraw-Hill Book Co., 1970.

11. Black, G. and Proschan, F., "On Optimal Redundancy", *Operations Research* 7 (September-October 1959):581-588.

12. Blanchard, B.S., *Logistics Engineering and Management*, Englewood Cliffs, NJ, Prentice-Hall, 1986.

13. Blanchard, B.S. and Fabrycky, W.J., *Systems Engineering and Analysis*, Englewood Cliffs, NJ, Prentice-Hall, 1981.

14. Box, G.E.P. and Wilson, K.B., "On the Experimental Attainment of Optimum Conditions", *Journal of the Royal Statistical Society, Series B*, 13 (1951):1-45, cited in Brooks [18].

15. Box, M.J., "A New Method of Constrained Optimization and a Comparison with Other Methods", *Computer Journal* 9(1) (May 1966):42-52.

16. Box, M.J., Davies, D. and Swann, W.H., *Non-Linear Optimization Techniques*, Edinburgh, Oliver and Boyd, 1969.

17. Brammer, K.W., "A Repairable Item Transient State Model for Maintenance Requirements Planning" (Master's Thesis, VPI&SU, 1985).

18. Brooks, S.H., "A Discussion of Random Methods for Seeking Maxima", *Operations Research* 6(2) (March-April 1958):244-251.

19. Brooks, S.H. and Mickley, M.R., "Optimal Estimation of Gradient Direction in Steepest Ascent Experiments", *Biometrics*, 17 (1961):48-56.

20. Brown, M., "On the Reliability of Repairable Systems", *Operations Research* 32 (May-June 1984):607-615.

21. Carroll, C.W., "The Created Response Surface Technique for Optimizing Nonlinear Restrained Systems", *Operations Research* 9 (1961):169-184.

22. Chang, E.Y. and Thompson, W.E., "Bayes Analysis of Reliability of Complex Systems", *Operations Research*, 24 (1976):156-168.

23. Claycombe, W.W. and Sullivan, W.G., *Foundations of Mathematical Programming*, Reston, VA, Reston Publishing Co., 1975.

24. Cohen, M.A., Pierskalla, W.P. and Nahmias, S., "A Dynamic Inventory System with Recycling", *Naval Research Logistics Quarterly* 27 (June 1980):289-296.

25. Cooper, M., "Survey of Methods for Pure Nonlinear Integer Programming", *Management Science* (1981):353-361.

26. Conley, W., *Computer Optimization Techniques*, New York, Petrocelli Books, 1980.

27. Conley, W., *Optimization: A Simplified Approach*, New York, Petrocelli Books, 1981.

28. Connelly, M., Manager, Blacksburg Transit System, Blacksburg, VA. Interview, 8 February 1985.

29. Coyle, J.J. and Bardi, E.J., *The Management of Business Logistics*, St. Paul, MN, West Publishing Co., 1984.

30. Crow, L.H., "Reliability Analysis for Complex Repairable Systems", *Reliability & Biometry: Statistical Analysis of Life Length*, Philadelphia, PA, SIAM, 1974.

31. Davis, G.B. and Hoffmann, T.R., *FORTRAN: A Structured, Disciplined Style*, New York, McGraw-Hill Book Co., 1978.

32. Derigs, U., "Using Confidence Limits for the Global Optimum in Combinatorial Optimization", *Operations Research*, 33(5) (September-October 1985):1024-1049.

33. Fabrycky, W.J., Ghare, P.M. and Torgersen, P.E., *Applied Operations Research and Management Science*, Englewood Cliffs, NJ, Prentice-Hall, 1984.

34. Fabrycky, W.J., "Logistics System Design Using Finite Queuing Analysis", *Proceedings*, 1981 International Logistics Congress, San Francisco, CA, April, 1981.

35. Fabrycky, W.J. and Banks, J., "A Hierarchy of Deterministic Procurement-Inventory Systems", *Operations Research* 14 (September-October 1966):888-901.

36. Fabrycky, W.J. and Hart, J.T., "Economic Optimization of a Finite Population System Deployed to Meet a Demand", *Proceedings*, AACE/AIIE Joint Conference, Houston, TX, pp. 35-41, 1975.

37. Fabrycky, W.J., Malmborg, C.J., Moore, T.P. and Brammer, K.W., "The Repairable Equipment Population System Demonstrator", User's Guide and program for IBM-PC, VPI&SU Report for US Navy contract N00039-84-C-0346T, 1984.

38. Falk, H.E. and Rappoport, H.K., "An Optimization Technique for a Multi-time Period Spares Provisioning Problem" (Technical Paper Serial T-408, Program in Logistics, The George Washington University, 1980).

39. Feller, W., *An Introduction to Probability Theory and Its Applications*, New York, John Wiley & Sons, 1968.

40. Fiacco, A.V. and McCormick, G.P., *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, New York, John Wiley & Sons, 1968.

41. Fletcher, R., "An Ideal Penalty Function for Constrained Optimization", *Nonlinear Programming: 2*, (eds. Mangasarian, O.L., et al., 1975):121-163.

42. Fletcher, R., *Optimization*, London, Academic Press, 1969.

43. Fox, D.B. and Liebman, J.S., "A Discrete Nonlinear Simplex Method for Optimized Engineering Design", *Engineering Optimization* 5 (1981):129-149.

44. Friedman, M. and Savage, L.S., "Planning Experiments Seeking Maxima", *Selected Techniques of Statistical Analysis*, (eds. Wisenhart, C., Hastay, M.S. and Wallis, W.S.) New York, McGraw-Hill Book Co., 1947, cited in Brooks [18] and in Beveridge and Schechter [10].

45. Frisch, F.A.P., "Mortality and Spare Parts: A Conceptual Analysis" (draft working paper, Department of the Navy, ELEX 63C, Washington, D.C., May 1983).

46. Frisch, K.R., "The Logarithmic Potential Method of Convex Programming", (memorandum of May 13, 1955, University Institute of Economics, Oslo, Norway, May 1955). Cited in Gill, Murray and Wright [52].

47. Geisler, M.A. and Murrie, B.L., "Assessment of Aircraft Logistics Planning Models", *Omega* 9 (1981):59-69.

48. Geisler, M.A. and Murrie, B.L., "A Survey of Models in the Services to Support Aircraft Logistics Planning During Acquisition" (unnumbered Technical Report, Logistics Management Institute, Washington, DC, 1979).

49. Ghare, P.M. and Taylor, R.E., "Optimal Redundancy for Reliability in Series Systems", *Operations Research*, 17(5) (September-October 1969):838-847.

50. Gill, P.E. and Murray, W., "The Computation of Lagrange Multiplier Estimates for Constrained Minimization", *Mathematical Programming* 17 (1979):32-60.

51. Gill, P.E. and Murray, W., *Numerical Methods for Constrained Optimization*, New York, Academic Press, 1974.

52. Gill, P.E., Murray, W., and Wright, M.H., *Practical Optimization*, New York, Academic Press, 1981.

53. Glankwahmdee, A., Liebman, J.S. and Hogg, G.L., "Unconstrained Discrete Nonlinear Programming", *Engineering Optimization* 4 (1979):95-107.

54. Gottfried, B.S. and Weisman, J., *Introduction to Optimization Theory*, Englewood Cliffs, NJ, Prentice-Hall, 1973.

55. Graves, S.C. and Keilson, J., "System Balance for Extended Logistic Systems", *Operations Research* 31 (March-April 1983):234-252.

56. Greig, D.M., *Optimisation*, London, Longman Group, Ltd., 1980.

57. Gross, D. and Ince, J.F., "Spares Provisioning for Repairable Items: Cyclic Queues in Light Traffic", *AIIE Transactions* 10 (September 1978):307-314.

58. Gross, D. and Ince, J.F., "The Machine Repair Problem with Heterogeneous Populations", *Operations Research*, 29(3) (May-June 1981):532-549.

59. Gross, D. and Kahn, H.D., "On the Machine Repair Problem with Spares under Unequal Failure Rates" (Technical Memorandum TM-66451, Program in Logistics, The George Washington University, 1976).

60. Gross, D., Kahn, H.D. and Marsh, J.D., "Queueing Models for Spares Provisioning", *Naval Research Logistics Quarterly* 24 (December 1977): 521-536.

61. Gross, D. and Miller, D.R., "Multiechelon Repairable Item Provisioning in a Time-Varying Environment Using the Randomization Technique", *Naval Research Logistics Quarterly* 31(3) (September 1984):347-361.

62. Haarhoff, P.C. and Buys, J.D., "A New Method for the Optimization of a Nonlinear Function Subject to Nonlinear Constraints", *Computer Journal* 13(2) (May 1970):178-184.

63. Hadley, G. and Whitin, T., *Analysis of Inventory Systems*, Englewood Cliffs, NJ, Prentice-Hall, 1963. Cited in Hillier and Leiberman [67].

64. Harrison, G.L., "The Acquisition Logistician," *Logistics Spectrum* (Spring 1981):11-12. Cited in Coyle and Bardi [29].

65. Hart, J.T., "A Model for the Optimal Design of a Finite Population Queuing System Deployed to Meet a Demand" (Master's Thesis, VPI&SU, 1971).

66. Hestenes, M.R., "Multiplier and Gradient Methods", *Journal of Optimization Theory and Applications* 4(5) (1969):303-320.

67. Hillier, F.S. and Lieberman, G.J., *Introduction to Operations Research*, 3d ed., San Francisco, Holden-Day, Inc., 1980.

68. Himmelblau, D.M., *Applied Nonlinear Programming*, New York, McGraw-Hill Book Co., 1972.

69. Hinger, C.K., "Selected Sensitivity and Constraint Studies for a Finite Population Queuing System Deployed to Meet a Demand" (Master's Thesis, VPI&SU, 1973).

70. Hock, W. and Schittkowski, K., *Test Examples for Nonlinear Programming Codes*, Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, 1980.

71. Hodgson, G.M., "Reliability and Maintainability in the Air Force", *Air Force Journal of Logistics* VIII (Summer, 1984):10-13.

72. Hooke, R. and Jeeves, T.A., "Direct Search Solution of Numerical and Statistical Problems," *Journal of the Association of Computer Machinery* 8 (1961):212-229. Cited in Bazarra and Shetty [6].

73. Jenson, P.A., "Optimum Repair Level Analysis: Some Extensions", *Logistics* I (1975):87-104.

74. Kabak, I.W., "System Availability and Some Design Implications", *Operations Research*, 17(5) (September-October 1969):827-837.

75. Kline, M.B., "A Survey of Logistic Analysis Models", *Proceedings*, 17th Annual International Logistics Symposium, Boston, MA, August, 1982.

76. Kuester, J.L. and Mize, J.H., *Optimization Techniques with FORTRAN*, New York, McGraw-Hill Book Co., 1973.

77. Land, A. and Powell, S., "Computer Codes for Problems of Integer Programming", *Annals of Discrete Mathematics*, 5 (1979):221-269.

78. Lawrence, S.H. and Schaefer, M.K., "Optimal Maintenance Center Inventories For Fault-Tolerant Repairable Systems", *Journal of Operations Management* 4(2) (February 1984):175-181.

79. Lee, S.M. and Moore, L.J., *Introduction to Decision Science*:, New York, Petrocelli/Charter, 1975.

80. Lehtonen, T., "On the Optimal Policies of an Exponential Machine Repair Problem", *Naval Research Logistics Quarterly* 31(2) (June 1984):173-181.

81. Locks, M.O., *Reliability, Maintainability and Availability Assessment*, Rochelle Park, NJ, Spartan Books, 1973.

82. Luenberger, D.G., *Introduction to Linear and Nonlinear Programming*, New York, Addison-Wesley Co., 1973.

83. Lureau, F., "A Queuing Theoretic Analysis of Logistics Repair Models with Spare Units" (Technical Report No. 167, Department of Operations Research, Stanford University, Stanford, CA, December 16, 1974).

84. Mackliet, B.R., "The Estimation of the Inherent Availability of a Finite Population Queuing System" (Master's Thesis, VPI&SU, 1984).

85. Matta, K.F. and Daschbach, J.M., "Dual Replenishment Inventory Control", *Operations Research News* XVII (Spring 1983):1-3.

86. McRoberts, K.L., "A Search Model for Evaluating Combinatorially Explosive Problems", *Operations Research*, 19(6) (October 1971):1331-1349.

87. Merchant, M.J. and Sturgul, J.R., *Applied FORTRAN Programming with Standard FORTRAN, WATFOR, WATFIV, and Structured WATFIV*, Belmont, CA, Wadsworth Publishing Co., 1977.

88. Miele, A., Cragg, E.E., Iyer, R.R. and Levy, A.V., "Use of the Augmented Penalty Function in Mathematical Programming Problems", *Journal of Optimization Theory and Applications* 8 (1971):115-153.

89. Miele, A., Moseley, P.E., Levy, A.V. and Coggins, G.M., "On the Method of Multipliers for Mathematical Programming Problems", *Journal of Optimization Theory and Applications* 10 (1972):1-33.

90. Miele, A., Moseley, P.E., and Cragg, E.E., "A Modification of the Method of Multipliers for Mathematical Programming Problems", *Techniques of Optimization* (eds. Balakrishnan, A.V.) New York, Academic Press, 1971.

91. Mirasol, N.M., "A Queuing Approach to Logistics Systems", *Operations Research* 12 (September-October 1964):707-727.

92. Mood, A.M., Graybill, F.A. and Boes, D.C., *Introduction to the Theory of Statistics*, New York, McGraw-Hill Book Co., 1974.

93. Moore, T.P., "A Classification Hierarchy for Repairable Equipment Logistic Systems," paper presented at the Joint National Meeting of The Institute for Management Science and the Operations Research Society of America, Boston, MA, 29 April 1985.

94. Moore, T.P. and Fabrycky, W.J., "Modeling Multiple Repairable Equipment and Logistic Systems", *Proceedings*, 20th International Symposium, Society of Logistics Engineers, San Diego, CA, August, 1985.

95. Morse, P.M., *Queues, Inventories and Maintenance*, New York, John Wiley & Sons, Inc., 1958.

96. Muckstadt, J.A., "A Model for a Multi-Item, Multi-Echelon, Multi-Indenture Inventory System", *Management Science* 20 (December 1973):472-481.

97. Myers, D.C., "The Design of Branch and Bound Lagrangian Relaxation and Subgradient Optimization Strategies for Mixed Integer Programming Problems", (Doctoral Dissertation, VPI&SU, 1984).

98. Nahmias, S., 'Managing Repairable Item Inventory Systems: A Review', "Multi-level Production/Inventory Control Systems: Theory and Practice", *Studies in Management Science* 16 (1981):253-277.

99. Nelder, J.A. and Mead, R., "A Simplex Method for Function Minimization", *Computer Journal* 7(4) (January 1965):308-313.

100. Newdorf, A.S., "Logistics Trends for the New Electronic Systems," *Logistics Spectrum* 19(2) (Summer 1985):12-20.

101. Nguyen, D.G. and Murthy, D.N.P., "Optimal Preventive Maintenance Policies for Repairable Systems", *Operations Research* 6 (November-December 1981):1181-1194.

102. Palm, C., "The Distribution of Repairmen in Servicing Automatic Machines," (in Swedish), *Industritidningen Norden* 75 (1947), pp. 75-80, 90-94, 119-123. Cited in Feller [39].

103. Peck, L.G., *Finite Queuing Tables*, New York, John Wiley & Sons, 1958.

104. Porteus, E.L. and Lansdowne, Z.F., "Optimal Design of a Multi-Item, Multi-Location, Multi-Repair Type Repair and Supply System", *Naval Research Logistics Quarterly* 21 (1974):213-237.

105. Powell, M.J.D., "A Method for Nonlinear Constraints in Minimization Problems", *Optimization* (ed. Fletcher, R., 1969):283-293.

106. Powell, M.J.D., "Algorithms for Nonlinear Constraints that use Lagrangian Functions", *Mathematical Programming* 14 (1978):224-248.

107. Rao, M.V.C. and Chandra, K.S., "A New Unidimensional Search Method for Optimization", *Engineering Optimization* 6 (1982):39-50.

108. Rao, S.S., *Optimization: Theory and Applications*, New Delhi, Wiley Eastern Limited, 1979.

109. Rau, J.G., *Optimization and Probability in Systems Engineering*, New York, Van Nostrand Reinhold Company, 1970.

110. Rosenbrock, H.H., "An Automatic Method for Finding the Greatest or Least Value of a Function," *Computer Journal* 3 (1960):175-184. Cited in Bazarra and Shetty [6].

111. Schittkowski, K., *Nonlinear Programming Codes: Information, Tests, Performance*, Springer-Verlag, 1980. Cited in Balachandran [2].

112. Sherbrooke, C.C., "An Evaluator of the Number of Operationally Ready Aircraft in a Multi-Level Supply System", *Operations Research* 19 (May-June 1971):618-635.

113. Sherbrooke, C.C., "METRIC: A Multi-Echelon Technique for Recoverable Item Control", *Operations Research* 16 (January-February 1968):122-141.

114. Shields, P.C., *Elementary Linear Programming*, New York, Worth Publishers, 1968.

115. Simmons, D.M., *Nonlinear Programming for Operations Research*, Englewood Cliffs, NJ, Prentice-Hall, 1975.

116. Stein, R.G., "Futuristic Logistics", *Logistics Spectrum* 19 (Spring 1985):48-49.

117. Stewart, G.W., III, "A Modification of Davidon's Minimization Method to Accept Difference Approximations to Derivatives," *Journal of the Association of Computing Machinery* 14 (1967):72. Cited in Fletcher [41].

118. Swann, W.H., "Report on the Development of a New Direct Search Method of Optimization," *Imperial Chemical Industries Ltd. Central Instrumentation Research Laboratory Research Note* London, 64/3, (1964). Cited in Bazarra and Shetty [6].

119. Tersine, R.J., *Materials Management and Inventory Systems*, New York, North-Holland, 1976, p. 2. Cited in Coyle and Bardi [29].

120. Thompson, W.E. and Haynes, R.D., "On the Reliability, Availability and Bayes Confidence Intervals for Multicomponent Systems", *Naval Research Logistics Quarterly* 27 (September 1980):345-358.

121. Thompson, W.E. and Springer, M.D., "A Bayes Analysis of Availability for a System Consisting of Several Independent Subsystems", *IEEE Transactions on Reliability* R21(4) (1972).

122. Thuesen, G.J. and Fabrycky, W.J., *Engineering Economy*, 6th ed., Englewood Cliffs, NJ, Prentice-Hall, 1984.

123. Trefethen, F.N., "A History of Operations Research", *Operations Research for Management*, ed. J.F. McCloskey and F.N. Trefethen, Baltimore, Johns Hopkins Press, 1954.

124. US Department of Commerce, "1983 Expenditures by U.S. Non-Farm Business", *Survey of Current Business*, Washington, D.C., January, 1984.

125. US Department of Commerce, "US Passenger Transit Industry", *Statistical Abstract of the United States*, Washington, D.C., December, 1983.

126. US Department of Defense, *Real and Personal Property*, Directorate for Information, Operations, and Reports, Washington HQ Services, Washington, D.C., April, 1982, p. 19.

127. US Department of the Navy, "Operational Availability of Naval Weapons Systems and Equipment" (Deputy Chief of Naval Operations (Logistics), February, 1984).

128. Venkatesan, M., "Production-Inventory with Equipment Replacement - PIER," *Operations Research* 32(6) (November-December 1984):1286-1295.

129. Veinott, A.F., "On the Optimality of (s,S) Inventory Policies: New Conditions and a New Proof," *SIAM Journel* 14:1067-1083. Cited in Venkatesan [128].

130. Weisman, J. and Wood, C.F., "The Use of 'Optimal Search' for Engineering Design", *Recent Advances in Optimization Techniques*, (eds. Lavi, A. and Vogl, T.P.) New York, John Wiley & Sons, Inc. (1966):219-228.

131. Wismer, D.A. and Chattergy, R., *Introduction to Nonlinear Optimization: A Problem Solving Approach*, New York, Elsevier North-Holland, 1978.

132. Zangwill, W.I., "Minimizing a Function without Calculating Derivatives," *Computer Journal* 10 (1967):293-296. Cited in Bazarra and Shetty [6].

133. Zangwill, W.I., *Nonlinear Programming: A Unified Approach*, Englewood Cliffs, NJ, Prentice-Hall, 1969.

Appendix A

GLOSSARY

Decision Variables

N_i	the size of the ith repairable equipment population.
M_i	the number of repair channels operated in the ith REAL system.
R_i	retirement age of the repairable equipment in the ith REAL system.
S_{ij}	takes the value 1 only if design j is selected in the ith REAL system.

System Parameters

d_i	the demand to be met by the equipment in the ith REAL system.
n_{reals}	the number of REAL Systems in the MREAL system.
s_i	the number of designs (sources) considered in the ith REAL system.
p_{ij}	unit procurement cost, design j, in the ith REAL system.
$f_{ij}(R_i)$	unit salvage value function.
$rmax_{ij_i}$	the maximum possible retirement age for the design (j_ith) chosen in the ith REAL system.
o_{ij}	annual unit operating cost, design j, in the ith REAL system.
c_i	the cost of one unit-year of shortage in the ith REAL system.
rp_i	acquisition cost of a repair facility in the ith REAL system.
rf_i	salvage value of a repair facility in the ith REAL system.
ro_i	annual operating cost of a repair facility in the ith REAL system.
rr_i	the retirement age of repair facilities in the ith REAL system.

$\lambda_{ij}(t)$ — the failure rate profile function for design j, in the ith REAL system.

$\mu_{ij}(t)$ — the repair rate profile function for design j, in the ith REAL system.

i_{nt} — the annual interest rate.

$(A \mid P, i_{nt}, R_i)$ — capital recovery factor.

sh_i — catastrophic shortage number for the ith REAL system.

m — the number of constraints in a mathematical programming problem.

b_{op} — the maximum, per period, operating budget.

b_{pr} — the maximum, per period, procurement budget.

b_{UNA_i} — maximum allowable system unavailability in the ith REAL system.

b_{CAT_i} — maximum allowable probability that the ith REAL system will ever experience sh_i or more shortages.

System Performance Measures

C_{p_i} — the equivalent annual procurement and operating cost for all of the repairable equipment in the ith REAL system.

C_{f_i} — the equivalent annual procurement and operating cost for all of the repair facilities operated in support of the ith REAL system.

C_{s_i} — the average annual total cost of shortages in the ith REAL system.

C_{pr_i} — annual requirement for procurement capital in the ith REAL system.

C_{op_i} — annual requirement for operating capital in the ith REAL system.

$S_{UNA_{ij}}$ — inherent unavailability for design j, in the ith REAL system.

$C_{UNA_{ij}}(sh_i)$ — the catastrophic unavailability. Probability that the ith REAL system experiences more than sh_i shortages under design j.

$\hat{\lambda}_{ij}(R_i)$ — the mean failure rate for design j, in the ith REAL system, retiring equipment at age R_i.

$\hat{\mu}_{ij}(R_i)$ — the mean repair rate for design j, in the ith REAL system, retiring equipment at age R_i.

ρ — the ratio of $\dfrac{\hat{\lambda}_{ij}(R_i)}{\hat{\mu}_{ij}(R_i)}$.

Vector Spaces

E^n	the n-dimensional Euclidean vector space.
E^n_+	the nonnegative orthant of E^n.
E^n_{++}	the positive orthant of E^n.
E^1	the one dimensional real space.
E^1_+	the nonnegative one dimensional real space.
E^n_{ef}	the region of the n-dimensional Euclidean vector space defined by the continuous relaxation of the explicit constraints of the MREAL1 model.

Vectors

$\bar{e}^{(1)}, \dots, \bar{e}^{(n)}$	the principal coordinate vectors of E^n.
$\bar{x} \in E^n$	a column vector in E^n with components x_i.
$\bar{x}^T$	the row vector that is the transpose of $\bar{x}$.
$\{\bar{x}^{(1)}, \dots, \bar{x}^{(m)}\}$	a set of m vectors.
$\bar{\lambda}$	the vector of Lagrange multipliers. $\bar{\lambda} \in E^m$.
$\bar{\rho}$	the vector of penalty parameters. $\bar{\rho} \in E^m$.
$(x_1, \dots, x_m)^T$	the column vector $\bar{x} \in E^m$.
$\Vert\bar{x}\Vert_2$	the Euclidean vector norm, also known as the Frobenius vector norm, of $\bar{x} \in E^n$. $\Vert\bar{x}\Vert_2 \equiv [\sum_{i=1}^{n}(x_i)^2]^{0.5}$
$\Vert\bar{x}\Vert_\infty$	the Chebyshev vector norm, also known as the maximum vector norm, of $\bar{x} \in E^n$. $\Vert\bar{x}\Vert_\infty \equiv \max_i\{\vert x_i\vert\}$

Matrices

$A = (a_{ij})$	a matrix A consisting of the elements (a_{ij}).
$A \in E^{mn}$	the matrix A with order m by n, i.e. m rows and n columns.
A^T	the transpose of the matrix A.
A^{-1}	the inverse of the matrix A, defined when $A \in E^{nn}$ and $\vert A\vert \neq 0$.
A^{-T}	$(A^{-1})^T = (A^T)^{-1}$, defined when $A \in E^{nn}$ and $\vert A\vert \neq 0$.

I	an identity matrix of appropriate size.
M	a matrix of the gradients of the constraints which are binding at a search point.
P	the gradient projection matrix in Rosen's gradient projection method.

Functions

$f(\overline{x})$	an objective function in mathematical programming. To be minimized or maximized depending upon the specific problem. $f(\overline{x}):E^n \rightarrow E^1$.
$g_i(\overline{x})$	the ith constraint function of the nonlinear problem. $g_1(\overline{x}):E^n \rightarrow E^1$.

Acronyms

CLP	The catastrophic loss problem in a REAL system.
CSP	The channel sharing problem in an MREAL system.
DEC	Digital Equipment Corporation, Inc.
DOD	The United States Department of Defense.
EDP	The equipment design problem in a REAL system.
FIFO	The first in, first out queuing discipline.
FORTRAN	A high level computer language.
GDP1	The geographic distribution problem associated with the spare parts provisioning problem.
GDP2	The geographic distribution problem associated with the levels and channels problem and the maintenance configuration problem.
GDP3	The geographic distribution problem associated with the spare equipment problem.
GDP4	The geographic distribution problem associated with the preventive maintenance problem.
GDP5	The geographic distribution problem associated with the inspection and testing problem.
IBM	International Business Machines, Inc.
ILS	Integrated logistics support.
IMSL	International Mathematics Subroutine Library.
ITP	The inspection and testing problem in a REAL system.
LCP	The levels and channels problem in an MREAL system.
LIFO	The last in, first out queuing discipline.

LORA	Level of repair analysis.
MCP	The maintenance configuration problem in an MREAL system.
MDT	Mean maintenance downtime.
MEMD	The multiple equipment, i.e. the multiple REAL system, multiple equipment design problem. This is the most general form of the MREAL system.
MESD	The multiple equipment, single design problem. This is equivalent to having chosen the specific design combination for an MREAL system.
METRIC	The US Air Force's multi-echelon technique for recoverable item control.
MIMS	The classical, multiple item, multiple source inventory theory problem.
MISS	The classical, multiple item, single source inventory theory problem.
MLDT	Mean logistics downtime.
MREAL	Multiple repairable equipment and logistic (system).
MSDT	Mean system downtime. This refers to a REAL system.
MTA	Metropolitan Transit Authority.
MTBF	Mean time between failures. This refers to a unit of repairable equipment.
MTBSF	Mean time between system failures. This refers to a REAL system.
MTP	The mechanic training problem in a REAL system.
MTTR	Mean time to repair. This usually refers to corrective (unscheduled) maintenance.
NCPDM	National Council of Physical Distribution Management, Inc.
OLP	The ORLA/LORA problem in a REAL system.
OR	Operations research.
ORLA	Optimal repair level analysis.
OTP	The operator training problem in a REAL system.
PM	Preventive maintenance.
PMP	The preventive maintenance problem in a REAL system.
RAM	Reliability, availability, and maintainability.
RAP	The retirement age problem in a REAL system.
REAL	Repairable equipment and logistic system.
RFP	The repair facility design problem in a REAL system.
RPP	The replacement policy problem in a REAL system.
R&D	Research and development.
SEMD	The single equipment, multiple design problem. This problem is equivalent to choosing an equipment design for a REAL system, then choosing values for all of the other decision variables.

SEP	The spare equipment problem in a REAL system.
SESD	The single equipment, single design problem.
SIMS	The classical, single item, multiple source inventory theory problem.
SISS	The classical, single item, single source inventory theory problem.
SOLE	The Society of Logistics Engineers.
SPP	The spare parts provisioning problem in a REAL system.
SPT	The shortest processing time queuing discipline.

Appendix B

MREAL1 ENUMERATOR

```
************************************************************************
*                                                                      *
*                   MREAL1 TEST PROBLEM ENUMERATOR                     *
*                                                                      *
*               COPYRIGHT 1986 BY THOMAS PRESTON MOORE                 *
*                                                                      *
*               Written in FORTRAN 77 (IBM VS-FORTRAN)                 *
*                                                                      *
************************************************************************

*        ******************************************
*        *                                        *
*        *            Begin MAIN Program          *
*        *                                        *
*        ******************************************


*           *** INTEGER VARIABLE DICTIONARY ***

*  D(2)        DEMAND IN EACH REAL SYSTEM, I.
*  DESIGN(2)   DESIGN NUMBER FOR CURRENT DESIGN IN REAL SYSTEM I.
*  DESGN1(2)   DESIGN NUMBER FOR CURRENT BEST SOLUTION IN REAL SYSTEM I.
*  DESGN2(2)   DESIGN NO. FOR CURRENT BEST FEASIBLE SOLUTION IN SYSTEM I.
*  FLAG1       RESULT OF CONSTRAINT CHECK: 0 = FEASIBLE, 1+ = INFEASIBLE.
*  NREALS      NUMBER OF REAL SYSTEMS UNDER CONSIDERATION.
*  RMAX(2,2)   MAXIMUM RETIREMENT AGE FOR DESIGN J, REAL SYSTEM I.
*  S(2)        NUMBER OF DESIGNS UNDER CONSIDERATION IN REAL SYSTEM I.
*  SH(2)       NUMBER OF SHORTAGES CONSIDERED CATASTROPHIC IN REALS I.
*  XB(6)       COORDINATES OF POINT CURRENTLY BEING ENUMERATED.

*           **** REAL VARIABLE DICTIONARY ****

*  ACUNA1(2)   CATASTROPHIC UNAVAILABILITY SLACK IN EACH REAL SYSTEM, I,
*              FOR CURRENT BEST SOLUTION.
*  ACUNA2(2)   CATASTROPHIC UNAVAILABILITY SLACK IN EACH REAL SYSTEM, I,
*              FOR THE CURRENT BEST FEASIBLE SOLUTION.
*  ASUNA1(2)   SLACK IN EXPECTED UNAVAILABILITY CONSTRAINT IN EACH REAL
*              SYSTEM, I, FOR THE CURRENT BEST SOLUTION.
*  ASUNA2(2)   SLACK IN EXPECTED UNAVAILABILITY CONSTRAINT IN EACH REAL
*              SYSTEM, I, FOR THE CURRENT BEST FEASIBLE SOLUTION.
*  AOPC1       SLACK IN OPERATING COST CONSTRAINT, CURRENT BEST SOLUTION.
*  AOPC2       SLACK IN OPERATING COST CONSTRAINT FOR THE CURRENT BEST
*              FEASIBLE SOLUTION.
*  APRC1       SLACK IN ANNUAL PROCUREMENT CAPITAL CONSTRAINT FOR THE
*              CURRENT BEST SOLUTION.
*  APRC2       SLACK IN ANNUAL PROCUREMENT CAPITAL CONSTRAINT FOR THE
*              CURRENT BEST FEASIBLE SOLUTION.
*  BCAT(2)     MAXIMUM PROBABILITY OF A CATASTROPHIC SHORTAGE OCCURRING.
*  BOP         MAXIMUM ANNUAL OPERATING CAPITAL BUDGET.
*  BPR         MAXIMUM ANNUAL REPLACEMENT CAPITAL BUDGET.
*  BUNA(2)     MAXIMUM EXPECTED SHORTAGES AS A PERCENT OF DEMAND.
*  C(2)        COST OF ONE SHORTAGE-YEAR IN REAL SYSTEM I.
*  INT         ANNUAL INTEREST RATE (10% ASSUMED).
*  NORM1(2)    FINITE SOURCE QUEUEING MODEL NORMALIZATION FACTOR, .
*  O(2,2)      ANNUAL OPERATING COST OF ONE REPAIRABLE EQUIPMENT FOR
```

```
*              DESIGN J, REAL SYSTEM I.
*  P(2,2)      PROCUREMENT COST OF ONE REPAIRABLE EQUIPMENT FOR DESIGN
*              J, REAL SYSTEM I.
*  RF(2)       SALVAGE VALUE OF A REPAIR FACILITY IN REAL SYSTEM I
*              AFTER RR(I) YEARS.
*  RHO(2,2,40) MEAN RHO VALUE FOR SYSTEM IF DESIGN J IS USED IN REAL
*              SYSTEM I, WITH RETIREMENT OCCURRING IN YEAR K.
*  RO(2)       ANNUAL OPERATING COST OF REPAIR FACILITY IN REAL SYSTEM I.
*  RP(2)       PROCUREMENT COST OF A REPAIR FACILITY IN REAL SYSTEM I.
*  RR(2)       REPAIR FACILITY RETIREMENT AGE IN REAL SYSTEM I.

*            ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), X1(6), X2(6),    A(6), B(6), DESGN1(2), DESGN2(2)

*            ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), INT, NORM1(2),
     $ O(2,2), P(2,2), RF(2), RHO(2,2,40), RO(2), RP(2),
     $ RR(2), ZB, Z1, Z2,   ACUNA1(2), ACUNA2(2), AOPC1, AOPC2, APRC1,
     $ APRC2, ASUNA1(2), ASUNA2(2)

*            ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1,
     $ INT, NORM1, NREALS, O, P, RF, RHO, RMAX, RO, RP, RR,
     $ S, SH, XB, X1, X2, ZB, Z1, Z2
      COMMON /BEST/ ACUNA1, ACUNA2, AOPC1, AOPC2, APRC1, APRC2,
     $              ASUNA1, ASUNA2, DESGN1, DESGN2

*            *** WRITE HEADER FOR OUTPUT TABLE ***

      WRITE(6,900)
     $'**************************************************************'
      WRITE(6,900)
     $'*                                                            *'
      WRITE(6,900)
     $'*                      MREAL1 ENUMERATOR                     *'
      WRITE(6,900)
     $'*                                                            *'
      WRITE(6,900)
     $'*              COPYRIGHT 1986 - THOMAS PRESTON MOORE          *'
      WRITE(6,900)
     $'*                                                            *'
      WRITE(6,900)
     $'*              WRITTEN IN FORTRAN 77 (IBM VS-FORTRAN)         *'
      WRITE(6,900)
     $'*                                                            *'
      WRITE(6,900)
     $'*                   LINE PRINTER OUTPUT FORMAT               *'
      WRITE(6,900)
     $'*                                                            *'
      WRITE(6,900)
     $'**************************************************************'
      WRITE(6,900) ' '

*         *** SET HIGH VALUES FOR Z1 AND Z2 ***

      Z1 = 7.1D+74
      Z2 = 7.1D+74

*            *** CHOOSE BETWEEN ADJACENT POINT ENUMERATION ***
*            *** (OPTION 0) AND SINGLE POINT COMPUTATION   ***
*            *** (OPTION 1)                                ***

      IFLAG = 0
```

```
      IF (IFLAG.EQ.0) GO TO 500

      DESIGN(1) = 2
      DESIGN(2) = 1
      XB(1) = 12
      XB(2) =  1
      XB(3) = 15
      XB(4) = 16
      XB(5) =  1
      XB(6) = 25

*        *** WRITE HEADER FOR TABLE ***

      WRITE (6,900) ' |<******* DECISION VARIABLES ******>|  |<*********
     $**** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES **
     $*********>|'
      WRITE (6,900) ' N1 M1 R1 N2 M2 R2 DESIGN(1) DESIGN(2)  TOTAL COST
     $FLAG1   APRC        AOPC          ASUNA(1)      ASUNA(2)      ACUNA(1)
     $    ACUNA(2)'
 900  FORMAT(1X,A)
      CALL COMP
      GO TO 600

*           *** SET POINT TO BE TESTED AS A HOLE ***

 500  DESIGN(1) = 2
      DESIGN(2) = 2
      XB(1) = 12
      XB(2) =  2
      XB(3) = 20
      XB(4) = 16
      XB(5) =  1
      XB(6) = 30

*           *** WRITE THE POINT BEING TESTED ***

      WRITE(6,910) XB
 910  FORMAT(1X,'FROM ENUM: INITIAL BASE POINT:       ',6I4,/,1X,'ALL ADJ
     $ACENT POINTS TO THIS BASE POINT ARE ENUMERATED BELOW:'//)

*           *** WRITE HEADER FOR TABLE ***

      WRITE (6,900) ' |<******* DECISION VARIABLES ******>|  |<*********
     $**** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES **
     $*********>|'
      WRITE (6,900) ' N1 M1 R1 N2 M2 R2 DESIGN(1) DESIGN(2)  TOTAL COST
     $FLAG1   APRC        AOPC          ASUNA(1)      ASUNA(2)      ACUNA(1)
     $    ACUNA(2)'

*           *** COMPUTE THE LIMITS OF THE DO LOOPS ***

      IF (XB(1).EQ.1) THEN
         A(1) = 1
         B(1) = 2
      ENDIF
      IF (XB(1).EQ.56) THEN
         A(1) = 55
         B(1) = 56
      ENDIF
      IF (XB(1).GT.1.AND.XB(1).LT.56) THEN
         A(1) = XB(1) - 1
         B(1) = XB(1) + 1
      ENDIF
      IF (XB(2).EQ.1) THEN
         A(2) = 1
         B(2) = 2
      ENDIF
```

```
      IF (XB(2).EQ.56) THEN
         A(2) = 55
         B(2) = 56
      ENDIF
      IF (XB(2).GT.1.AND.XB(2).LT.56) THEN
         A(2) = XB(2) - 1
         B(2) = XB(2) + 1
      ENDIF
      IF (XB(3).EQ.1) THEN
         A(3) = 1
         B(3) = 2
      ENDIF
      IF (XB(3).EQ.RMAX(1,DESIGN(1))) THEN
         A(3) = XB(3) - 1
         B(3) = XB(3)
      ENDIF
      IF (XB(3).GT.1.AND.XB(3).LT.RMAX(1,DESIGN(1))) THEN
         A(3) = XB(3) - 1
         B(3) = XB(3) + 1
      ENDIF
      IF (XB(4).EQ.1) THEN
         A(4) = 1
         B(4) = 2
      ENDIF
      IF (XB(4).EQ.56) THEN
         A(4) = 55
         B(4) = 56
      ENDIF
      IF (XB(4).GT.1.AND.XB(4).LT.56) THEN
         A(4) = XB(4) - 1
         B(4) = XB(4) + 1
      ENDIF
      IF (XB(5).EQ.1) THEN
         A(5) = 1
         B(5) = 2
      ENDIF
      IF (XB(5).EQ.56) THEN
         A(5) = 55
         B(5) = 56
      ENDIF
      IF (XB(5).GT.1.AND.XB(5).LT.56) THEN
         A(5) = XB(5) - 1
         B(5) = XB(5) + 1
      ENDIF
      IF (XB(6).EQ.1) THEN
         A(6) = 1
         B(6) = 2
      ENDIF
      IF (XB(6).EQ.RMAX(2,DESIGN(2))) THEN
         A(6) = XB(6) - 1
         B(6) = XB(6)
      ENDIF
      IF (XB(6).GT.1.AND.XB(6).LT.RMAX(2,DESIGN(2))) THEN
         A(6) = XB(6) - 1
         B(6) = XB(6) + 1
      ENDIF

*           ******* SET DESIGN COMBINATION *******

      DO 1  L1 = DESIGN(1), DESIGN(1)

         DO 2  L2 = DESIGN(2), DESIGN(2)
*
*
            DESIGN(1) = L1
            DESIGN(2) = L2
            DO 3  J1 = A(1), B(1)
```

```
            XB(1) = J1
         DO 4  J4 = A(4), B(4)
            XB(4) = J4
         DO 5  J2 = A(2), B(2)
            XB(2) = J2
         DO 6  J5 = A(5), B(5)
            XB(5) = J5
         IF (L1.EQ.1) THEN
            DO 7  J3 = A(3), B(3)
               XB(3) = J3
            IF (L2.EQ.1) THEN
               DO 8  J6 = A(6), B(6)
                  XB(6) = J6
                  CALL COMP
 8             CONTINUE
            ENDIF
            IF (L2.EQ.2) THEN
               DO 9  J6 = A(6), B(6)
                  XB(6) = J6
                  CALL COMP
 9             CONTINUE
            ENDIF
 7          CONTINUE
         ENDIF

         IF (L1.EQ.2) THEN
            DO 10 J3 = A(3), B(3)
               XB(3) = J3
            IF (L2.EQ.1) THEN
               DO 11 J6 = A(6), B(6)
                  XB(6) = J6
                  CALL COMP
 11            CONTINUE
            ENDIF
            IF (L2.EQ.2) THEN
               DO 12 J6 = A(6), B(6)
                  XB(6) = J6
                  CALL COMP
 12            CONTINUE
            ENDIF
 10         CONTINUE
         ENDIF

 6       CONTINUE
 5       CONTINUE
 4       CONTINUE
 3       CONTINUE
 2    CONTINUE
 1  CONTINUE

*   DO 1  I = 1, 6
*      XB(I) = XB(I) - 1
*      CALL COMP
*      XB(I) = XB(I) + 2
*      CALL COMP
*      XB(I) = XB(I) - 1
*1  CONTINUE

*      *** WRITE BEST POINT FOUND ***

 600 WRITE (6,900) '-------------------------------------------------'
     WRITE (6,900) 'N1  M1  R1  N2  M2  R2 DESIGN(1) DESIGN(2) BEST COS
    $T    APRC         AOPC        ASUNA(1)      ASUNA(2)      ACUNA(1)
    $ACUNA(2)'
     WRITE (6,902) X1, DESGN1, Z1, APRC1, AOPC1, ASUNA1, ACUNA1
 902 FORMAT (1X,6(I2,2X),1X,I2,8X,I2,3X,F12.0,2(F10.0,1X),4(E12.
    $6,1X)/)
```

```
*        *** WRITE BEST FEASIBLE POINT FOUND ***

      WRITE (6,900) '---------------------------------------------------'
      WRITE (6,900) 'N1  M1  R1  N2  M2  R2 DESIGN(1) DESIGN(2) BEST FEA
     $S. COST APRC     AOPC       ASUNA(1)      ASUNA(2)     ACUNA(1)
     $ACUNA(2)'
      WRITE (6,902) X2, DESGN2, Z2, APRC2, AOPC2, ASUNA2, ACUNA2

      END

*        ******************************************
*        *                                        *
*        *     TEMPORARY BLOCK DATA INPUT AREA    *
*        *                                        *
*        ******************************************


      BLOCK DATA TOM

      INTEGER D(2), DESIGN(2), FLAG1, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), X1(6), X2(6)
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), INT, NORM1(2),
     $ O(2,2), P(2,2), RF(2), RHO(2,2,40), RO(2), RP(2),
     $ RR(2), ZB, Z1, Z2
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1,
     $ INT, NORM1, NREALS, O, P, RF, RHO, RMAX, RO, RP, RR,
     $ S, SH, XB, X1, X2, ZB, Z1, Z2
      DATA NREALS, D, S /2, 10, 15, 2, 2/
      DATA RMAX, SH      /15, 25, 20, 30, 3, 5/
      DATA INT, P        /0.10, 1.4E5, 3.5E5, 1.6E5, 4.2E5/
      DATA O, C          /1.0E5, 0.7E5, 0.9E5, 0.65E5, 18.25E5, 54.75E5/
      DATA RP, RF, RO    /3.0E5, 5.0E5, 0.8E5, 1.2E5, 0.9E5, 1.3E5/
      DATA RR, BPR, BOP  /15., 15., 0.6E6, 3.0E6/
      DATA BUNA, BCAT    /5.0E7, 1.0E7, 1.0E6, 1.0E5/
      DATA (((RHO(I,J,K),I=1,2),J=1,2),K=1,8)/.0250, .010, .0240, .0060,
     $          0.0175,  0.0043,  0.0177,  0.0033,
     $     0.014545455,  0.0027,  0.0153,  0.0021,
     $     0.013333333,  0.0020,  0.0141,  0.0016,
     $     0.012954545,  0.0016,  0.0133,  0.0013,
     $     0.013333333,  0.0014,  0.0129,  0.0012,
     $     0.013809524,  0.0013,  0.0129,  0.0011,
     $     0.014366197,  0.0012,  0.0129,  0.0010/
      DATA (((RHO(I,J,K),I=1,2),J=1,2),K=9,16)/.015, .0011, .0132,.0009,
     $     0.015529412,  0.0012,  0.0134,  0.0009,
     $     0.016195652,  0.0011,  0.0136,  0.0009,
     $     0.017319588,  0.0012,  0.0140,  0.0009,
     $     0.018800000,  0.0012,  0.0143,  0.0008,
     $     0.020291262,  0.0012,  0.0151,  0.0008,
     $     0.022000000,  0.0013,  0.0160,  0.0008,
     $          0.0,     0.0014,  0.0171,  0.0009/
      DATA (((RHO(I,J,K),I=1,2),J=1,2),K=17,24)/.0, .0014, .0183, .0009,
     $          0.0,     0.0016,  0.0201,  0.0009,
     $          0.0,     0.0017,  0.0218,  0.0010,
     $          0.0,     0.0018,  0.0238,  0.0010,
     $          0.0,     0.0021,  0.0,     0.0011,
     $          0.0,     0.0023,  0.0,     0.0011,
     $          0.0,     0.0025,  0.0,     0.0011,
     $          0.0,     0.0028,  0.0,     0.0012/
      DATA (((RHO(I,J,K),I=1,2),J=1,2),K=25,30)/0.0, .0030, 0.0, .0012,
     $          0.0,     0.0,     0.0,     0.0014,
     $          0.0,     0.0,     0.0,     0.0015,
     $          0.0,     0.0,     0.0,     0.0016,
     $          0.0,     0.0,     0.0,     0.0017,
     $          0.0,     0.0,     0.0,     0.0018/
      END
```

```
      BLOCK DATA APE
*     **** CAPITAL RECOVERY FACTOR, AP, AT 10%, YEARS 1 - 40 ****

      REAL APRC, AOPC, AP(40), ACUNA(2), ASUNA(2)
      COMMON /OPT1/ APRC, AOPC, ACUNA, ASUNA, AP
      DATA AP / 1.1, 0.576190, 0.402115, 0.315471, 0.263797, 0.229607,
     $     0.205405, 0.187444, 0.173641, 0.162745, 0.153963, 0.146763,
     $     0.140779, 0.135746, 0.131474, 0.127817, 0.124664, 0.121930,
     $     0.119547, 0.117460, 0.115624, 0.114005, 0.112572, 0.111300,
     $     0.110168, 0.109159, 0.108258, 0.107451, 0.106728, 0.106079,
     $     0.105496, 0.104972, 0.104499, 0.104074, 0.103690, 0.103343,
     $     0.103030, 0.102747, 0.102491, 0.102259/
      END

      SUBROUTINE COMP

*            ****************************************
*            *                                      *
*            *  ENUMERATION COMPUTATION SUBROUTINE  *
*            *                                      *
*            ****************************************


*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), X1(6), X2(6),   XF(6), DESGN1(2), DESGN2(2)

*           ********* REAL DECLARATIONS *********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), INT, NORM1(2),
     $ O(2,2), P(2,2), RF(2), RHO(2,2,40), RO(2), RP(2),
     $ RR(2), ZB, Z1, Z2,   ACUNA(2), AP(40), AOPC, APRC, ASUNA(2),
     $ ACUNA1(2), ACUNA2(2), AOPC1, AOPC2, APRC1, APRC2, ASUNA1(2),
     $ ASUNA2(2)

*           ******** LABELED COMMON BLOCKS ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1,
     $ INT, NORM1, NREALS, O, P, RF, RHO, RMAX, RO, RP, RR,
     $ S, SH, XB, X1, X2, ZB, Z1, Z2
      COMMON /OPT1/ APRC, AOPC, ACUNA, ASUNA, AP
      COMMON /BEST/ ACUNA1, ACUNA2, AOPC1, AOPC2, APRC1, APRC2,
     $              ASUNA1, ASUNA2, DESGN1, DESGN2

      FLAG1 = 0


      CALL NORMAL (XB)

*           *** ANNUAL REPLACEMENT PROCUREMENT ***
*           ********* CAPITAL CONSTRAINT *********

      APRC = BPR
      DO 1  I = 1, NREALS

            APRC = APRC - CPR(I,XB)

 1    CONTINUE

      IF (APRC.LT.0.0) FLAG1 = 1

*           ****** ANNUAL OPERATING CAPITAL ******
*           ************ CONSTRAINT *************

      AOPC = BOP
      DO 2  I = 1, NREALS
```

```
            AOPC = AOPC - COP(I,XB)

  2     CONTINUE

        IF (AOPC.LT.0.0) FLAG1 = 2

*           *** CATASTROPHIC UNAVAILABILITY *******
*           ********* CONSTRAINTS *************

        DO 3 I = 1, NREALS

           ACUNA(I) = -CUNA(I,XB) + BCAT(I)
           IF (ACUNA(I).LT.0.0) FLAG1 = I + 4

  3     CONTINUE


*           ******* SYSTEM UNAVAILABILITY *******
*           ********* CONSTRAINTS *************

        DO 4 I = 1, NREALS

           ASUNA(I) = -SUNA(I,XB) + BUNA(I)
           IF (ASUNA(I).LT.0.0) FLAG1 = I + 2

  4     CONTINUE

*         *** COMPUTE THE TOTAL SYSTEM COST FOR SOLUTION XB ***

        CALL OBJ(XB,ZB)

*         *** RECORD BEST POINT FOUND SO FAR ***

        IF (ZB.LT.Z1) THEN
           Z1 = ZB
           DO 15 K = 1, NREALS
              DESGN1(K) = DESIGN(K)
              ACUNA1(K) = ACUNA(K)
              ASUNA1(K) = ASUNA(K)
 15        CONTINUE
           APRC1 = APRC
           AOPC1 = AOPC
           DO 20  K = 1, 3*NREALS
              X1(K) = XB(K)
 20        CONTINUE
        ENDIF

*         *** RECORD BEST FEASIBLE POINT FOUND SO FAR ***

        IF (ZB.LT.Z2.AND.FLAG1.EQ.0) THEN
           Z2 = ZB
           DO 16 K = 1, NREALS
              DESGN2(K) = DESIGN(K)
              ACUNA2(K) = ACUNA(K)
              ASUNA2(K) = ASUNA(K)
 16        CONTINUE
           APRC2 = APRC
           AOPC2 = AOPC
           DO 21  K = 1, 3*NREALS
              X2(K) = XB(K)
 21        CONTINUE
        ENDIF

*         *** WRITE THE RESULTS ON ONE LINE USING THE TITLE FORMAT ***

        WRITE (6,901) XB(1), XB(2), XB(3), XB(4), XB(5), XB(6), DESIGN(1),
```

```
     $DESIGN(2), ZB, FLAG1, APRC, AOPC, ASUNA, ACUNA

901   FORMAT (1X,6I3,4X,I2,8X,I2,4X,E12.7,2X,I2,1X,2(F10.0,1X),2X,4(E12.
     $6,1X))
      RETURN
      END


      FUNCTION CPR(I,XF)

      INTEGER D(2), DESIGN(2), FLAG1, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), X1(6), X2(6),    XF(6)
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), INT, NORM1(2),
     $ O(2,2), P(2,2), RF(2), RHO(2,2,40), RO(2), RP(2),
     $ RR(2), ZB, Z1, Z2
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1,
     $ INT, NORM1, NREALS, O, P, RF, RHO, RMAX, RO, RP, RR,
     $ S, SH, XB, X1, X2, ZB, Z1, Z2

          CPR = XF(3*I-2) * P(I,DESIGN(I))/XF(3*I)

      RETURN
      END


      FUNCTION COP(I,XF)

      INTEGER D(2), DESIGN(2), FLAG1, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), X1(6), X2(6),    XF(6)
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), INT, NORM1(2),
     $ O(2,2), P(2,2), RF(2), RHO(2,2,40), RO(2), RP(2),
     $ RR(2), ZB, Z1, Z2
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1,
     $ INT, NORM1, NREALS, O, P, RF, RHO, RMAX, RO, RP, RR,
     $ S, SH, XB, X1, X2, ZB, Z1, Z2

          COP = XF(3*I-2) * O(I, DESIGN(I)) + XF(3*I-1) * RO(I)

      RETURN
      END


      FUNCTION CUNA(I,XF)

      INTEGER D(2), DESIGN(2), FLAG1, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), X1(6), X2(6),    XF(6), RTEMP
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), INT, NORM1(2),
     $ O(2,2), P(2,2), RF(2), RHO(2,2,40), RO(2), RP(2),
     $ RR(2), ZB, Z1, Z2
      DOUBLE PRECISION DCUNA, MDBLE, MTEMP, NFACT, NTEMP, TEMP1
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1,
     $ INT, NORM1, NREALS, O, P, RF, RHO, RMAX, RO, RP, RR,
     $ S, SH, XB, X1, X2, ZB, Z1, Z2

      N = XF(3*I-2)
      M = XF(3*I-1)
      MDBLE = DBLE(XF(3*I-1))
      RTEMP = XF(3*I)
      NTEMP = NFACT(N)
      MTEMP = NFACT(M)
      DCUNA = 0.0

      DO 1 K = SH(I), D(I)

         IN = N-D(I)+K
         IF (IN.LE.M) TEMP1 = (NTEMP*RHO(I,DESIGN(I),RTEMP)**IN/
     $                         NFACT(N-IN))/NFACT(IN)
```

```
         IF (IN.GT.M) TEMP1 = ((NTEMP*RHO(I,DESIGN(I),RTEMP)**IN/
     $                          NFACT(N-IN))/MTEMP)/(MDBLE**(IN-M))
         DCUNA = DCUNA + TEMP1

 1    CONTINUE

      CUNA = 1.0E10*SNGL(DCUNA)/NORM1(I)

      RETURN
      END


      FUNCTION SUNA(I,XF)

      INTEGER D(2), DESIGN(2), FLAG1, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), X1(6), X2(6),   XF(6), RTEMP
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), INT, NORM1(2),
     $ O(2,2), P(2,2), RF(2), RHO(2,2,40), RO(2), RP(2),
     $ RR(2), ZB, Z1, Z2
      DOUBLE PRECISION DSUNA, MDBLE, MTEMP, NFACT, NTEMP, TEMP1
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1,
     $ INT, NORM1, NREALS, O, P, RF, RHO, RMAX, RO, RP, RR,
     $ S, SH, XB, X1, X2, ZB, Z1, Z2

      N = XF(3*I-2)
      M = XF(3*I-1)
      MDBLE = DBLE(XF(3*I-1))
      RTEMP = XF(3*I)
      NTEMP = NFACT(N)
      MTEMP = NFACT(M)
      DSUNA = 0.0

      DO 1 K = 1, D(I)

         IN = N-D(I)+K
         IF (IN.LE.M) TEMP1 = (NTEMP*(K*RHO(I,DESIGN(I),RTEMP)**IN)/
     $                        NFACT(N-IN))/NFACT(IN)
         IF (IN.GT.M) TEMP1 = ((NTEMP*(K*RHO(I,DESIGN(I),RTEMP)**IN)/
     $                          NFACT(N-IN))/MTEMP)/(MDBLE**(IN-M))
         DSUNA = DSUNA + TEMP1

 1    CONTINUE

      SUNA = 1.0E9*(SNGL(DSUNA)/NORM1(I))/D(I)

      RETURN
      END

      FUNCTION SUNA1(I,XF)

      INTEGER D(2), DESIGN(2), FLAG1, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), X1(6), X2(6),   XF(6), RTEMP
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), INT, NORM1(2),
     $ O(2,2), P(2,2), RF(2), RHO(2,2,40), RO(2), RP(2),
     $ RR(2), ZB, Z1, Z2,   SUNA, SUNA1
      DOUBLE PRECISION DSUNA, MDBLE, MTEMP, NFACT, NTEMP, TEMP1
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1,
     $ INT, NORM1, NREALS, O, P, RF, RHO, RMAX, RO, RP, RR,
     $ S, SH, XB, X1, X2, ZB, Z1, Z2

      N = XF(3*I-2)
      M = XF(3*I-1)
      MDBLE = DBLE(XF(3*I-1))
      RTEMP = XF(3*I)
      NTEMP = NFACT(N)
      MTEMP = NFACT(M)
      DSUNA = 0.0
```

```
      DO 1 K = 1, D(I)

         IN = N-D(I)+K
         IF (IN.LE.M) TEMP1 = (NTEMP*(K*RHO(I,DESIGN(I),RTEMP)**IN)/
     $                        NFACT(N-IN))/NFACT(IN)
         IF (IN.GT.M) TEMP1 = ((NTEMP*(K*RHO(I,DESIGN(I),RTEMP)**IN)/
     $                         NFACT(N-IN))/MTEMP)/(MDBLE**(IN-M))
         DSUNA = DSUNA + TEMP1

 1    CONTINUE

      SUNA1 = SNGL(DSUNA)/NORM1(I)

      RETURN
      END

      SUBROUTINE OBJ(XT,ZT)

*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), X1(6), X2(6),   XT(6)

*           ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), INT, NORM1(2),
     $ O(2,2), P(2,2), RF(2), RHO(2,2,40), RO(2), RP(2),
     $ RR(2), ZB, Z1, Z2,   ACUNA(2), AP(40), APRC, AOPC, ASUNA(2),
     $ BMU, RT, ZT

*           ******** LABELED COMMON BLOCKS ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1,
     $ INT, NORM1, NREALS, O, P, RF, RHO, RMAX, RO, RP, RR,
     $ S, SH, XB, X1, X2, ZB, Z1, Z2

      COMMON /OPT1/ APRC, AOPC, ACUNA, ASUNA, AP

      ZT = 0.0

      DO 3 I = 1, NREALS

         RT = REAL(XT(3*I))
         FT = P(I,DESIGN(I))*(1.0-RT/RMAX(I,DESIGN(I)))
         CP = XT(3*I-2)*(AP(RT)*(P(I,DESIGN(I))-FT)+INT*FT+O(I,DESIGN(I)
     $ ))
         CF = XT(3*I-1)*(AP(RR(I))*(RP(I)-RF(I))+INT*RF(I)+RO(I))
         CS = C(I)*SUNA1(I,XT)

         ZT = ZT + CP + CF + CS


 3    CONTINUE
      RETURN
      END

      SUBROUTINE NORMAL (XT)

*              ***************************************
*              *                                     *
*              *     COMPUTES FINITE SOURCE QUEUE    *
*              *         NORMALIZATION FACTOR        *
*              *                                     *
*              ***************************************

*           ********* INTEGER DECLARATIONS *********
```

```
      INTEGER D(2), DESIGN(2), FLAG1, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), X1(6), X2(6), XT(6)

*           ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), INT, NORM1(2),
     $ O(2,2), P(2,2), RF(2), RHO(2,2,40), RO(2), RP(2),
     $ RR(2), ZB, Z1, Z2

*           ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1,
     $ INT, NORM1, NREALS, O, P, RF, RHO, RMAX, RO, RP, RR,
     $ S, SH, XB, X1, X2, ZB, Z1, Z2

*           ***** DOUBLE PRECISION DECLARATION *****

      DOUBLE PRECISION DNORM1(2), MDBLE, MTEMP, NFACT, NTEMP, TEMP1

      DO 1 I = 1, NREALS

         DNORM1(I) = 0.0
         NTEMP = NFACT(XT(3*I-2))
         MTEMP = NFACT(XT(3*I-1))
         MDBLE = DBLE(XT(3*I-1))

         DO 2 K = 0, XT(3*I-1)

            TEMP1 = (((RHO(I,DESIGN(I),XT(3*I))**K)*NTEMP)/NFACT(XT(3*I-
     $              2)-K))/NFACT(K)
            DNORM1(I) = DNORM1(I) + TEMP1

 2       CONTINUE

         DO 3 K = XT(3*I-1)+1, XT(3*I-2)

            TEMP1 = ((((RHO(I,DESIGN(I),XT(3*I))**K)*NTEMP)/NFACT(XT(3*I
     $              -2)-K))/MTEMP)/MDBLE**(K-XT(3*I-1))
            DNORM1(I) = DNORM1(I) + TEMP1

 3       CONTINUE

         NORM1(I) = SNGL(DNORM1(I))

 1    CONTINUE

      RETURN
      END

      DOUBLE PRECISION FUNCTION NFACT(N)
      DOUBLE PRECISION NRAY(56)
      DATA NRAY(1)   / 1.0D+00/
      DATA NRAY(2)   / 2.0D+00/
      DATA NRAY(3)   / 6.0D+00/
      DATA NRAY(4)   / 2.4D+01/
      DATA NRAY(5)   / 1.2D+02/
      DATA NRAY(6)   / 7.2D+02/
      DATA NRAY(7)   / 5.04D+03/
      DATA NRAY(8)   / 4.032D+04/
      DATA NRAY(9)   / 3.6288D+05/
      DATA NRAY(10)  / 3.6288D+06/
      DATA NRAY(11)  / 3.99168D+07/
      DATA NRAY(12)  / 4.790016D+08/
      DATA NRAY(13)  / 6.2270208D+09/
      DATA NRAY(14)  / 8.71782912D+10/
      DATA NRAY(15)  / 1.307674368D+12/
```

```
      DATA NRAY(16)  / 2.0922789888D+13/
      DATA NRAY(17)  / 3.55687428096D+14/
      DATA NRAY(18)  / 6.402373705728D+15/
      DATA NRAY(19)  / 1.2164510040883D+17/
      DATA NRAY(20)  / 2.4329020081766D+18/
      DATA NRAY(21)  / 5.1090942171709D+19/
      DATA NRAY(22)  / 1.1240007277776D+21/
      DATA NRAY(23)  / 2.5852016738885D+22/
      DATA NRAY(24)  / 6.2044840173324D+23/
      DATA NRAY(25)  / 1.5511210043331D+25/
      DATA NRAY(26)  / 4.0329146112661D+26/
      DATA NRAY(27)  / 1.0888869450418D+28/
      DATA NRAY(28)  / 3.0488834461171D+29/
      DATA NRAY(29)  / 8.8417619937397D+30/
      DATA NRAY(30)  / 2.6525285981219D+32/
      DATA NRAY(31)  / 8.2228386541779D+33/
      DATA NRAY(32)  / 2.6313083693369D+35/
      DATA NRAY(33)  / 8.6833176188119D+36/
      DATA NRAY(34)  / 2.9523279903960D+38/
      DATA NRAY(35)  / 1.0333147966386D+40/
      DATA NRAY(36)  / 3.7199332678990D+41/
      DATA NRAY(37)  / 1.3763753091226D+43/
      DATA NRAY(38)  / 5.2302261746660D+44/
      DATA NRAY(39)  / 2.0397882081197D+46/
      DATA NRAY(40)  / 8.1591528324790D+47/
      DATA NRAY(41)  / 3.3452526613164D+49/
      DATA NRAY(42)  / 1.4050061177529D+51/
      DATA NRAY(43)  / 6.0415263063374D+52/
      DATA NRAY(44)  / 2.6582715747884D+54/
      DATA NRAY(45)  / 1.1962222086548D+56/
      DATA NRAY(46)  / 5.5026221598121D+57/
      DATA NRAY(47)  / 2.5862324151117D+59/
      DATA NRAY(48)  / 1.2413915592536D+61/
      DATA NRAY(49)  / 6.0828186403427D+62/
      DATA NRAY(50)  / 3.0414093201713D+64/
      DATA NRAY(51)  / 1.5511187532874D+66/
      DATA NRAY(52)  / 8.0658175170944D+67/
      DATA NRAY(53)  / 4.2748832840600D+69/
      DATA NRAY(54)  / 2.3084369733924D+71/
      DATA NRAY(55)  / 1.2696403353658D+73/
      DATA NRAY(56)  / 7.1099858780486D+74/
*
      IF (N.GT.56.OR.N.LT.0) GOTO 199
      IF (N.EQ.0) THEN
          NFACT = 1.000D+00
          RETURN
          ENDIF
*
      NFACT = NRAY(N)
      RETURN
*
 199  WRITE(6,299)
 299  FORMAT('  ERROR IN FUNCTION NFACT: N = ',I4)
      RETURN
      END
```

Appendix C

MREAL1 OPTIMIZER

```
*************************************************************************
*                                                                       *
*                         MREAL1 OPTIMIZER                              *
*                                                                       *
*              COPYRIGHT 1986 BY THOMAS PRESTON MOORE                   *
*                                                                       *
*              Written in FORTRAN 77 (IBM VS-FORTRAN)                   *
*                                                                       *
*************************************************************************


*        ******************************************
*        *                                        *
*        *           Begin MAIN Program           *
*        *                                        *
*        ******************************************


*          *** INTEGER VARIABLE DICTIONARY ***

*  D(2)        DEMAND IN EACH REAL SYSTEM, I.
*  DESIGN(2)   DESIGN NUMBER FOR CURRENT DESIGN IN EACH REAL SYSTEM, I.
*  FLAG1       RESULT OF CONSTRAINT CHECK: 0 = FEASIBLE, 1+ = INFEASIBLE.
*  NREALS      NUMBER OF REAL SYSTEMS UNDER CONSIDERATION.
*  RMAX(2,2)   MAXIMUM RETIREMENT AGE FOR DESIGN J, REAL SYSTEM I.
*  S(2)        NUMBER OF DESIGNS UNDER CONSIDERATION IN REAL SYSTEM I.
*  SH(2)       NUMBER OF SHORTAGES CONSIDERED CATASTROPHIC IN REALS I.
*  XB(6)       COORDINATES OF CURRENT BASE SOLUTION.
*  X1(6)       COORDINATES OF XB+.
*  X2(6)       COORDINATES OF XB-.

*          **** REAL VARIABLE DICTIONARY ****

*  BCAT(2)     MAXIMUM PROBABILITY OF A CATASTROPHIC SHORTAGE OCCURRING.
*  BOP         MAXIMUM POSSIBLE ANNUAL OPERATING BUDGET.
*  BPR         MAXIMUM POSSIBLE ANNUAL REPLACEMENT CAPITAL BUDGET.
*  BUNA(2)     MAXIMUM EXPECTED SHORTAGES AS A PERCENT OF DEMAND IN REAL
*              SYSTEM I.
*  C(2)        COST OF ONE SHORTAGE-YEAR IN REAL SYSTEM I.
*  INT         ANNUAL INTEREST RATE (10% ASSUMED).
*  NORM1(2)    FINITE SOURCE QUEUEING MODEL NORMALIZATION FACTOR, .
*  O(2,2)      ANNUAL OPERATING COST OF ONE REPAIRABLE EQUIPMENT FOR
*              DESIGN J, REAL SYSTEM I.
*  P(2,2)      PROCUREMENT COST OF ONE REPAIRABLE EQUIPMENT FOR DESIGN
*              J, REAL SYSTEM I.
*  RF(2)       SALVAGE VALUE OF A REPAIR FACILITY IN REAL SYSTEM I
*              AFTER RR(I) YEARS.
*  RHO(2,2,40) MEAN RHO VALUE FOR SYSTEM IF DESIGN J IS USED IN REAL
*              SYSTEM I, WITH RETIREMENT OCCURRING IN YEAR K.
*  RO(2)       ANNUAL OPERATING COST OF REPAIR FACILITY IN REAL SYSTEM I.
*  RP(2)       PROCUREMENT COST OF A REPAIR FACILITY IN REAL SYSTEM I.
*  RR(2)       REPAIR FACILITY RETIREMENT AGE IN REAL SYSTEM I.
*  ZB          VALUE OF ALAG OR OBJECTIVE FUNCTION FOR SOLUTION XB.
*  Z1          VALUE OF ALAG OR OBJECTIVE FUNCTION FOR SOLUTION X1.
*  Z2          VALUE OF ALAG OR OBJECTIVE FUNCTION FOR SOLUTION X2.
```

```
*          ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   BESTX(6), COUNT2,
     $ DV(4,8), FDSGN(2), L1, L2, XTEMP1(6), XTEMP2(6)

*          ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,
     $    BESTZ, GFINAL(6), SPM(4,7), ZTEMP, ZTEMP1, ZTEMP2

*          ******** LABELED COMMON BLOCKS *******

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2

      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN

      COMMON /FINAL/ BESTX, BESTZ, COUNT2, DV, FDSGN, GFINAL, SPM, ZTEMP

 900  FORMAT(1X,A)
 901  FORMAT(35X,A)

*          *** WRITE HEADER FOR OUTPUT TABLE ***

      WRITE(6,901)
     $'************************************************************'
      WRITE(6,901)
     $'*                                                          *'
      WRITE(6,901)
     $'*                    MREAL1 OPTIMIZER                      *'
      WRITE(6,901)
     $'*                                                          *'
      WRITE(6,901)
     $'*          COPYRIGHT 1986 - THOMAS PRESTON MOORE           *'
      WRITE(6,901)
     $'*                                                          *'
      WRITE(6,901)
     $'*          WRITTEN IN FORTRAN 77 (IBM VS-FORTRAN)          *'
      WRITE(6,901)
     $'*                                                          *'
      WRITE(6,901)
     $'*               LINE PRINTER OUTPUT FORMAT                 *'
      WRITE(6,901)
     $'*                                                          *'
      WRITE(6,901)
     $'************************************************************'
      WRITE(6,901) ' '
      WRITE(6,901) ' '

*          *** INITIALIZE FINAL TOTAL COST VARIABLE ***

      BESTZ = 7.1D+74
      COUNT2 = 1

*          ******* SET DESIGN COMBINATION *******

      DO 1  L1 = 1, S(1)
         DO 2  L2 = 1, S(2)

            DESIGN(1) = L1
            DESIGN(2) = L2
            ZTEMP = 7.1D+74
```

```
*          *** GENERATE INITIAL XB BY HEURISTIC GUESS. ***

           CALL INIT

*          *** PERFORM SEARCH PROCEDURE FOR THIS XB. ***

           CALL DUNLOP
           CALL RESULT

*          *** RECORD THE RESULTING BEST SOLUTION. ***

           ZTEMP1 = ZBEST
           DO 3  I = 1, DIMNSN
              XTEMP1(I) = XBEST(I)
 3         CONTINUE

*          *** GENERATE A NEW XB FROM THIS BEST POINT. ***
*          *** THIS IS THE "UPPER" NEW XB.             ***

 100       CALL INIT1(XTEMP1)

*          *** PERFORM SEARCH PROCEDURE FOR THIS XB. ***

           CALL DUNLOP
           CALL RESULT

*          *** RECORD THE RESULTING BEST SOLUTION. ***

           ZTEMP2 = ZBEST
           DO 4  I = 1, DIMNSN
              XTEMP2(I) = XBEST(I)
 4         CONTINUE

*          *** GENERATE THE "LOWER" NEW XB FROM XTEMP1. ***

           CALL INIT2(XTEMP1)

*          *** PERFORM SEARCH PROCEDURE FOR THIS XB. ***

           CALL DUNLOP
           CALL RESULT

*          *** CHECK THE STOPPING CONDITION (NO IMPROVEMENT ***
*          *** OF ZBEST IN THE LAST TWO SEARCHES).          ***

           IF (ZTEMP1.GT.ZTEMP2.OR.ZTEMP1.GT.ZBEST) THEN

*          *** OTHERWISE, SEEK A NEW BEST SOLUTION AND GO ***
*          *** BACK TO LINE 100 AND REPEAT THE PROCESS.   ***

              IF (ZTEMP2.LE.ZBEST) THEN
                 ZTEMP1 = ZTEMP2
                 DO 5  I = 1, DIMNSN
                    XTEMP1(I) = XTEMP2(I)
 5               CONTINUE
                 GO TO 100
              ENDIF

              IF (ZTEMP2.GT.ZBEST) THEN
                 ZTEMP1 = ZBEST
                 DO 6  I = 1, DIMNSN
                    XTEMP1(I) = XBEST(I)
 6               CONTINUE
                 GO TO 100
              ENDIF
```

```
          ENDIF
*         *** INCREMENT DESIGN COMBINATION COUNTER ***
          COUNT2 = COUNT2 + 1
 2      CONTINUE
 1    CONTINUE
*         ******** CALL FOR FINAL REPORT *******
      CALL MANOUT
      END
*       *******************************************
*       *                                         *
*       *     TEMPORARY BLOCK DATA INPUT AREA     *
*       *                                         *
*       *******************************************

      BLOCK DATA TOM
      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6)
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN

      DATA NREALS, DIMNSN, D, S /2, 6, 10, 15, 2, 2/
      DATA RMAX, SH      /15, 25, 20, 30, 3, 5/
      DATA INT, P        /0.10, 1.4E5, 3.5E5, 1.6E5, 4.2E5/
      DATA O, C          /1.0E5, 0.7E5, 0.9E5, 0.65E5, 18.25E5, 54.75E5/
      DATA RP, RF, RO    /3.0E5, 5.0E5, 0.8E5, 1.2E5, 0.9E5, 1.3E5/
      DATA RR, BPR, BOP  /15., 15., 0.6E6, 3.0E6/
      DATA BUNA, BCAT    /5.0E7, 1.0E7, 1.0E6, 1.0E5/
      DATA (((RHO(I,J,K),I=1,2),J=1,2),K=1,8)/.0250, .010, .0240, .0060,
     $          0.0175,  0.0043,  0.0177,  0.0033,
     $        0.014545,  0.0027,  0.0153,  0.0021,
     $        0.013333,  0.0020,  0.0141,  0.0016,
     $        0.012954,  0.0016,  0.0133,  0.0013,
     $        0.013333,  0.0014,  0.0129,  0.0012,
     $        0.013809,  0.0013,  0.0129,  0.0011,
     $        0.014366,  0.0012,  0.0129,  0.0010/
      DATA (((RHO(I,J,K),I=1,2),J=1,2),K=9,16)/.015, .0011, .0132,.0009,
     $        0.015529,  0.0012,  0.0134,  0.0009,
     $        0.016196,  0.0011,  0.0136,  0.0009,
     $       0.0173196,  0.0012,  0.0140,  0.0009,
     $          0.0188,  0.0012,  0.0143,  0.0008,
     $        0.020291,  0.0012,  0.0151,  0.0008,
     $          0.0220,  0.0013,  0.0160,  0.0008,
     $          0.0,     0.0014,  0.0171,  0.0009/
      DATA (((RHO(I,J,K),I=1,2),J=1,2),K=17,24)/.0, .0014, .0183, .0009,
     $          0.0,     0.0016,  0.0201,  0.0009,
     $          0.0,     0.0017,  0.0218,  0.0010,
     $          0.0,     0.0018,  0.0238,  0.0010,
     $          0.0,     0.0021,  0.0,     0.0011,
     $          0.0,     0.0023,  0.0,     0.0011,
     $          0.0,     0.0025,  0.0,     0.0011,
     $          0.0,     0.0028,  0.0,     0.0012/
      DATA (((RHO(I,J,K),I=1,2),J=1,2),K=25,30)/0.0, .0030, 0.0, .0012,
     $          0.0,     0.0,     0.0,     0.0014,
```

```
     $             0.0,     0.0,     0.0,     0.0015,
     $             0.0,     0.0,     0.0,     0.0016,
     $             0.0,     0.0,     0.0,     0.0017,
     $             0.0,     0.0,     0.0,     0.0018/
      END
      SUBROUTINE DUNLOP

*              **************************************
*              *                                    *
*              *    DISCRETE, UNARMED, NON-LINEAR   *
*              *              OPTIMIZER             *
*              *                                    *
*              **************************************


*           ***** INTEGER VARIABLE DICTIONARY *****

*  FLAG1    RESULT OF CONSTRAINT CHECK: 0 = INFEASIBLE, 1 = FEASIBLE.
*  FLAG2    COUNTER FOR BIG M PENALTY METHOD.
*  FLAG3    COUNTER FOR BARRIER METHOD.

*           ****** REAL VARIABLE DICTIONARY ******

*  BMU      BARRIER METHOD MU.
*  PMU      PENALTY METHOD MU.
*  ZB       OBJECTIVE FUNCTION VALUE FOR BASE POINT.
*  ZBEST    BEST OBJECTIVE FUNCTION VALUE FOUND SO FAR.


*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),    FLAGM

*           ********** REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2

*           ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2

*           *** FORMAT STATEMENTS ***

 900  FORMAT(1X,A)

*           *** WRITE HEADER FOR OUTPUT TABLE ***

      WRITE (6,900) ' |<***** DECISION VARIABLES ****>|  |<*************
     $** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ****
     $***********>|'
      WRITE (6,900) ' DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FL
     $AGC       APRC       AOPC          ASUNA(1)      ASUNA(2)      ACUNA(1)
     $    ACUNA(2)'

*           *** PERFORM UNCONSTRAINED OPTIMIZATION OF THE ***
*           *** AUGMENTED LAGRANGIAN PENALTY FUNCTION     ***

 1    CALL INNER

*           *** DO CONVERGENCE CHECK ***

      CALL LOCMIN (FLAGM)
```

```
*          *** IF FLAGM = 1, WE HAVE CONVERGENCE! (SORT OF...) ***

      IF (FLAGM.EQ.1) RETURN

*          *** IF NO CONVERGENCE, PERFORM OUTER ITERATION ***

      CALL OUTER

*          *** CHECK FOR TOO MANY TIMES THRU MAIN LOOP ***

      IF (IC.LE.4) GO TO 1

      WRITE (6,900) ' FROM DUNLOP: ITERATION COUNTER HAS EXCEEDED 04 - W
     $E HAVE A PROBLEM.'

      RETURN
      END
      SUBROUTINE RESULT
*          **************************************
*          *                                    *
*          *   WRITES RESULTS FOR DESIGN COMBI- *
*          *   NATION (I,J) AND RECORDS BEST    *
*          *   SOLUTION FOUND SO FAR.           *
*          *                                    *
*          **************************************

*              *** INTEGER VARIABLE DICTIONARY ***

*  BESTX(6)   CONTAINS VALUES OF THE DECISION VARIABLES FOR THE OPTIMAL
*             SOLUTION.

*                *** REAL VARIABLE DICTIONARY ***

*  BESTZ      TOTAL COST OF OPTIMAL SOLUTION.
*  GFINAL(6)  SLACK VALUES OF THE 6 CONSTRAINTS AT THE OPTIMAL SOLUTION.

*          ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6), BESTX(6), COUNT2,
     $ DV(4,8), FDSGN(2), FLAGC(6)

*          ********** REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,
     $    BESTZ, GFINAL(6), SPM(4,7), ZTEMP

*          ******** LABELED COMMON BLOCKS *******

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN
      COMMON /FINAL/ BESTX, BESTZ, COUNT2, DV, FDSGN, GFINAL, SPM, ZTEMP

 900  FORMAT(/1X,A)
 901  FORMAT(4X,I1,7X,I1,4X,6I3,2X,F12.0,7X,2F11.0,2X,4(E12.6,1X)/)
 902  FORMAT(I1)

*          *** RECHECK FINAL SOLUTION FROM DUNLOP FOR FEASIBILITY ***

      CALL CNSTR1(XBEST,FLAGC)
```

```
      IF (FLAG1.NE.0) THEN
         WRITE(6,900)  'WARNING FROM RESULT: FINAL SOLUTION FROM DUNLOP
     $IS NOT FEASIBLE.'
         WRITE(6,901) DESIGN, XBEST, ZBEST, GEE
         RETURN
      ENDIF

*           *** WRITE OPTIMAL SOLUTION, TOTAL COST AND SLACKS ***

      WRITE(6,900) 'FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINAT
     $ION IS:'
      WRITE (6,901) DESIGN, XBEST, ZBEST, GEE

*           *** CAUSE A PAGE EJECT ON THE PRINTER ***

      I = 1
      WRITE(6,902) I

*           *** CHECK TO SEE IF THIS IS THE BEST SOLUTION FOUND ***
*           *** FROM AMONG ALL DESIGN COMBINATIONS SO FAR.  IF  ***
*           *** SO, RECORD IT IN BESTX, BESTZ AND GFINAL.       ***

      IF (ZBEST.LT.BESTZ) THEN
         DO 1  I = 1, DIMNSN
            BESTX(I) = XBEST(I)
 1       CONTINUE
         BESTZ = ZBEST
         DO 2  I = 1, NREALS
            FDSGN(I) = DESIGN(I)
 2       CONTINUE
         DO 3 J = 1, 6
            GFINAL(J) = GEE(J)
 3       CONTINUE
      ENDIF

*           *** RECORD BEST DECISION VARIABLE AND PERFORMANCE ***
*           *** MEASURE VALUES IN DV AND SPM RESPECTIVELY.    ***

      IF (ZBEST.LT.ZTEMP) THEN
         DV(COUNT2,1) = DESIGN(1)
         DV(COUNT2,2) = DESIGN(2)
         DV(COUNT2,3) = XBEST(1)
         DV(COUNT2,4) = XBEST(2)
         DV(COUNT2,5) = XBEST(3)
         DV(COUNT2,6) = XBEST(4)
         DV(COUNT2,7) = XBEST(5)
         DV(COUNT2,8) = XBEST(6)
         SPM(COUNT2,1) = ZBEST
         SPM(COUNT2,2) = GEE(1)
         SPM(COUNT2,3) = GEE(2)
         SPM(COUNT2,4) = GEE(3)
         SPM(COUNT2,5) = GEE(4)
         SPM(COUNT2,6) = GEE(5)
         SPM(COUNT2,7) = GEE(6)
         ZTEMP = ZBEST
      ENDIF

      RETURN
      END
      SUBROUTINE MANOUT

*           **************************************
*           *                                    *
*           *   WRITES FINAL RESULTS FOR MREAL1  *
*           *                                    *
*           **************************************
```

```
*           *** INTEGER VARIABLE DICTIONARY ***

*  BESTX(6)    CONTAINS VALUES OF THE DECISION VARIABLES FOR THE OPTIMAL
*              SOLUTION.

*           **** REAL VARIABLE DICTIONARY ****

*  BESTZ       TOTAL COST OF OPTIMAL SOLUTION.
*  GFINAL(6)   SLACK VALUES OF THE 6 CONSTRAINTS AT THE OPTIMAL SOLUTION.

*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   BESTX(6), COUNT2,
     $ DV(4,8), FDSGN(2)

*           ********* REAL DECLARATIONS *********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,
     $    BESTZ, GFINAL(6), SPM(4,7), ZTEMP

*           ******** LABELED COMMON BLOCKS *******

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      COMMON /FINAL/ BESTX, BESTZ, COUNT2, DV, FDSGN, GFINAL, SPM, ZTEMP

 900  FORMAT(1X,A)
 901  FORMAT(5X,I1,9X,I1,5X,6I3,1X,E12.7,1X,2(F10.0,1X),4(E12.6,1X))
 902  FORMAT(1X)
 903  FORMAT(' BUS DESIGN ',I1,4X,F13.2,5X,F13.2,7X,F9.6,8X,F10.8,6X,I2)
 904  FORMAT(' SYSTEMS TOTAL:  ',F13.2,5X,F13.2)
 905  FORMAT(' BUDGET: ',8X,F13.2,5X,F13.2)
 906  FORMAT(' RAILCAR DESIGN ',I1,F13.2,5X,F13.2,7X,F9.6,8X,F10.8,6X,I2
     $)

*           *** WRITE HEADER FOR SUMMARY SECTION ***

      WRITE(6,900) '                                      SUMMARY OF RESULTS
     $ FROM THE MREAL1 OPTIMIZER'
      WRITE(6,900) ' '
      WRITE(6,900) 'OPTIMAL SOLUTIONS FOR EACH DESIGN COMBINATION:'
      WRITE(6,900) ' '
      WRITE (6,900) ' DESIGN(1) DESIGN(2) N1 M1 R1 N2 M2 R2  TOTAL COST
     $      APRC        AOPC       ASUNA(1)     ASUNA(2)     ACUNA(1)     A
     $SUNA(2)'

*           *** WRITE RESULTS FOR EACH DESIGN COMBINATION ***

      DO 1  I = 1, S(1)*S(2)
         WRITE(6,901) DV(I,1),DV(I,2),DV(I,3),DV(I,4),DV(I,5),DV(I,6),
     $    DV(I,7),DV(I,8),SPM(I,1),SPM(I,2),SPM(I,3),SPM(I,4),SPM(I,5),
     $    SPM(I,6),SPM(I,7)
 1    CONTINUE

*           *** WRITE HEADER FOR OPTIMAL SOLUTION ***

      WRITE(6,902)
      WRITE(6,902)
      WRITE(6,900)  'FROM MANOUT: THE OPTIMAL SOLUTION TO PROBLEM MREAL1
     $ (TEST) IS:'
      WRITE (6,900) ' DESIGN(1) DESIGN(2) N1 M1 R1 N2 M2 R2  TOTAL COST
     $      APRC        AOPC       ASUNA(1)     ASUNA(2)     ACUNA(1)     A
     $SUNA(2)'
```

```
*          *** WRITE OPTIMAL SOLUTION, TOTAL COST AND SLACKS ***

      WRITE (6,901) FDSGN, BESTX, BESTZ, GFINAL

*          *** WRITE DETAILED DESCRIPTION OF OPTIMAL SOLUTION ***

      WRITE(6,902)
      WRITE(6,902)
      WRITE(6,900) '                                   DETAILED SUMMARY OF
     $ THE OPTIMAL SOLUTION'
      WRITE(6,902)
      WRITE(6,900) '                 ANNUAL REPLACEMENT  ANNUAL OPERATING
     $EXPECTED NUMBER  PROBABILITY OF N'
      WRITE(6,900) 'REAL SYSTEM      CAPITAL COST           COST
     $ OF SHORTAGES    OR MORE SHORTAGES  N'
      WRITE(6,902)

*          *** COMPUTE DETAILED VALUES AND WRITE THEM ***

      DO 2  I = 1, NREALS
         DESIGN(I) = FDSGN(I)
         TEMP1 = CPR(I,BESTX)
         TEMP2 = COP(I,BESTX)
         TEMP3 = SUNA1(I,BESTX)
         TEMP4 = CUNA(I,BESTX)/1.0E10
         IF (I.EQ.1) WRITE(6,903) FDSGN(I), TEMP1, TEMP2, TEMP3, TEMP4,
     $               SH(I)
         IF (I.EQ.2) WRITE(6,906) FDSGN(I), TEMP1, TEMP2, TEMP3, TEMP4,
     $               SH(I)
 2    CONTINUE

      TEMP1 = BPR - GFINAL(1)
      TEMP2 = BOP - GFINAL(2)

      WRITE(6,904) TEMP1, TEMP2
      WRITE(6,905) BPR, BOP

      RETURN
      END
      SUBROUTINE INIT

*                *****************************************
*                *                                       *
*                *   GENERATES INITIAL SEARCH POINT, XB, *
*                *   SETS ITERATION COUNTER, AND INI-    *
*                *   TIALIZES THE PENALTY PARAMETERS,    *
*                *   LAGRANGE MULTIPLIERS AND MAXIMUM    *
*                *   CONSTRAINT VIOLATION.               *
*                *                                       *
*                *****************************************

*          ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   BESTX(6), COUNT2,
     $ DV(4,8), FDSGN(2)

*          ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,
     $    BESTZ, GFINAL(6), SPM(4,7), ZTEMP

*          ******** LABELED COMMON BLOCK ********
```

```
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      COMMON /FINAL/ BESTX, BESTZ, COUNT2, DV, FDSGN, GFINAL, SPM, ZTEMP
      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN

*           *** SET ITERATION COUNTER FOR THIS DESIGN COMBINATION ***

      IC = 0

*           *** SET INITIAL BASE POINT, XB ***

         DO 1  I = 1, NREALS
            XB(3*I-2) = D(I) + 1
            XB(3*I-1) = 2
            XB(3*I)   = NINT(0.75*RMAX(I,DESIGN(I)))
 1       CONTINUE

*           *** SET LAGRANGE MULTIPLIERS AND PENALTY PARAMETERS ***

 60   DO 3  J = 1, 6
         PRHO(J) = 1.0
         LAMBDA(J) = 0.0
 3    CONTINUE

*           *** CHECK CONSTRAINTS AND OBTAIN MAXIMUM ***
*           *** CONSTRAINT VIOLATION                 ***

      CALL CONSTR(XB)
      MAXG2 = MAXG1

*           *** WRITE INITIAL BASE POINT FOR THIS DESIGN COMBO ***

      WRITE(6,900) DESIGN
 900  FORMAT(1X,'                                  DESIGN COMBINATION:   E
     $QUIPMENT 1:  ',I1,'      EQUIPMENT 2:  ',I1/)

      WRITE(6,901) XB, PRHO, LAMBDA
 901  FORMAT(' FROM INIT: INITIAL BASE POINT:            ',6I5/
     $     1X,'FROM INIT: INITIAL PENALTY PARAMETERS:    ',6(F3.1,2X)/
     $     1X,'FROM INIT: INITIAL LAGRANGE MULTIPLIERS:  ',6(F3.1,2X)/)

      RETURN
      END
      SUBROUTINE INIT1(XTEMP1)

*                ****************************************
*                *                                      *
*                *   GENERATES "UPPER" INITIAL SEARCH   *
*                *   POINT, SETS ITERATION COUNTER, AND  *
*                *   INITIALIZES THE PENALTY PARAMETERS, *
*                *   LAGRANGE MULTIPLIERS AND MAXIMUM    *
*                *   CONSTRAINT VIOLATION.              *
*                *                                      *
*                ****************************************

*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   BESTX(6), COUNT2,
     $ DV(4,8), FDSGN(2),    XTEMP1(6)

*           ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
```

```
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,
     $    BESTZ, GFINAL(6), SPM(4,7), ZTEMP

*          ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      COMMON /FINAL/ BESTX, BESTZ, COUNT2, DV, FDSGN, GFINAL, SPM, ZTEMP
      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN

*          *** SET ITERATION COUNTER FOR THIS DESIGN COMBINATION ***

      IC = 0

*          *** SET NEW INITIAL BASE POINT, XB ***

         XB(1) = XTEMP1(1) + 2
         XB(2) = MAX0(1,XTEMP1(2)-2)
         XB(3) = MAX0(1,NINT(0.7*XTEMP1(3)))
         XB(4) = XTEMP1(4) + 2
         XB(5) = MAX0(1,XTEMP1(5)-2)
         XB(6) = MAX0(1,NINT(0.7*XTEMP1(6)))

*          *** SET LAGRANGE MULTIPLIERS AND PENALTY PARAMETERS ***

 60   DO 3  J = 1, 6
         PRHO(J) = 1.0
         LAMBDA(J) = 0.0
 3    CONTINUE

*          *** CHECK CONSTRAINTS AND OBTAIN MAXIMUM ***
*          *** CONSTRAINT VIOLATION                 ***

      CALL CONSTR(XB)
      MAXG2 = MAXG1

*          *** WRITE INITIAL BASE POINT FOR THIS DESIGN COMBO ***

      WRITE(6,900) DESIGN
 900  FORMAT(1X,'                                  DESIGN COMBINATION:   E
     $QUIPMENT 1:  ',I1,'      EQUIPMENT 2:  ',I1/)

      WRITE(6,901) XB, PRHO, LAMBDA
 901  FORMAT(' FROM INIT1: INITIAL BASE POINT:             ',6I5/
     $     1X,'FROM INIT1: INITIAL PENALTY PARAMETERS:    ',6(F3.1,2X)/
     $     1X,'FROM INIT1: INITIAL LAGRANGE MULTIPLIERS:  ',6(F3.1,2X)/)

      RETURN
      END
      SUBROUTINE INIT2(XTEMP1)

*               *****************************************
*               *                                       *
*               *   GENERATES "LOWER" INITIAL SEARCH    *
*               *   POINT, SETS ITERATION COUNTER, AND  *
*               *   INITIALIZES THE PENALTY PARAMETERS, *
*               *   LAGRANGE MULTIPLIERS AND MAXIMUM    *
*               *   CONSTRAINT VIOLATION.               *
*               *                                       *
*               *****************************************

*          ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   BESTX(6), COUNT2,
```

```
     $ DV(4,8), FDSGN(2),    XTEMP1(6)

*          ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,
     $    BESTZ, GFINAL(6), SPM(4,7), ZTEMP

*          ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      COMMON /FINAL/ BESTX, BESTZ, COUNT2, DV, FDSGN, GFINAL, SPM, ZTEMP
      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN

*          *** SET ITERATION COUNTER FOR THIS DESIGN COMBINATION ***

      IC = 0

*          *** SET NEW INITIAL BASE POINT, XB ***

         XB(1) = MAX0(D(1),XTEMP1(1)-1)
         XB(2) = XTEMP1(2) + 1
         XB(3) = MIN0(RMAX(1,DESIGN(1)),NINT(1.3*XTEMP1(3)))
         XB(4) = MAX0(D(2),XTEMP1(4)-1)
         XB(5) = XTEMP1(5) + 1
         XB(6) = MIN0(RMAX(2,DESIGN(2)),NINT(1.3*XTEMP1(6)))

*          *** SET LAGRANGE MULTIPLIERS AND PENALTY PARAMETERS ***

 60   DO 3  J = 1, 6
         PRHO(J) = 1.0
         LAMBDA(J) = 0.0
 3    CONTINUE

*          *** CHECK CONSTRAINTS AND OBTAIN MAXIMUM ***
*          *** CONSTRAINT VIOLATION                 ***

      CALL CONSTR(XB)
      MAXG2 = MAXG1

*          *** WRITE INITIAL BASE POINT FOR THIS DESIGN COMBO ***

      WRITE(6,900) DESIGN
 900  FORMAT(1X,'                                   DESIGN COMBINATION:   E
     $QUIPMENT 1:  ',I1,'      EQUIPMENT 2:  ',I1/)

      WRITE(6,901) XB, PRHO, LAMBDA
 901  FORMAT(' FROM INIT2: INITIAL BASE POINT:           ',6I5/
     $     1X,'FROM INIT2: INITIAL PENALTY PARAMETERS:    ',6(F3.1,2X)/
     $     1X,'FROM INIT2: INITIAL LAGRANGE MULTIPLIERS:  ',6(F3.1,2X)/)

      RETURN
      END
      SUBROUTINE INNER

*             ****************************************
*             *                                      *
*             *    SUBROUTINE INNER PERFORMS AN UN-  *
*             *    CONSTRAINED MINIMIZATION OF THE   *
*             *      AUGMENTED LAGRANGIAN PENALTY    *
*             *      FUNCTION FOR PROBLEM MREAL1.    *
*             *                                      *
*             ****************************************
```

```
*           ***** INTEGER VARIABLE DICTIONARY *****

*  FLAG1    RESULT OF CONSTRAINT CHECK: 0 = INFEASIBLE, 1 = FEASIBLE.
*  FLAG2    INDICATOR FLAG FOR +DIR(*,L) AND -DIR(*,L).
*  JBEST    BEST POINT NUMBER SO FAR IN FIBONACCI LINE SEARCH.
*  IC       ITERATION COUNTER.
*  LEFT     POINT NO. OF LEFT SIDE OF FIBONACCI INTERVAL.
*  RIGHT    POINT NO. OF RIGHT SIDE OF FIBONACCI INTERVAL.
*  S1       LEFT POINT NUMBER INSIDE FIBONACCI INTERVAL.
*  S2       RIGHT POINT NO. INSIDE FIBONACCI INTERVAL.
*  XBEST(6) COORDINATES OF BEST POINT FOUND SO FAR.
*  XT(6)    COORDINATES OF TEMPORARY (SEARCH) POINT.
*  XT1(6)   COORDINATES OF S1.
*  XT2(6)   COORDINATES OF S2.
*  X1       COORDINATES OF NEXT-TO-LAST BASE POINT.
*  X2       COORDINATES OF LAST BASE POINT.

*           ****** REAL VARIABLE DICTIONARY ******

*  A(6,6)   CONTAINS A-LOOP RESULTS OF GRAM-SCHMIDT ORTHOGONALIZATION.
*  B(6,6)   CONTAINS B-LOOP RESULTS OF GRAM-SCHMIDT ORTHOGONALIZATION.
*  DCOMP(6) DIRECTION COMPONENT FROM X1 TO X2 ALONG DIRECTION I.
*  DIR(6,6) ORTHOGONALIZED DIRECTIONS FROM GRAM-SCHMIDT.
*  DIST1    DISTANCE BETWEEN XB AND XBEST.
*  GRAD(6)  COMPONENTS OF THE NORMALIZED GRADIENT APPROXIMATION.
*  LAMBDA(6)LAGRANGE MULTIPLIERS FOR CONSTRAINTS.
*  NORMB    VALUE OF THE NORM OF B(I,J), RECOMPUTED FOR EACH J.
*  NORMG    VALUE OF THE NORM OF GRAD(6).
*  PRHO(6)  PENALTY PARAMETERS FOR CONSTRAINTS.
*  RATIO1   OBJECTIVE FUNCTION PERCENT IMPROVEMENT PER UNIT DISTANCE.
*  TEMP1    VALUE OF A-DIR SCALAR FOR EACH J IN THE GRAM-SCHMIDT.
*  ZB       OBJECTIVE FUNCTION VALUE FOR BASE POINT.
*  ZBEST    BEST OBJECTIVE FUNCTION VALUE FOUND SO FAR.
*  ZT       OBJECTIVE FUNCTION VALUE FOR TEMPORARY (SEARCH) POINT.
*  ZT1      OBJECTIVE FUNCTION VALUE FOR S1.
*  ZT2      OBJECTIVE FUNCTION VALUE FOR S2.

*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   FLAG2, JBEST,
     $ LEFT, RIGHT, S1, S2, ITEMP, XT(6), XT1(6), XT2(6)

*           ********* REAL DECLARATIONS *********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,   A(6,6),
     $ B(6,6), DEL1, DIR(6,6), DIST1, DCOMP(6), GRAD(6), NORMB, NORMG,
     $ PMU, RATIO1, TEMP1, ZT, ZT1, ZT2

*           ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN

*           *** FORMAT STATEMENTS ***

 900  FORMAT (1X,A)

*           *** SET NORM OF GRADIENT = 0.0 ***
```

```
933   NORMG = 0.0

*           *** COMPUTE ZB BY THE AUGMENTED LAGRANGIAN ***
*           *** PENALTY METHOD FOR THE POINT XB.       ***

      CALL ALAG(XB,ZB)
      ZBEST = ZB

*           *** RECORD COORDINATES OF CURRENT BASE POINT ***
*           ********* IN X1(6), X2(6) AND XBEST(6) *********

      DO 20 I = 1, DIMNSN
         X1(I) = XB(I)
         X2(I) = XB(I)
         XBEST(I) = XB(I)
 20   CONTINUE

*           *** LARGE LOOP CONTROLLING THE GRADIENT APPROX- ***
*           *** IMATION FOR EACH COORDINATE AXIS (DECISION  ***
*           *** VARIABLE).                                  ***

      DO 21 I = 1, DIMNSN

*           *** COMPUTE THE UPPER ADJACENT POINT FOR THIS ***
*           *** COORDINATE AXIS (DECISION VARIABLE).      ***

         X2(I) = XB(I) + 1

*           *** CHECK IT FOR EXPLICIT FEASIBILITY ***

         CALL XPLSIT(X2)

*           *** IF NOT FEASIBLE, COMPUTE LOWER ADJACENT POINT ***

         IF (X2(I).EQ.XB(I)) THEN
            X1(I) = XB(I) - 1

*           *** CHECK LOWER ADJACENT POINT FOR FEASIBILITY ***

            CALL XPLSIT(X1)

*           *** IF INFEASIBLE, SET GRAD(I) = 0.0  (AT 1) ***

            IF (X1(I).EQ.XB(I)) GO TO 1

*           *** OTHERWISE CHECK Z1 FOR IMPROVEMENT OVER ZB ***

            CALL ALAG(X1,Z1)
            IF (Z1.LT.ZBEST) THEN
               ZBEST = Z1
               DO 6  K = 1, DIMNSN
                  XBEST(K) = X1(K)
 6             CONTINUE
            ENDIF

*           *** IF NOT IMPROVED, SET GRAD(I) = 0.0  (AT 1) ***

            IF (Z1.GE.ZB)       GO TO 1

*           *** OTHERWISE DO BACKWARDS GRADIENT APPROXIMATION ***

            GO TO 2
         ENDIF

*           *** RETURN TO CHECKING UPPER ADJACENT POINT ***
*           *** CHECK X2 FOR IMPROVEMENT OVER XB        ***
```

```
         CALL ALAG(X2,Z2)
         IF (Z2.LT.ZBEST) THEN
            ZBEST = Z2
            DO 7  K = 1, DIMNSN
               XBEST(K) = X2(K)
 7          CONTINUE
         ENDIF

*           *** IF Z2 IS NOT IMPROVED, COMPUTE LOWER ADJACENT POINT ***

         IF (Z2.GE.ZB) THEN
            X1(I) = XB(I) - 1

*           *** CHECK LOWER ADJACENT POINT FOR FEASIBILITY ***

            CALL XPLSIT(X1)

*           *** IF INFEASIBLE, SET GRAD(I) = 0.0  (AT 1). ***

            IF (X1(I).EQ.XB(I)) GO TO 1

*           *** OTHERWISE CHECK Z1 FOR IMPROVEMENT OVER ZB ***

            CALL ALAG(X1,Z1)
            IF (Z1.LT.ZBEST) THEN
               ZBEST = Z1
               DO 8  K = 1, DIMNSN
                  XBEST(K) = X1(K)
 8             CONTINUE
            ENDIF

*           *** IF NOT IMPROVED, SET GRAD(I) = 0.0  (AT 1). ***

            IF (Z1.GE.ZB)        GO TO 1

*           *** OTHERWISE DO BACKWARDS DIFFERENCE ***
*           *** GRADIENT APPROXIMATION (AT 2).    ***

            GO TO 2
         ENDIF

*           *** FORWARD DIFFERENCE GRADIENT APPROXIMATION ***

         GRAD(I) = ZB - Z2
         GO TO 5

*           *** GRADIENT COMPONENT EQUALS ZERO ***

 1       GRAD(I) = 0.0
         GO TO 5

*           *** BACKWARD DIFFERENCE GRADIENT APPROXIMATION ***

 2       GRAD(I) = Z1 - ZB
         GO TO 5

*           *** CENTRAL DIFFERENCE GRADIENT APPROXIMATION ***

 3       GRAD(I) = (Z1-Z2)/2
         GO TO 5

*           *** ACCUMULATE NORMALIZATION FACTORS AND RESET X1 & X2 ***

 5       NORMG = NORMG + GRAD(I)**2
         X1(I) = XB(I)
         X2(I) = XB(I)
 21   CONTINUE
```

```
*           *** IF ALL GRADIENT COMPONENTS = 0 THEN NORMG = 0 ***
*           *** AND WE'RE IN A LOCAL MINIMUM OR RESOLUTION.   ***
*           *** RETURN TO DUNLOP FOR START OF LOCMIN.         ***

      IF (NORMG.EQ.0.0) THEN
         WRITE(6,900) 'FROM INNER: RESOLUTION VALLEY OR LOCAL MINIMUM EN
     $COUNTERED IN GRADIENT APPOXIMATION SECTION - RETURN TO DUNLOP.'
         RETURN
      ENDIF

*           *** FINISH NORMALIZATION OF THE APPROXIMATE GRADIENT ***

      NORMG = NORMG**0.5

*           *** STORE NORMALIZED GRADIENT APPROX IN GRAD(I) ***
*           *** FIND MAX COMPONENT, STORE IT IN DEL1      ***

      DEL1 = 0.0
      DO 22 I = 1, DIMNSN
         GRAD(I) = GRAD(I)/NORMG
         DEL1 = AMAX1(DEL1,ABS(GRAD(I)))
 22   CONTINUE

*               ******************************************************
*               * START FIBONACCI LINE SEARCH IN -GRADIENT DIRECTION *
*               ******************************************************

      JBEST = 0
      J     = 1
      J1    = 1

*          *** COMPUTE COORDINATES OF NEXT SEARCH POINT, XT(I) ***

 24   DO 23 I = 1, DIMNSN
         XT(I) = NINT(XB(I) + J*GRAD(I)/DEL1)
 23   CONTINUE

*          *** CHECK NEXT SEARCH POINT FOR EXPLICIT FEASIBILITY ***
*          *** & COMPUTE CORRESPONDING OBJECTIVE FUNCTION VALUE ***

      CALL ALAG1(XT,ZT)

*          *** IF XT IS BETTER THAN XBEST, RESET XBEST, JBEST, ZBEST ***

      IF (ZT.LE.ZBEST) THEN
         JBEST = J
         ZBEST = ZT
         DO 26 I = 1, DIMNSN
            XBEST(I) = XT(I)
 26      CONTINUE

*          *** SET INDICIES J AND J1 FOR NEXT FIBONACCI SEARCH POINT ***
*          *** AND RETURN TO COMPUTE NEW XT AT STATEMENT 24.         ***

         J  = J + J1
         J1 = J - J1
         GO TO 24
      ENDIF

*          *** XT ISN'T BETTER THAN XBEST, SO CHECK IF FIBONACCI ***
*          *** POINT COUNTER IS L.E. 3.  IF SO, FIBONACCI SEARCH ***
*          *** IS COMPLETED.  GO TO ROSENBROCKS METHOD, LINE 50. ***
*          *** OTHERWISE BEGIN FIBONACCI INTERVAL SEARCH.        ***

      IF (J.LE.3) GO TO 50
```

```
*          ****************************************
*          * BEGIN FIBONACCI INTERVAL SEARCH (GRAD) *
*          ****************************************

      JBEST = J1
      LEFT  = J - J1
      RIGHT = J
      S1    = J1
      ZT1   = ZBEST
      DO 27 I = 1, DIMNSN
         XT1(I) = XBEST(I)
 27   CONTINUE

*          *** RESET COUNTERS TO PREP FOR FIBONACCI INTERVAL SEARCH ***

      DO 28 N = 1, 3
         J1 = J - J1
         J  = J - J1
 28   CONTINUE

*          *** COMPUTE INNER RIGHT FIBONACCI POINT, S2 ***

 33   S2 = RIGHT - J

*          *** COMPUTE COORDINATES OF POINT S2 ***

      DO 29 I = 1, DIMNSN
         XT2(I) = NINT(XB(I)+S2*GRAD(I)/DEL1)
 29   CONTINUE

*          *** CHECK S2 FOR EXPLICIT FEASIBILITY AND COMPUTE ZT2 ***

      CALL ALAG1(XT2,ZT2)

*          *** DETERMINE WHICH INTERIOR POINT, S1 OR S2, IS BETTER ***
*          *** AND BEGIN LEFT FIBONACCI INTERVAL REDUCTION SECTION ***

 34   IF (ZT1.GT.ZT2) THEN
         JBEST = S2
         ZBEST = ZT2
         DO 30 I = 1, DIMNSN
            XBEST(I) = XT2(I)
 30      CONTINUE

*          *** IF J=1, THEN THE INTERVAL HAS BEEN COMPLETELY ***
*          *** REDUCED, SO GO TO ROSENBROCK ***

         IF (J.LE.1) GO TO 50

*          *** OTHERWISE DO LEFT REDUCTION AND RETURN TO 33 ***

         J1   = J - J1
         J    = J - J1
         LEFT = S1
         S1   = S2
         ZT1  = ZT2
         DO 31 I = 1, DIMNSN
            XT1(I) = XT2(I)
 31      CONTINUE
         GO TO 33
      ENDIF

*          *** END OF LEFT INTERVAL REDUCTION SECTION ***
*          *** BEGIN RIGHT INTERVAL REDUCTION SECTION ***

      JBEST = S1
      ZBEST = ZT1
```

```
      DO 32 I = 1, DIMNSN
         XBEST(I) = XT1(I)
 32   CONTINUE

*          *** IF J=1, THEN THE INTERVAL HAS BEEN COMPLETELY ***
*          *** REDUCED, SO GO TO ROSENBROCK ***

      IF (J.LE.1) GO TO 50

*          *** OTHERWISE DO RIGHT REDUCTION AND RETURN TO 34 ***

      J1    = J - J1
      J     = J - J1
      RIGHT = S2
      S2    = S1
      S1    = LEFT + J
      ZT2   = ZT1
      DO 35 I = 1, DIMNSN
         XT2(I) = XT1(I)
         XT1(I) = NINT(XB(I) + S1*GRAD(I)/DEL1)
 35   CONTINUE

*          *** CHECK S1 FOR EXPLICIT FEASIBILITY AND COMPUTE ZT1 ***

      CALL ALAG1(XT1,ZT1)
      GO TO 34

*          *** END OF RIGHT INTERVAL REDUCTION SECTION ***

*          *** COMPUTE THE MAXIMUM OF THE COMPONENT DIFFERENCES ***
*          *** BETWEEN XB AND XBEST.                            ***

 50   ITEMP = 0
      DO 40 I = 1, DIMNSN
         XT(I) = IABS(XB(I)-XBEST(I))
         ITEMP = MAX0(ITEMP,XT(I))
 40   CONTINUE

*           *** CHECK IF XB = XBEST.  IF SO, RETURN TO DUNLOP ***

      IF (ITEMP.LT.1) THEN
         WRITE (6,900) 'GRADIENT SEARCH PROCEDURE HAS NOT MOVED FROM THE
     $ PREVIOUS BASE POINT - RETURN TO DUNLOP.'
         RETURN
      ENDIF

*           *** OTHERWISE, COMPUTE THE DISTANCE COMPONENTS ***
*           *** AND RESET XB = XBEST.                      ***

      DIST1 = 0.0
      DO 51 I = 1, DIMNSN
         DCOMP(I) = ABS(REAL(XB(I) - XBEST(I)))
         DIST1 = DIST1 + DCOMP(I)**2
         XB(I) = XBEST(I)
 51   CONTINUE

*               ********************************
*               * BEGIN ROSENBROCK'S PROCEDURE *
*               ********************************

*          *** SET INITIAL DIRECTION VECTORS ***

      DO 60  I = 1, DIMNSN
         DO 61  J = 1, DIMNSN
            DIR(I,J) = 0.0
            IF (I.EQ.J) DIR(I,J) = 1.0
 61      CONTINUE
```

```
 60   CONTINUE

*            **************************************
*            * BEGIN GRAM-SCHMIDT ORTHOGONALIZATION *
*            **************************************

*           *** RECORD THE CURRENT BEST SOLUTION IN X1. ***
*           *** AT THE END OF THIS ITERATION OF ROSEN-  ***
*           *** BROCK'S PROCEDURE, X1 AND XBEST WILL BE ***
*           *** COMPARED.  A RETURN TO DUNLOP WILL BE   ***
*           *** EXECUTED IF THEY ARE EQUAL.             ***

 150  DO 59  I = 1, DIMNSN
         X1(I) = XBEST(I)
 59   CONTINUE

*           *** LOOP TO COMPUTE A(I,J) ***

      DO 62  J = 1, DIMNSN

         IF (DCOMP(J).EQ.0.0) THEN
            DO 63  I = 1, DIMNSN
               A(I,J) = DIR(I,J)
 63         CONTINUE
         ENDIF

         IF (DCOMP(J).NE.0.0) THEN
            DO 64  I = 1, DIMNSN
               A(I,J) = 0.0
               DO 65  K = J, DIMNSN
                  A(I,J) = A(I,J) + DCOMP(K)*DIR(I,K)
 65            CONTINUE
 64         CONTINUE
         ENDIF

 62   CONTINUE

*          *** COMPUTE DIR(I,1) (B SUB 1) ***

      NORMB = 0.0
      DO 66  I = 1, DIMNSN
         NORMB = NORMB + A(I,1)**2
 66   CONTINUE

      NORMB = NORMB**0.5

      DO 67  I = 1, DIMNSN
         DIR(I,1) = A(I,1)/NORMB
 67   CONTINUE

*          *** START MAIN LOOP FOR COMPUTING NEW DIR(I,J) ***

      DO 70  J = 2, DIMNSN

*            *** INITIALIZE THE B(I,J) ***

         DO 71  I = 1, DIMNSN
            B(I,J) = 0.0
 71      CONTINUE

*            *** COMPUTE & APPLY THE A(I,J) DIR(I,K) SCALAR ***

         DO 72  K = 1, J-1

            TEMP1 = 0.0

            DO 73  L = 1, DIMNSN
```

```
            TEMP1 = TEMP1 + A(L,J)*DIR(L,K)
 73       CONTINUE

          DO 74  I = 1, DIMNSN
            B(I,J) = B(I,J) + TEMP1*DIR(I,K)
 74       CONTINUE

 72     CONTINUE
        NORMB = 0.0

*             *** COMPUTE THE B(I,J) THEN THE DIR(I,J) ***

        DO 75  I = 1, DIMNSN
          B(I,J) = A(I,J) - B(I,J)
          NORMB = NORMB + B(I,J)**2

 75     CONTINUE

        NORMB = NORMB**0.5

        DO 76  I = 1, DIMNSN
          DIR(I,J) = B(I,J)/NORMB
 76     CONTINUE
 70   CONTINUE

*            ******************************************
*            * END OF GRAM-SCHMIDT ORTHOGONALIZATION. *
*            * PERFORM FIBONACCI SEARCHES IN THE NEW  *
*            * DIRECTIONS (ROSENBROCK).               *
*            ******************************************

      ZB = ZBEST
      Z1 = ZBEST

*            ******************************************************
*            * LOOP ONCE (ON L) FOR EACH OF THE DIMNSN DIRECTIONS *
*            * THIS IS A SEARCH IN DIRECTION DIR(I,L)             *
*            ******************************************************

      DO 81  L = 1, DIMNSN

*          *** FIND DEL1 FOR THIS SEARCH DIRECTION ***

        DEL1 = 0.0
        DO 82  I = 1, DIMNSN
          IF (ABS(DIR(I,L)).GT.DEL1)  DEL1 = ABS(DIR(I,L))
 82     CONTINUE

*          *** SET THE FIBONACCI COUNTERS AND THE ***
*          *** REVERSED DIRECTION FLAG.           ***

        JBEST = 0
        J     = 1
        J1    = 1
        FLAG2 = 0

*          *** COMPUTE COORDINATES OF NEXT SEARCH POINT, ***
*          *** XT(I), IN THE DIR(*,L) DIRECTION.         ***

 84     DO 83 I = 1, DIMNSN
          XT(I) = NINT(XB(I) + J*DIR(I,L)/DEL1)
 83     CONTINUE

*          *** CHECK THIS SEARCH POINT FOR EXPLICIT FEASIBILITY ***
*          *** & COMPUTE CORRESPONDING OBJECTIVE FUNCTION VALUE ***

        CALL ALAG1(XT,ZT)
```

```
*          *** IF XT IS BETTER THAN XBEST, RESET XBEST, JBEST, ZBEST ***

        IF (ZT.LE.ZBEST) THEN
           JBEST = J
           ZBEST = ZT
           DO 86 I = 1, DIMNSN
              XBEST(I) = XT(I)
 86        CONTINUE

*          *** SET INDICIES J AND J1 FOR NEXT FIBONACCI SEARCH POINT ***
*          *** AND RETURN TO COMPUTE NEW XT AT STATEMENT 84 ***

           J  = J + J1
           J1 = J - J1
           GO TO 84
        ENDIF

*          *** IF XT ISN'T BETTER THAN XBEST, CHECK JBEST AND FLAG2 ***
*          *** TO SEE IF THE FIRST POINT IN THE DIRECTION +DIR(*,L) ***
*          *** IS THE CULPRIT.  IF SO, TURN SEARCH IN DIRECTION     ***
*          *** -DIR(*,L).                                           ***

        IF (JBEST.LE.0.AND.FLAG2.NE.-1) THEN
           DEL1 = -DEL1
           FLAG2 = -1
           GO TO 84
        ENDIF

*          *** IF XT ISN'T BETTER THAN XBEST, CHECK TO SEE IF THIS ***
*          *** IS THE FIRST POINT EXAMINED IN THE -DIR(*,L) DIREC- ***
*          *** TION.  IF SO, LOOP ON THE NEXT DIRECTION.           ***

        IF (JBEST.LE.0.AND.FLAG2.EQ.-1) GO TO 100

*          *** CHECK IF FIBONACCI POINT COUNTER IS LESS THAN OR    ***
*          *** EQUAL TO 3.  IF SO, XBEST IS KNOWN - LOOP ON THE    ***
*          *** NEXT DIRECTION.                                     ***

        IF (J.LE.3) GO TO 100

*             ************************************************
*             * BEGIN FIBONACCI INTERVAL SEARCH (ROSENBROCK) *
*             ************************************************

        LEFT  = J - J1
        RIGHT = J
        S1    = J1
        ZT1   = ZBEST
        JBEST = J1
        DO 87 I = 1, DIMNSN
           XT1(I) = XBEST(I)
 87     CONTINUE

*       *** RESET COUNTERS TO PREPARE FOR FIBONACCI INTERVAL SEARCH ***

        DO 88  N = 1, 3
           J1 = J - J1
           J  = J - J1
 88     CONTINUE

*       *** COMPUTE INNER RIGHT FIBONACCI POINT, S2 ***

 93     S2 = RIGHT - J

*       ****** COMPUTE COORDINATES OF POINT S2 ******
```

```
      DO 89  I = 1, DIMNSN
         XT2(I) = NINT(XB(I)+S2*DIR(I,L)/DEL1)
 89   CONTINUE

*       *** CHECK S2 FOR EXPLICIT FEASIBILITY AND COMPUTE ZT2 ***

      CALL ALAG1(XT2,ZT2)

*     *** DETERMINE WHICH INTERIOR POINT, S1 OR S2, IS BETTER ***
*     *** AND BEGIN LEFT FIBONACCI INTERVAL REDUCTION SECTION ***

 94   IF (ZT1.GT.ZT2) THEN
         JBEST = S2
         ZBEST = ZT2
         DO 90  I = 1, DIMNSN
            XBEST(I) = XT2(I)
 90      CONTINUE

*     *** IF J=1, INTERVAL HAS BEEN COMPLETELY REDUCED, ***
*     *** LOOP ON NEW DIRECTION, DIR(*,L) ***

         IF (J.LE.1) GO TO 100

*     *** OTHERWISE DO LEFT REDUCTION AND RETURN TO 93 ***

         J1   = J - J1
         J    = J - J1
         LEFT = S1
         S1   = S2
         ZT1  = ZT2
         DO 91  I = 1, DIMNSN
            XT1(I) = XT2(I)
 91      CONTINUE
         GO TO 93
      ENDIF

*     *** END OF LEFT INTERVAL REDUCTION SECTION ***
*     *** BEGIN RIGHT INTERVAL REDUCTION SECTION ***

      JBEST = S1
      ZBEST = ZT1
      DO 92  I = 1, DIMNSN
         XBEST(I) = XT1(I)
 92   CONTINUE

*     *** IF J=1, INTERVAL HAS BEEN COMPLETELY REDUCED, ***
*     *** LOOP ON NEW DIRECTION, DIR(I,L) ***

      IF (J.LE.1) GO TO 100

*     *** OTHERWISE DO RIGHT REDUCTION AND RETURN TO 94 ***

      J1    = J - J1
      J     = J - J1
      RIGHT = S2
      S2    = S1
      S1    = LEFT + J
      ZT2   = ZT1
      DO 95  I = 1, DIMNSN
         XT2(I) = XT1(I)
         XT1(I) = NINT(XB(I) + S1*DIR(I,L)/DEL1)
 95   CONTINUE

*       *** CHECK S1 FOR EXPLICIT FEASIBILITY AND COMPUTE ZT1 ***

      CALL ALAG1(XT1,ZT1)
      GO TO 94
```

```
*          *** END OF RIGHT INTERVAL REDUCTION SECTION ***

*          ***********************************************
*          * COMPUTE DCOMP(L) FOR THIS DIRECTION AND RESET XB *
*          ***********************************************

 100      DCOMP(L) = 0.0
          DO 101  I = 1, DIMNSN
             DCOMP(L) = DCOMP(L) + (XB(I) - XBEST(I))**2
             XB(I) = XBEST(I)
 101      CONTINUE

          DCOMP(L) = DCOMP(L)**0.5

*          *********************************
*          * END OF L-LOOP: RETURN TO DO 81 *
*          *********************************

 81    CONTINUE

*          *** RESET ITEMP AND COMPUTE IT ***

       ITEMP = 0
       DO 102  I = 1, DIMNSN
          XT(I) = IABS(X1(I)-XBEST(I))
          ITEMP = MAX0(ITEMP,XT(I))
 102   CONTINUE

*        *** CHECK IF XB = XBEST IF SO, RETURN TO DUNLOP ***

       IF (ITEMP.LT.1) THEN
          WRITE (6,900) 'FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FRO
      $M THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO T
      $O LOCMIN.'
          WRITE(6,969) XBEST, ZBEST
969       FORMAT(17X,6I3,2X,F12.0)
          RETURN
       ENDIF

*        *** IF THE STOPPING CONDITION ISN'T MET, GO BACK TO 150 ***

       GO TO 150

       END
       SUBROUTINE LOCMIN(FLAGM)

*           *********************************************
*           *                                           *
*           *   DETERMINES IF THE CURRENT XBEST (RESULTING   *
*           *   FROM SUBROUTINE INNER) IS A LOCAL MININUM.   *
*           *                                           *
*           *********************************************

*           ***** INTEGER VARIABLE DICTIONARY *****

*  COUNT1   NUMBER OF BINDING CONSTRAINTS.
*  FLAGC(6) RESULTS OF FEASIBILITY CHECK FOR EACH CONSTRAINT FOR THE
*           CURRENT SEARCH POINT.
*  FLAGE(12)RESULT OF EXPLICIT FEASIBILITY CHECK IN SUBROUTINE XPLST2.
*  FLAGE1   RESULT OF EXPLICIT FEASIBILITY CHECK, UPPER ADJACENT POINT.
*  FLAGE2   RESULT OF EXPLICIT FEASIBILITY CHECK, LOWER ADJACENT POINT.
*  FLAGM    RESULT OF LOCMIM: 0 = NOT A LOCAL MIN, 1 = LOCAL MINIMUM.
*  ICOL     IMSL VARIABLE CONTAINING THE NUMBER OF COLUMNS IN M MATRIX.
*  IERROR   IMSL SUBROUTINE ERROR RETURN CODE.
*  MDIM     IMSL VARIABLE CONTAINING ROW DIMENSION OF THE M MATRIX.
*  XT(6)    HOLDS COORDINATES OF CURRENT SEARCH POINT.
```

```
*  XT1(6)   TEMPORARILY HOLDS COORDINATES OF UPPER ADJACENT POINTS.
*  XT2(6)   TEMPORARITY HOLDS COORDINATES OF LOWER ADJACENT POINTS.

*           ****** REAL VARIABLE DICTIONARY ******

*  GRAD(6)  COMPONENTS OF THE NORMALIZED GRADIENT APPROXIMATION OF F(X).
*  GRADC(6) COMPONENTS OF THE NORMALIZED GRADIENT APPROXIMATION OF A(X).
*  NORMG    VALUE OF THE NORM OF GRAD(6).
*  NORMP    VALUE OF THE NORM OF PDIR(6).
*  M(6,6)   MATRIX OF THE GRADIENTS OF THE BINDING CONSTRAINTS AT XBEST.
*  MTEMP(6,6)  USED FOR INTERIM RESULTS OF GRADIENT PROJECTION.
*  PDIR(6)  NORMALIZED PROJECTED GRADIENT DIRECTION.
*  PROJ(6,6)   GRADIENT PROJECTION MATRIX.
*  ZT       VALUE OF OBJECTIVE FUNCTION AT POINT XT.
*  ZT1      VALUE OF OBJECTIVE FUNCTION AT POINT XT1.
*  ZT2      VALUE OF OBJECTIVE FUNCTION AT POINT XT2.

*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   COUNT1, FLAGC(6),
     $ FLAGE(12), FLAGE1, FLAGE2, FLAGM, ICOL, IDGT, IERROR, MDIM,
     $ XT(6), XT1(6), XT2(6)

*           ********* REAL DECLARATIONS *********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,
     $ DEL1, GRAD(6), GRADC(6), M(6,6), MTEMP(6,6), NORMG, NORMP,
     $ PDIR(6), PROJ(6,6), WKAREA(6), ZT, ZT1, ZT2

*           ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN

*           *** FORMAT STATEMENTS ***

 900  FORMAT(1X,A)

*           *** INITIALIZE THE GRADIENT NORM VARIABLE, ***
*           *** THE MAXIMUM MATRIX DIMENSION AND THE   ***
*           *** NUMBER OF DECISION VARIABLES IN MREAL1 ***

      NORMG = 0.0
      MDIM = 6
      ICOL = DIMNSN


*           *** CHECK XBEST FOR FEASIBILITY ***

      CALL CNSTR1(XBEST,FLAGC)

*           *** IF INFEASIBLE, RETURN TO DUNLOP ***

      IF (FLAG1.NE.0) THEN
         FLAGM = 0
         WRITE(6,900) 'FROM LOCMIN: RETURN 1'
         RETURN
      ENDIF

*           *** ELSE INITIALIZE XT1 AND XT2 ***
```

```
      DO 1  I = 1, DIMNSN
         XT1(I) = XBEST(I)
         XT2(I) = XBEST(I)
 1    CONTINUE

*           *** GET VALUE OF ZBEST FOR XBEST ***

      CALL OBJ(XBEST,ZBEST)

*           *** SET FIRST MAIN LOOP ***

      DO 2  I = 1, DIMNSN

*           *** SET UPPER AND LOWER ADJACENT POINTS ***

         XT1(I) = XBEST(I) + 1
         XT2(I) = XBEST(I) - 1

*           *** CHECK THESE POINTS FOR EXPLICIT FEASIBILITY ***

         CALL XPLST1(XT1,FLAGE1)
         CALL XPLST1(XT2,FLAGE2)

*           *** COMPUTE THE ACTUAL OBJECTIVE FUNCTION VALUES ***

         CALL OBJ(XT1,ZT1)
         CALL OBJ(XT2,ZT2)

*           *** CHECK TO SEE IF XT1 HAS A BETTER OBJECTIVE FUNCTION ***
*           *** VALUE.  IF SO, CHECK FEASIBILITY.  IF FEASIBLE THEN ***
*           *** SET NEW BASE POINT = XT1, FLAGM = 0 AND RETURN.     ***

         IF (ZT1.LT.ZBEST) THEN
            CALL CNSTR1(XT1,FLAGC)
            IF (FLAG1.EQ.0.AND.FLAGE1.EQ.0) THEN
               DO 5  J = 1, DIMNSN
                  XB(J) = XT1(J)
 5             CONTINUE
               WRITE(6,900) 'FROM LOCMIN: RETURN 2'
               FLAGM = 0
               RETURN
            ENDIF
         ENDIF

*           *** CHECK TO SEE IF XT2 HAS A BETTER OBJECTIVE FUNCTION ***
*           *** VALUE.  IF SO, CHECK FEASIBILITY.  IF FEASIBLE THEN ***
*           *** SET NEW BASE POINT = XT2, FLAGM = 0 AND RETURN.     ***

         IF (ZT2.LT.ZBEST) THEN
            CALL CNSTR1(XT2,FLAGC)
            IF (FLAG1.EQ.0.AND.FLAGE2.EQ.0) THEN
               DO 6  J = 1, DIMNSN
                  XB(J) = XT2(J)
 6             CONTINUE
               FLAGM = 0
               WRITE(6,900) 'FROM LOCMIN: RETURN 3'
               RETURN
            ENDIF
         ENDIF

*           *** AT THIS POINT NEITHER UPPER OR LOWER ADJACENT POINT ***
*           *** IS BETTER AND FEASIBLE.  PROCEED TO APPROXIMATE THE ***
*           *** GRADIENT OF F(XBEST).                               ***

         IF (ZT1.LT.ZT2) THEN
            IF (FLAGE1.NE.0) GRAD(I) = 0.0
            IF (FLAGE1.EQ.0) GRAD(I) = ZT2 - ZBEST
```

```
         ENDIF

         IF (ZT1.GE.ZT2) THEN
            IF (FLAGE2.NE.0) GRAD(I) = 0.0
            IF (FLAGE2.EQ.0) GRAD(I) = ZBEST - ZT1
         ENDIF

*           *** ADD IN COMPONENT TO NORMG ***

         NORMG = NORMG + GRAD(I)**2

*           *** RESET UPPER AND LOWER ADJACENT POINTS ***

         XT1(I) = XBEST(I)
         XT2(I) = XBEST(I)

*           *** AND REPEAT FOR REMAINING COORDINATE AXES ***

 2    CONTINUE

*           *** COMPLETE THE COMPUTATION OF THE NORM OF THE GRADIENT ***

      NORMG = NORMG**0.5

*           *** IF NORMG = 0.0 THEN ALL GRADIENT COMPONENTS WERE ***
*           *** SET TO ZERO BECAUSE OF EXPLICIT CONSTRAINTS.     ***
*           *** RETURN TO DUNLOP WITH FLAGM = 1, CONSTRAINED     ***
*           *** LOCAL MINIMUM FOUND.                             ***

      IF (NORMG.EQ.0.0) THEN
         WRITE(6,900) 'WE HAVE AN EXPLICITLY CONSTRAINED LOCAL MINIMUM -
     $ RETURN TO DUNLOP AND MAIN.'
         FLAGM = 1
         WRITE(6,900) 'FROM LOCMIN: RETURN 4'
         RETURN
      ENDIF

*           *** INITIALIZE DEL1 TO 0.0, NORMALIZE THE GRADIENT ***
*           *** F(X), AND COMPUTE DEL1                         ***

      DEL1 = 0.0
      DO 10  I = 1, DIMNSN
         GRAD(I) = GRAD(I)/NORMG
         IF (ABS(GRAD(I)).GT.DEL1) DEL1 = ABS(GRAD(I))
 10   CONTINUE

*           *** NOW WE ARE READY TO BEGIN SEARCHING IN THE  ***
*           *** DIRECTION OF STEEPEST DESCENT BY COMPUTING  ***
*           *** THE NEXT 5 INTEGER POINTS IN THIS DIRECTION ***

      DO 15  J = 1, DIMNSN + 3

*           *** COMPUTE THE NEXT INTEGER POINT IN DIRECTION GRAD ***

         DO 20  I = 1, DIMNSN
            XT(I) = NINT(XBEST(I) + J*GRAD(I)/DEL1)
 20      CONTINUE

*           *** FORCE EXPLICIT FEASIBILITY ***

         CALL XPLST2(XT,FLAGE1,FLAGE)

*           *** CHECK IMPLICIT FEASIBILITY ***

         CALL CNSTR1(XT,FLAGC)

*           *** WRITE THE COORDINATES OF THE SEARCH POINT ***
```

```
      WRITE(6,990) XT, FLAGC
990   FORMAT(1X,'FROM LOCMIN: NEXT GRAD SEARCH POINT:',6I3,'  FLAGC =
     $',6I1)

*          *** IF INFEASIBLE, GO TO THE PROJECTED GRADIENT METHOD ***

      IF (FLAG1.NE.0.OR.FLAGE1.NE.0) GO TO 50

*          *** COMPUTE OBJECTIVE FUNCTION VALUE FOR FEASIBLE XT ***

      CALL OBJ(XT,ZT)

*          *** IF BETTER THAN ZBEST, SET XB = XT, SET LOCAL ***
*          *** MINIMUM FLAG = 0 (NO), AND RETURN TO DUNLOP. ***

      IF (ZT.LT.ZBEST) THEN
         FLAGM = 0
         DO 22  K = 1, DIMNSN
            XB(K) = XT(K)
22       CONTINUE
         WRITE(6,900) 'FROM LOCMIN: RETURN 5'
         RETURN
      ENDIF

15    CONTINUE

*          *** NO BETTER, FEASIBLE POINT FOUND - RETURN WITH ***
*          *** XBEST AS LOCAL MINIMUM.                       ***

      FLAGM = 1
      WRITE(6,900) 'FROM LOCMIN: RETURN 6'
      RETURN

*          ***********************************
*          * BEGIN GRADIENT PROJECTION METHOD *
*          ***********************************

*          *** SET BINDING CONSTRAINT COUNTER TO ZERO ***

50    COUNT1 = 0

*          *** SET UP LOOP TO CHECK ALL CONSTRAINTS ***
*          *** TO SEE WHICH ARE BINDING             ***

      DO 52  J = 1, 6

*          *** IS CONSTRAINT J BINDING?  IF SO, COMPUTE ITS ***
*          *** GRADIENT AT XBEST AND PLACE ROW IN MATRIX M. ***

      IF (FLAGC(J).EQ.1) THEN
         COUNT1 = COUNT1 + 1
         CALL CGRAD(XBEST,J,GRADC)
         DO 54  L = 1, DIMNSN
            M(COUNT1,L) = GRADC(L)
54       CONTINUE
      ENDIF
52    CONTINUE

      WRITE(6,891) COUNT1
891   FORMAT(1X,'FROM LOCMIN: THERE ARE',I3,1X,'BINDING IMPLICIT CONSTRA
     $INTS IN M.')

*          *** CHECK IF ALL IMPLICIT CONSTRAINTS ARE SATISFIED ***
*          *** IF SO, ONLY EXPLICIT CONSTRAINTS ARE VIOLATED,  ***
*          *** SO WE ENTER A ROUTINE TO DEAL WITH THEM.        ***
```

```
      IF (FLAGE1.NE.0) THEN

*           *** SET UP LOOP TO CHECK THE IMPLICIT    ***
*           *** CONSTRAINTS TO SEE WHICH ARE BINDING ***

         DO 56  J = 1, 12

*           *** IS CONSTRAINT J BINDING?  IF SO, COMPUTE ITS ***
*           *** GRADIENT AT XBEST AND PLACE ROW IN MATRIX M. ***

            IF (FLAGE(J).EQ.1) THEN
               COUNT1 = COUNT1 + 1
               CALL ECGRAD(XBEST,J,GRADC)
               DO 58  L = 1, DIMNSN
                  M(COUNT1,L) = GRADC(L)
 58            CONTINUE
            ENDIF
 56      CONTINUE
      ENDIF

      WRITE(6,892) COUNT1
892   FORMAT(1X,'FROM LOCMIN: THERE ARE',I3,1X,'BINDING EXPLICIT AND IMP
     $LICIT CONSTRAINTS IN M.')

*           ************************************************
*           * START COMPUTATION OF PROJECTION MATRIX AND PDIR *
*           ************************************************

*           *** COMPUTE M * M-TRANSPOSE AND PLACE IN MTEMP ***

      CALL VMULFP(M,M,COUNT1,ICOL,COUNT1,MDIM,MDIM,MTEMP,MDIM,IERROR)

*           *** COMPUTE THE INVERSE OF M * M-TRANSPOSE ***

      IDGT = 0
      CALL LINV1F(MTEMP,COUNT1,MDIM,PROJ,IDGT,WKAREA,IERROR)

*           *** COMPUTE M-TRANSPOSE * PROJ ***

      CALL VMULFM(M,PROJ,COUNT1,ICOL,COUNT1,MDIM,MDIM,MTEMP,MDIM,IERROR)

*           *** COMPUTE MTEMP * M = PROJ ***

      CALL VMULFF(MTEMP,M,ICOL,COUNT1,ICOL,MDIM,MDIM,PROJ,MDIM,IERROR)

*           *** COMPUTE THE PROJECTION MATRIX = I - PROJ ***

      DO 60  I = 1, DIMNSN
         DO 62  J = 1, DIMNSN
            PROJ(I,J) = -PROJ(I,J)
 62      CONTINUE
 60   CONTINUE

      DO 65  I = 1, DIMNSN
         PROJ(I,I) = 1 + PROJ(I,I)
 65   CONTINUE

*           *** NOW COMPUTE THE PROJECTED GRADIENT DIRECTION, ***
*           *** PDIR AND NORMALIZE IT USING NORMP             ***

      NORMP = 0.0
      DO 68  I = 1, DIMNSN
         PDIR(I) = 0.0
         DO 69  J = 1, DIMNSN
            PDIR(I) = PDIR(I) + PROJ(I,J)*GRAD(J)
 69      CONTINUE
         NORMP = NORMP + PDIR(I)**2
```

```
 68   CONTINUE

*            *** FINISH THE NORMALIZATION ***

      NORMP = NORMP**0.5

*            *** CHECK TO SEE IF NORMP = 0.0.  IF SO, SET ***
*            *** FLAGM = 1 AND RETURN TO DUNLOP AND MAIN. ***

      IF (NORMP.EQ.0.0) THEN
         WRITE(6,900) 'FROM LOCMIN: NORMP = 0.0 - RETURN TO DUNLOP AND M
     $AIN.'
         FLAGM = 1
         WRITE(6,900) 'FROM LOCMIN: RETURN 7'
         RETURN
      ENDIF

*            *** OBTAIN DEL1 ***

      DEL1 = 0.0
      DO 70  I = 1, DIMNSN
         PDIR(I) = PDIR(I)/NORMP
         IF (ABS(PDIR(I)).GT.DEL1) DEL1 = ABS(PDIR(I))
 70   CONTINUE

*            *** COMPUTE THE NEXT DIMNSN + 3 INTEGER POINTS    ***
*            *** NEAREST DIRECTION PDIR AND CHECK FOR FEASI-   ***
*            *** BILITY AND IMPROVEMENT OF OBJECTIVE FUNCTION. ***

      DO 72  J = 1, DIMNSN + 3

*            *** COMPUTE THE NEXT INTEGER POINT IN DIRECTION ***
*            *** PDIR FROM POINT XBEST.                      ***

         DO 75  I = 1, DIMNSN
            XT(I) = NINT(XBEST(I) + J*PDIR(I)/DEL1)
 75      CONTINUE

*            *** COMPUTE FEASIBILITIES ***

         CALL XPLSIT(XT)
         CALL CONSTR(XT)

*            *** WRITE THE CURRENT SEARCH POINT ***

         WRITE(6,893) XT, FLAG1
 893     FORMAT(1X,'FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT: '
     $,6I3,'   FLAG1=',I2)

*            *** CHECK FOR FEASIBILITY.  IF FEASIBLE, COMPUTE F(XT) ***

         IF(FLAG1.NE.0) GO TO 72

*            *** COMPUTE F(XT) ***

         CALL OBJ(XT,ZT)

*            *** CHECK TO SEE IF F(XT) < F(XBEST).  IF SO, ***
*            *** RETURN WITHOUT LOCAL MINIMUM (FLAGM = 0). ***

         IF (ZT.LT.ZBEST) THEN
            FLAGM = 0
            DO 77 K = 1, DIMNSN
               XB(K) = XT(K)
 77         CONTINUE
            WRITE(6,900) 'FROM LOCMIN: RETURN 8'
            RETURN
```

```
         ENDIF
 72   CONTINUE
*           *** OTHERWISE, PRESUME A LOCAL MINIMUM HAS BEEN ***
*           *** FOUND AND RETURN WITH FLAGM = 1.            ***
      FLAGM = 1
      WRITE(6,900) 'FROM LOCMIN: RETURN 9'
      RETURN
      END
      SUBROUTINE OUTER
*           ****************************************************
*           *                                                  *
*           *      OUTER INTERATION WHICH UPDATES THE          *
*           *      PENALTY PARAMETERS, THE LAGRANGE            *
*           *      MULTIPLIERS, THE ITERATION COUNTER          *
*           *      AND THE OLD MAX CONSTRAINT VIOLATION.       *
*           *                                                  *
*           ****************************************************
*           ********* INTEGER DECLARATIONS *********
      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6)
*           ********* REAL DECLARATIONS *********
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2
*           ******** LABELED COMMON BLOCK ********
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
*           *** OBTAIN NEW MAX CONSTRAINT VIOLATION ***
      CALL CONSTR(XBEST)
*           *** IF THE OLD MAX CONSTRAINT VIOLATION IS    ***
*           *** LE THE NEW MAX CONSTRAINT VIOLATION THEN ***
*           *** UPDATE ONLY THE LAGRANGE MULTIPLIERS.     ***
      IF (MAXG1.LT.MAXG2) THEN
         DO 1  J = 1, 6
            LAMBDA(J) = LAMBDA(J) - AMIN1(LAMBDA(J), PRHO(J)*GEE(J))
 1       CONTINUE
      GO TO 10
      ENDIF
*           *** IF THE OLD MAX CONSTRAINT VIOLATION IS    ***
*           *** GE THE NEW MAX CONSTRAINT VIOLATION THEN ***
*           *** UPDATE THE PENALTY PARAMETERS FOR THE     ***
*           *** CONSTRAINTS WORSE THAN THE OLD MAX VIOL. ***
      DO 2  J = 1, 6
         IF (-GEE(J).GE.MAXG2) PRHO(J) = 10 * PRHO(J)
 2    CONTINUE
*           *** UPDATE THE ITERATION COUNTER AND SET THE ***
*           *** OLD MAX VIOLATION EQUAL TO THE NEW MAX.  ***
```

```
 10   IC = IC + 1
      MAXG2 = MAXG1

      RETURN
      END
      SUBROUTINE CONSTR (XF)

*              ****************************************
*              *                                      *
*              *    CONSTRAINT SUBROUTINE FOR LOCMIN  *
*              *              AND OUTER               *
*              *                                      *
*              ****************************************


*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   XF(6)

*           ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,
     $ ACUNA(2), AOPC, AP(40), APRC, ASUNA(2),   MAXG1T

*           ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2

      COMMON /OPT1/ APRC, AOPC, ACUNA, ASUNA, AP

*           **** DOUBLE PRECISION DECLARATION ****

      DOUBLE PRECISION NFACT

*        *** CHECK NON-NEGATIVITY AND VARIABLE MAXIMUM CONSTRAINTS ***

      CALL NORMAL (XF)

*           *** ANNUAL REPLACEMENT PROCUREMENT ***
*           ********* CAPITAL CONSTRAINT *********

      GEE(1) = BPR
      DO 1  I = 1, NREALS
            GEE(1) = GEE(1) - CPR(I,XF)
 1    CONTINUE

*           ****** ANNUAL OPERATING CAPITAL ******
*           ************ CONSTRAINT *************

      GEE(2) = BOP
      DO 2  I = 1, NREALS
            GEE(2) = GEE(2) - COP(I,XF)
 2    CONTINUE

*           *** CATASTROPHIC UNAVAILABILITY AND   ***
*           *** SYSTEM UNAVAILABILITY CONSTRAINTS ***

      DO 3 I = 1, NREALS

          ACUNA(I) = BCAT(I) - CUNA(I,XF)
          ASUNA(I) = BUNA(I) - SUNA(I,XF)
```

```
 3    CONTINUE

*           *** PUT SLACK VALUES IN GEE(I), I=3,...,6 ***

      GEE(3) = ASUNA(1)
      GEE(4) = ASUNA(2)
      GEE(5) = ACUNA(1)
      GEE(6) = ACUNA(2)

*           *** RESET LARGEST CONSTRAINT VIOLATION, MAXG1, ***
*           *** AND FEASIBILITY FLAG, FLAG1, TO ZERO       ***

      MAXG1 = 0
      FLAG1 = 0

*           *** FIND LARGEST CONSTRAINT VIOLATION ***

      DO 4  J = 1, 6
         MAXG1T = ABS(AMIN1(GEE(J),0.0))
         IF (MAXG1T.GT.MAXG1) THEN
            MAXG1 = MAXG1T
            FLAG1 = J
         ENDIF
 4    CONTINUE

      RETURN
      END
      SUBROUTINE CNSTR1 (XF,FLAGC)

*              *****************************************
*              *                                       *
*              *    CONSTRAINT SUBROUTINE FOR ALAG     *
*              *                                       *
*              *****************************************

*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   FLAGC(6), XF(6)

*           ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,
     $ ACUNA(2), AOPC, AP(40), APRC, ASUNA(2)

*           ******** LABELED COMMON BLOCKS ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2

      COMMON /OPT1/ APRC, AOPC, ACUNA, ASUNA, AP

      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN

*           **** DOUBLE PRECISION DECLARATION ****

      DOUBLE PRECISION NFACT

      FLAG1 = 0
      DO 1  J = 1, DIMNSN
         FLAGC(J) = 0
 1    CONTINUE
```

```
*       *** COMPUTE NORM1 VIA SUBROUTINE NORMAL(XF) ***

      CALL NORMAL (XF)

*           *** ANNUAL REPLACEMENT PROCUREMENT ***
*           ********* CAPITAL CONSTRAINT *********

      GEE(1) = BPR
      DO 2  I = 1, NREALS
            GEE(1) = GEE(1) - CPR(I,XF)
 2    CONTINUE

*           ****** ANNUAL OPERATING CAPITAL ******
*           ************ CONSTRAINT **************

      GEE(2) = BOP
      DO 3  I = 1, NREALS
            GEE(2) = GEE(2) - COP(I,XF)
 3    CONTINUE

*           *** CATASTROPHIC UNAVAILABILITY AND   ***
*           *** SYSTEM UNAVAILABILITY CONSTRAINTS ***

      DO 4 I = 1, NREALS

          ACUNA(I) = BCAT(I) - CUNA(I,XF)
          ASUNA(I) = BUNA(I) - SUNA(I,XF)

 4    CONTINUE

*           *** PUT SLACK VALUES IN GEE(I), I=3,...,6 ***

      GEE(3) = ASUNA(1)
      GEE(4) = ASUNA(2)
      GEE(5) = ACUNA(1)
      GEE(6) = ACUNA(2)

*           *** SET INDICATOR FLAG1 TO LAST VIOLATED CONSTRAINT ***
*           *** SET FLAGC(J) = 1 FOR EACH VIOLATED CONSTRAINT J ***


      DO 5  J = 1, 6
         IF (GEE(J).LT.0.0) THEN
            FLAG1 = J
            FLAGC(J) = 1
         ENDIF
 5    CONTINUE

      RETURN
      END


      FUNCTION CPR(I,XF)

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),    XF(6)
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2

          CPR = XF(3*I-2) * P(I,DESIGN(I))/XF(3*I)

      RETURN
      END
```

```
      FUNCTION COP(I,XF)

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),    XF(6)
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2

          COP = XF(3*I-2) * O(I, DESIGN(I)) + XF(3*I-1) * RO(I)

      RETURN
      END


      FUNCTION CUNA(I,XF)

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),    XF(6), RTEMP
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      DOUBLE PRECISION DCUNA, MDBLE, MTEMP, NFACT, NTEMP, TEMP1

      CALL XUFLOW(0)

      N = XF(3*I-2)
      M = XF(3*I-1)
      MDBLE = DBLE(XF(3*I-1))
      RTEMP = XF(3*I)
      NTEMP = NFACT(N)
      MTEMP = NFACT(M)
      DCUNA = 0.0

      DO 1 K = SH(I), D(I)

         IN = N-D(I)+K
         IF (IN.LE.M) TEMP1 = (NTEMP*RHO(I,DESIGN(I),RTEMP)**IN/
     $                          NFACT(N-IN))/NFACT(IN)
         IF (IN.GT.M) TEMP1 = ((NTEMP*RHO(I,DESIGN(I),RTEMP)**IN/
     $                          NFACT(N-IN))/MTEMP)/(MDBLE**(IN-M))
         DCUNA = DCUNA + TEMP1

 1    CONTINUE

*            *** SCALE AND NORMALIZE THE CONSTRAINT ***

      CUNA = 1.0E10*SNGL(DCUNA)/NORM1(I)

      CALL XUFLOW(1)

      RETURN
      END


      FUNCTION SUNA(I,XF)

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),    XF(6), RTEMP
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
```

```
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      DOUBLE PRECISION DSUNA, MDBLE, MTEMP, NFACT, NTEMP, TEMP1

      CALL XUFLOW(0)

      N = XF(3*I-2)
      M = XF(3*I-1)
      MDBLE = DBLE(XF(3*I-1))
      RTEMP = XF(3*I)
      NTEMP = NFACT(N)
      MTEMP = NFACT(M)
      DSUNA = 0.0

      DO 1 K = 1, D(I)

         IN = N-D(I)+K
         IF (IN.LE.M) TEMP1 = (NTEMP*(K*RHO(I,DESIGN(I),RTEMP)**IN)/
     $                         NFACT(N-IN))/NFACT(IN)
         IF (IN.GT.M) TEMP1 = ((NTEMP*(K*RHO(I,DESIGN(I),RTEMP)**IN)/
     $                          NFACT(N-IN))/MTEMP)/(MDBLE**(IN-M))
         DSUNA = DSUNA + TEMP1

 1    CONTINUE

*           *** SCALE AND NORMALIZE THE CONSTRAINT ***

      SUNA = 1.0E9*(SNGL(DSUNA)/NORM1(I))/D(I)

      CALL XUFLOW(1)

      RETURN
      END


      FUNCTION SUNA1(I,XF)

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),    XF(6), RTEMP
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      DOUBLE PRECISION DSUNA, MDBLE, MTEMP, NFACT, NTEMP, TEMP1

      CALL XUFLOW(0)

      N = XF(3*I-2)
      M = XF(3*I-1)
      MDBLE = DBLE(XF(3*I-1))
      RTEMP = XF(3*I)
      NTEMP = NFACT(N)
      MTEMP = NFACT(M)
      DSUNA = 0.0

      DO 1 K = 1, D(I)

         IN = N-D(I)+K
         IF (IN.LE.M) TEMP1 = (NTEMP*(K*RHO(I,DESIGN(I),RTEMP)**IN)/
     $                         NFACT(N-IN))/NFACT(IN)
         IF (IN.GT.M) TEMP1 = ((NTEMP*(K*RHO(I,DESIGN(I),RTEMP)**IN)/
     $                          NFACT(N-IN))/MTEMP)/(MDBLE**(IN-M))
```

```
         DSUNA = DSUNA + TEMP1
 1     CONTINUE
       SUNA1 = SNGL(DSUNA)/NORM1(I)
       CALL XUFLOW(1)
       RETURN
       END
       SUBROUTINE XPLSIT(XT)
*             ****************************************
*             *                                      *
*             *      CHECKS EXPLICIT CONSTRAINTS     *
*             *     AND RESETS XT(I) AS NECESSARY    *
*             *                                      *
*             ****************************************

*          ********* INTEGER DECLARATIONS *********

       INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
      $ SH(2), XB(6), XBEST(6), X1(6), X2(6),    XT(6)

*          ********* REAL DECLARATIONS **********

       REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
      $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
      $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2

*          ******** LABELED COMMON BLOCK ********

       COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
      $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
      $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2

*          *** FORCE DECISION VARIABLES ABOVE ***
*          *** MINIMUM VALUES (DEMAND AND 1) ****

       XT(1) = MAX0(XT(1),D(1))
       XT(2) = MAX0(XT(2),1)
       XT(3) = MAX0(XT(3),1)
       XT(4) = MAX0(XT(4),D(2))
       XT(5) = MAX0(XT(5),1)
       XT(6) = MAX0(XT(6),1)

*          *** FORCE DECISION VARIABLES BELOW ***
*          *** MAXIMUM VALUES AS APPROPRIATE ****

       XT(1) = MIN0(XT(1),56)
       XT(4) = MIN0(XT(4),56)
       XT(2) = MIN0(XT(2),XT(1))
       XT(5) = MIN0(XT(5),XT(4))
       XT(3) = MIN0(XT(3),RMAX(1,DESIGN(1)))
       XT(6) = MIN0(XT(6),RMAX(2,DESIGN(2)))

       RETURN
       END
       SUBROUTINE XPLST1(XT,FLAG)

*             ****************************************
*             *                                      *
*             *      CHECKS EXPLICIT CONSTRAINTS,    *
*             *      RESETS XT(I) AS NECESSARY AND   *
*             *      SETS INDICATOR FLAG = 0 IF FEAS. *
```

```
*              *                                        *
*              ****************************************

*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),    FLAG, XT(6), XTEMP(6)

*           ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2

*           ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN

      FLAG = 0

*           *** RECORD XT IN A TEMPORARY VARIABLE ***

      DO 1  I = 1, DIMNSN
         XTEMP(I) = XT(I)
 1    CONTINUE

*           *** FORCE DECISION VARIABLES ABOVE ***
*           *** MINIMUM VALUES (DEMAND AND 1) ****

      XT(1) = MAX0(XT(1),D(1))
      XT(2) = MAX0(XT(2),1)
      XT(3) = MAX0(XT(3),1)
      XT(4) = MAX0(XT(4),D(2))
      XT(5) = MAX0(XT(5),1)
      XT(6) = MAX0(XT(6),1)

*           *** FORCE DECISION VARIABLES BELOW ***
*           *** MAXIMUM VALUES AS APPROPRIATE ****

      XT(1) = MIN0(XT(1),56)
      XT(2) = MIN0(XT(2),XT(1))
      XT(3) = MIN0(XT(3),RMAX(1,DESIGN(1)))
      XT(4) = MIN0(XT(4),56)
      XT(5) = MIN0(XT(5),XT(4))
      XT(6) = MIN0(XT(6),RMAX(2,DESIGN(2)))

*           *** CHECK FOR INFEASIBILITY AND SET FLAG ***

      DO 4  I = 1, DIMNSN
         IF (XTEMP(I).NE.XT(I)) FLAG = 1
 4    CONTINUE

      RETURN
      END
      SUBROUTINE XPLST2(XT,FLAG,FLAGE)

*              ****************************************
*              *                                      *
*              *      CHECKS EXPLICIT CONSTRAINTS,    *
*              *      SETS FLAGE(I) = 1 IF CON-       *
*              *      STRAINT I IS VIOLATED.          *
```

```
*             *                                       *
*             *****************************************

*          ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),    FLAG, FLAGE(12),
     $ XT(6), XTEMP(6)

*          ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2

*          ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2

      FLAG = 0
      DO 1  I = 1, 12
         FLAGE(I) = 0
 1    CONTINUE

*          *** CHECK FOR CONSTRAINT VIOLATIONS ***
*          *** AND SET FLAGC(I) ACCORDINGLY.   ***

      IF (XT(1).LT.D(1))                 FLAGE(1) = 1
      IF (XT(2).LT.1)                    FLAGE(2) = 1
      IF (XT(3).LT.1)                    FLAGE(3) = 1
      IF (XT(4).LT.D(2))                 FLAGE(4) = 1
      IF (XT(5).LT.1)                    FLAGE(5) = 1
      IF (XT(6).LT.1)                    FLAGE(6) = 1


      IF (XT(1).GT.56)                   FLAGE(7) = 1
      IF (XT(2).GT.XT(1))                FLAGE(8) = 1
      IF (XT(3).GT.RMAX(1,DESIGN(1)))    FLAGE(9) = 1
      IF (XT(4).GT.56)                   FLAGE(10) = 1
      IF (XT(5).GT.XT(4))                FLAGE(11) = 1
      IF (XT(6).GT.RMAX(2,DESIGN(2)))    FLAGE(12) = 1

*          *** SET OTHER FLAG ***

      DO 2  I = 1, 12
         IF (FLAGE(I).EQ.1) FLAG = 1
 2    CONTINUE

*          *** FORCE DECISION VARIABLES ABOVE ***
*          *** MINIMUM VALUES (DEMAND AND 1) ****

      XT(1) = MAX0(XT(1),D(1))
      XT(2) = MAX0(XT(2),1)
      XT(3) = MAX0(XT(3),1)
      XT(4) = MAX0(XT(4),D(2))
      XT(5) = MAX0(XT(5),1)
      XT(6) = MAX0(XT(6),1)

*          *** FORCE DECISION VARIABLES BELOW ***
*          *** MAXIMUM VALUES AS APPROPRIATE ****

      XT(1) = MIN0(XT(1),56)
      XT(2) = MIN0(XT(2),XT(1))
      XT(3) = MIN0(XT(3),RMAX(1,DESIGN(1)))
```

```
      XT(4) = MINO(XT(4),56)
      XT(5) = MINO(XT(5),XT(4))
      XT(6) = MINO(XT(6),RMAX(2,DESIGN(2)))

      RETURN
      END
      SUBROUTINE XPLST3(XT,ZT)

*               ****************************************
*               *                                      *
*               *       CHECKS EXPLICIT CONSTRAINTS    *
*               *      AND RESETS XT(I) AS NECESSARY.  *
*               *      COMPUTES PENALTY VALUE FOR VIO- *
*               *      LATIONS AND PLACES VALUE IN ZT. *
*               *      CALLED FROM ALAG1.              *
*               *                                      *
*               ****************************************


*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),    XT(6)

*           ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,    ZT

*           ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2

*           *** INITIALIZE PENALTY VALUE IN ZT TO ZERO ***

      ZT = 0.0

*           *** COMPUTE PENALTY VALUES FOR LOWER VIOLATIONS ***

      ZT = ZT + 1.0E6*REAL(MINO(XT(1)-D(1),0))**2
      ZT = ZT + 1.0E6*REAL(MINO(XT(2)-1,0))**2
      ZT = ZT + 1.0E6*REAL(MINO(XT(3)-1,0))**2
      ZT = ZT + 1.0E6*REAL(MINO(XT(4)-D(2),0))**2
      ZT = ZT + 1.0E6*REAL(MINO(XT(5)-1,0))**2
      ZT = ZT + 1.0E6*REAL(MINO(XT(6)-1,0))**2

*           *** COMPUTE PENALTY VALUES FOR UPPER VIOLATIONS ***

      ZT = ZT + 1.0E6*REAL(MINO(56-XT(1),0))**2
      ZT = ZT + 1.0E6*REAL(MINO(XT(1)-XT(2),0))**2
      ZT = ZT + 1.0E6*REAL(MINO(RMAX(1,DESIGN(1))-XT(3),0))**2
      ZT = ZT + 1.0E6*REAL(MINO(56-XT(4),0))**2
      ZT = ZT + 1.0E6*REAL(MINO(XT(4)-XT(5),0))**2
      ZT = ZT + 1.0E6*REAL(MINO(RMAX(2,DESIGN(2))-XT(6),0))**2

*           *** FORCE DECISION VARIABLES ABOVE ***
*           *** MINIMUM VALUES (DEMAND AND 1) ****

      XT(1) = MAXO(XT(1),D(1))
      XT(2) = MAXO(XT(2),1)
      XT(3) = MAXO(XT(3),1)
      XT(4) = MAXO(XT(4),D(2))
      XT(5) = MAXO(XT(5),1)
      XT(6) = MAXO(XT(6),1)
```

```
*          *** FORCE DECISION VARIABLES BELOW ***
*          *** MAXIMUM VALUES AS APPROPRIATE ****

      XT(1) = MIN0(XT(1),56)
      XT(4) = MIN0(XT(4),56)
      XT(2) = MIN0(XT(2),XT(1))
      XT(5) = MIN0(XT(5),XT(4))
      XT(3) = MIN0(XT(3),RMAX(1,DESIGN(1)))
      XT(6) = MIN0(XT(6),RMAX(2,DESIGN(2)))

      RETURN
      END
      SUBROUTINE ALAG(XT,ZT)

*                 *************************************
*                 *                                   *
*                 *    COMPUTES THE VALUE OF THE AUG-  *
*                 *    MENTED LAGRANGIAN PENALTY FUNC- *
*                 *    TION FOR THE UNCONSTRAINED OP-  *
*                 *    TIMIZATION.                     *
*                 *                                   *
*                 *************************************

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   FLAGC(6), XT(6)
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,   ACUNA(2),
     $ AP(40), APRC, AOPC, ASUNA(2),BMU, RT, ZT
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      COMMON /OPT1/ APRC, AOPC, ACUNA, ASUNA, AP


      ZT = 0.0

*         *** COMPUTE THE GEE(I) FOR EACH CONSTRAINT ***

      CALL CNSTR1(XT,FLAGC)

*         *** COMPUTE THE LAGRANGIAN & PENALTY PORTION OF ALAG ***

      DO 1  J = 1, 6
         ZT = ZT + PRHO(J)*AMIN1(GEE(J)-LAMBDA(J)/PRHO(J),0.0)**2
 1    CONTINUE

      ZT = ZT/2


*         *** COMPUTE NORM1 USING SUBROUTINE NORMAL ***

      CALL NORMAL(XT)

*         *** COMPUTE THE TRUE OBJECTIVE FUNCTION PORTION OF ALAG ***

      DO 3 I = 1, NREALS

         RT = REAL(XT(3*I))
         FT = P(I,DESIGN(I))*(1.0-RT/RMAX(I,DESIGN(I)))
         CP = XT(3*I-2)*(AP(RT)*(P(I,DESIGN(I))-FT)+INT*FT+O(I,DESIGN(I)
     $ ))
         CF = XT(3*I-1)*(AP(RR(I))*(RP(I)-RF(I))+INT*RF(I)+RO(I))
         CS = C(I)*SUNA1(I,XT)
```

```
         ZT = ZT + CP + CF + CS

 3    CONTINUE

*        *** WRITE THE RESULTS ON ONE LINE USING THE TITLE FORMAT ***

      WRITE (6,905) DESIGN, XT, ZT, FLAGC, GEE

 905  FORMAT (4X,I1,7X,I1,4X,6I3,2X,E12.7,1X,6I1,1X,2(F10.0,1X),1X,4(E12
     $.6,1X))
      RETURN
      END

      SUBROUTINE ALAG1(XT,ZT)

*                   *************************************
*                   *                                   *
*                   *   COMPUTES THE VALUE OF THE AUG-  *
*                   *   MENTED LAGRANGIAN PENALTY FUNC- *
*                   *   TION FOR THE UNCONSTRAINED OP-  *
*                   *   TIMIZATION. CHECKS EXPLICIT     *
*                   *   CONSTRAINTS VIA XPLST3.         *
*                   *                                   *
*                   *************************************


      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   FLAGC(6), XT(6)
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,   ACUNA(2),
     $ AP(40), APRC, AOPC, ASUNA(2),BMU, RT, ZT
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      COMMON /OPT1/ APRC, AOPC, ACUNA, ASUNA, AP


      ZT = 0.0

*         *** CHECK FOR EXPLICIT FEASIBILITY.  COMPUTE A ***
*         *** PENALTY VALUE FOR VIOLATIONS AND RESET THE ***
*         *** POINT TO THE NEAREST FEASIBLE POINT.       ***

      CALL XPLST3(XT,ZT)

*         *** COMPUTE THE GEE(I) (SLACK OR VIOLATION) ***
*         *** FOR EACH CONSTRAINT.                    ***

      CALL CNSTR1(XT,FLAGC)

*         *** COMPUTE THE LAGRANGIAN & PENALTY PORTION OF ALAG ***

      DO 1  J = 1, 6
         ZT = ZT + PRHO(J)*AMIN1(GEE(J)-LAMBDA(J)/PRHO(J),0.0)**2
 1    CONTINUE

      ZT = ZT/2


*         *** COMPUTE NORM1 USING SUBROUTINE NORMAL ***

      CALL NORMAL(XT)

*         *** COMPUTE THE TRUE OBJECTIVE FUNCTION PORTION OF ALAG ***

      DO 3 I = 1, NREALS
```

```
         RT = REAL(XT(3*I))
         FT = P(I,DESIGN(I))*(1.0-RT/RMAX(I,DESIGN(I)))
         CP = XT(3*I-2)*(AP(RT)*(P(I,DESIGN(I))-FT)+INT*FT+O(I,DESIGN(I)
     $ ))
         CF = XT(3*I-1)*(AP(RR(I))*(RP(I)-RF(I))+INT*RF(I)+RO(I))
         CS = C(I)*SUNA1(I,XT)

         ZT = ZT + CP + CF + CS

 3    CONTINUE

*        *** WRITE THE RESULTS ON ONE LINE USING THE TITLE FORMAT ***

      WRITE (6,905) DESIGN, XT, ZT, FLAGC, GEE

 905  FORMAT (4X,I1,7X,I1,4X,6I3,2X,E12.7,1X,6I1,1X,2(F10.0,1X),1X,4(E12
     $.6,1X))
      RETURN
      END
      BLOCK DATA APE
*     **** CAPITAL RECOVERY FACTOR, AP, AT 10%, YEARS 1 - 40 ****

      REAL APRC, AOPC, AP(40), ACUNA(2), ASUNA(2)
      COMMON /OPT1/ APRC, AOPC, ACUNA, ASUNA, AP
      DATA AP / 1.1, 0.576190, 0.402115, 0.315471, 0.263797, 0.229607,
     $     0.205405, 0.187444, 0.173641, 0.162745, 0.153963, 0.146763,
     $     0.140779, 0.135746, 0.131474, 0.127817, 0.124664, 0.121930,
     $     0.119547, 0.117460, 0.115624, 0.114005, 0.112572, 0.111300,
     $     0.110168, 0.109159, 0.108258, 0.107451, 0.106728, 0.106079,
     $     0.105496, 0.104972, 0.104499, 0.104074, 0.103690, 0.103343,
     $     0.103030, 0.102747, 0.102491, 0.102259/
      END
      SUBROUTINE OBJ(XT,ZT)

*                  *************************************
*                  *                                   *
*                  *    COMPUTES THE VALUE OF THE OBJ- *
*                  *    ECTIVE FUNCTION FOR THE LOCAL  *
*                  *    MINIMUM SUBROUTINE (LOCMIN).   *
*                  *                                   *
*                  *************************************


      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   XT(6)
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,    ACUNA(2),
     $ AP(40), APRC, AOPC, ASUNA(2),BMU, RT, ZT
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      COMMON /OPT1/ APRC, AOPC, ACUNA, ASUNA, AP

      ZT = 0.0

*         *** COMPUTE NORM1 USING SUBROUTINE NORMAL ***

      CALL NORMAL(XT)

*         *** COMPUTE THE TRUE OBJECTIVE FUNCTION PORTION OF ALAG ***

      DO 3 I = 1, NREALS

         RT = REAL(XT(3*I))
         FT = P(I,DESIGN(I))*(1-RT/RMAX(I,DESIGN(I)))
```

```
         CP = XT(3*I-2)*(AP(RT)*(P(I,DESIGN(I))-FT)+INT*FT+O(I,DESIGN(I)
     $ ))
         CF = XT(3*I-1)*(AP(RR(I))*(RP(I)-RF(I))+INT*RF(I)+RO(I))
         CS = C(I)*SUNA1(I,XT)

         ZT = ZT + CP + CF + CS

 3    CONTINUE
      RETURN
      END
      SUBROUTINE NORMAL (XT)

*              *****************************************
*              *                                       *
*              *     COMPUTES FINITE SOURCE QUEUE      *
*              *         NORMALIZATION FACTOR          *
*              *                                       *
*              *****************************************

*           ********* INTEGER DECLARATIONS *********

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),   XT(6)

*           ********* REAL DECLARATIONS **********

      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2

*           ******** LABELED COMMON BLOCK ********

      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2

*           ***** DOUBLE PRECISION DECLARATION *****

      DOUBLE PRECISION DNORM1(2), MDBLE, MTEMP, NFACT, NTEMP, TEMP1

*           *** CALL IBM SUBROUTINE TO SUPPRESS PRINTING OF ***
*           *** RUN-TIME ERROR IFY208I - FLOATING POINT     ***
*           *** UNDERFLOW EXCEPTION.                        ***

      CALL XUFLOW(0)

*           *** BEGIN COMPUTATION OF NORMALIZATION FACTOR ***

      DO 1 I = 1, NREALS

         DNORM1(I) = 0.0
         NTEMP = NFACT(XT(3*I-2))
         MTEMP = NFACT(XT(3*I-1))
         MDBLE = DBLE(XT(3*I-1))

         DO 2 K = 0, XT(3*I-1)

            TEMP1 = (((RHO(I,DESIGN(I),XT(3*I))**K)*NTEMP)/NFACT(XT(3*I-
     $              2)-K))/NFACT(K)
            DNORM1(I) = DNORM1(I) + TEMP1

 2       CONTINUE

         DO 3 K = XT(3*I-1)+1, XT(3*I-2)

            TEMP1 = ((((RHO(I,DESIGN(I),XT(3*I))**K)*NTEMP)/NFACT(XT(3*I
     $              -2)-K))/MTEMP)/MDBLE**(K-XT(3*I-1))
```

```
            DNORM1(I) = DNORM1(I) + TEMP1

 3       CONTINUE

         NORM1(I) = SNGL(DNORM1(I))

 1     CONTINUE

*           *** TURN FLOATING POINT UNDERFLOW EXCEPTION ***
*           *** ERROR BACK ON.                          ***

       CALL XUFLOW(1)

       RETURN
       END

       DOUBLE PRECISION FUNCTION NFACT(N)
       DOUBLE PRECISION NRAY(56)
       DATA NRAY(1)   / 1.0D+00/
       DATA NRAY(2)   / 2.0D+00/
       DATA NRAY(3)   / 6.0D+00/
       DATA NRAY(4)   / 2.4D+01/
       DATA NRAY(5)   / 1.2D+02/
       DATA NRAY(6)   / 7.2D+02/
       DATA NRAY(7)   / 5.04D+03/
       DATA NRAY(8)   / 4.032D+04/
       DATA NRAY(9)   / 3.6288D+05/
       DATA NRAY(10)  / 3.6288D+06/
       DATA NRAY(11)  / 3.99168D+07/
       DATA NRAY(12)  / 4.790016D+08/
       DATA NRAY(13)  / 6.2270208D+09/
       DATA NRAY(14)  / 8.71782912D+10/
       DATA NRAY(15)  / 1.307674368D+12/
       DATA NRAY(16)  / 2.0922789888D+13/
       DATA NRAY(17)  / 3.55687428096D+14/
       DATA NRAY(18)  / 6.402373705728D+15/
       DATA NRAY(19)  / 1.2164510040883D+17/
       DATA NRAY(20)  / 2.4329020081766D+18/
       DATA NRAY(21)  / 5.1090942171709D+19/
       DATA NRAY(22)  / 1.1240007277776D+21/
       DATA NRAY(23)  / 2.5852016738885D+22/
       DATA NRAY(24)  / 6.2044840173324D+23/
       DATA NRAY(25)  / 1.5511210043331D+25/
       DATA NRAY(26)  / 4.0329146112661D+26/
       DATA NRAY(27)  / 1.0888869450418D+28/
       DATA NRAY(28)  / 3.0488834461171D+29/
       DATA NRAY(29)  / 8.8417619937397D+30/
       DATA NRAY(30)  / 2.6525285981219D+32/
       DATA NRAY(31)  / 8.2228386541779D+33/
       DATA NRAY(32)  / 2.6313083693369D+35/
       DATA NRAY(33)  / 8.6833176188119D+36/
       DATA NRAY(34)  / 2.9523279903960D+38/
       DATA NRAY(35)  / 1.0333147966386D+40/
       DATA NRAY(36)  / 3.7199332678990D+41/
       DATA NRAY(37)  / 1.3763753091226D+43/
       DATA NRAY(38)  / 5.2302261746660D+44/
       DATA NRAY(39)  / 2.0397882081197D+46/
       DATA NRAY(40)  / 8.1591528324790D+47/
       DATA NRAY(41)  / 3.3452526613164D+49/
       DATA NRAY(42)  / 1.4050061177529D+51/
       DATA NRAY(43)  / 6.0415263063374D+52/
       DATA NRAY(44)  / 2.6582715747884D+54/
       DATA NRAY(45)  / 1.1962222086548D+56/
       DATA NRAY(46)  / 5.5026221598121D+57/
       DATA NRAY(47)  / 2.5862324151117D+59/
       DATA NRAY(48)  / 1.2413915592536D+61/
       DATA NRAY(49)  / 6.0828186403427D+62/
```

```
      DATA NRAY(50)  / 3.0414093201713D+64/
      DATA NRAY(51)  / 1.5511187532874D+66/
      DATA NRAY(52)  / 8.0658175170944D+67/
      DATA NRAY(53)  / 4.2748832840600D+69/
      DATA NRAY(54)  / 2.3084369733924D+71/
      DATA NRAY(55)  / 1.2696403353658D+73/
      DATA NRAY(56)  / 7.1099858780486D+74/
C
      IF (N.GT.56.OR.N.LT.0) GOTO 199
      IF (N.EQ.0) THEN
          NFACT = 1.000D+00
          RETURN
          ENDIF
C
      NFACT = NRAY(N)
      RETURN
C
 199  WRITE(6,299) N
 299  FORMAT('  ERROR IN FUNCTION NFACT: N =',I4)
      RETURN
      END
      SUBROUTINE CGRAD(XT,FLAG,GRADC)

*            *********************************************
*            *                                           *
*            *  THIS SUBROUTINE COMPUTES THE GRADIENT OF *
*            *  THE EXPLICIT CONSTRAINTS AS NECESSARY.   *
*            *                                           *
*            *********************************************

*            ******* REAL VARIABLE DICTIONARY *******

*  GRADC(6)  GRADIENT OF CONSTRAINT NUMBER FLAG

      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),     FLAG, XT(6), XT1(6)
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,
     $ GRADC(6), NORMC
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN

*            *** INITIALIZE GRADC TO ZERO ***

      DO 30  I = 1, DIMNSN
         GRADC(I) = 0.0
 30   CONTINUE

*            *** INITIALIZE XT1 TO XT ***

      DO 32  I = 1, DIMNSN
         XT1(I) = XT(I)
 32   CONTINUE

*           *** GO TO THE APPROPRIATE CONSTRAINT ***

      GO TO (1,2,3,4,5,6) FLAG
      RETURN

*           *** SET GRADIENT OF CONSTRAINT NO. 1 USING THE ***
*           *** ALGEBRAICALLY COMPUTED GRADIENT FUNCTION.  ***

 1    GRADC(1) = P(1,DESIGN(1))/XT(3)
```

```
      GRADC(3) = -P(1,DESIGN(1))*XT(1)/XT(3)**2
      GRADC(4) = P(2,DESIGN(2))/XT(6)
      GRADC(6) = -P(2,DESIGN(2))*XT(4)/XT(6)**2
      GO TO 100

*         *** SET GRADIENT OF CONSTRAINT NO. 2 USING THE ***
*         *** ALGEBRAICALLY COMPUTED GRADIENT FUNCTION.  ***

 2    GRADC(1) = O(1,DESIGN(1))
      GRADC(2) = RO(1)
      GRADC(4) = O(2,DESIGN(2))
      GRADC(5) = RO(2)
      GO TO 100

*         *** SET GRADIENT OF CONSTRAINT NO. 3 USING ***
*         *** APPROXIMATELY COMPUTED GRADIENT VALUES ***

*         *** SET UP LOOP FOR EACH DECISION VARIABLE ***

 3    DO 40  I = 1, DIMNSN

*         *** MAKE XT1 A POINT ADJACENT TO XT ALONG AXIS I ***

         XT1(I) = XT(I) + 1

*         *** CHECK THIS POINT FOR EXPLICIT FEASIBILITY ***

         CALL XPLSIT(XT1)

*         *** IF FEASIBLE, IT WILL RETURN UNCHANGED ***
*         *** OTHERWISE USE POINT ADJACENT TO XT IN ***
*         *** THE OPPOSITE DIRECTION.               ***

         IF (XT1(I).EQ.XT(I)) THEN
            XT1(I) = XT(I) - 1

*          *** COMPUTE THE GRADIENT COMPONENT FOR THIS ***
*          *** DECISION VARIABLE (COORDINATE AXIS).    ***

            GRADC(I) = SUNA(1,XT) - SUNA(1,XT1)

*          *** RESET XT1 TO XT ***

            XT1(I) = XT(I)
            GO TO 40
         ENDIF

*          *** COMPUTE THE GRADIENT COMPONENT FOR THIS ***
*          *** DECISION VARIABLE (COORDINATE AXIS).    ***

         GRADC(I) = SUNA(1,XT1) - SUNA(1,XT)

*          *** RESET XT1 TO XT ***

         XT1(I) = XT(I)
 40   CONTINUE

      GO TO 100

*         *** SET GRADIENT OF CONSTRAINT NO. 4 USING ***
*         *** APPROXIMATELY COMPUTED GRADIENT VALUES ***

 4    DO 42  I = 1, DIMNSN
         XT1(I) = XT(I) + 1
         CALL XPLSIT(XT1)
         IF (XT1(I).EQ.XT(I)) THEN
            XT1(I) = XT(I) - 1
```

```
            GRADC(I) = SUNA(2,XT) - SUNA(2,XT1)
            XT1(I) = XT(I)
            GO TO 42
         ENDIF
         GRADC(I) = SUNA(2,XT1) - SUNA(2,XT)
         XT1(I) = XT(I)
 42   CONTINUE

      GO TO 100

*           *** SET GRADIENT OF CONSTRAINT NO. 5 USING ***
*           *** APPROXIMATELY COMPUTED GRADIENT VALUES ***

 5    DO 44  I = 1, DIMNSN
         XT1(I) = XT(I) + 1
         CALL XPLSIT(XT1)
         IF (XT1(I).EQ.XT(I)) THEN
            XT1(I) = XT(I) - 1
            GRADC(I) = CUNA(1,XT) - CUNA(1,XT1)
            XT1(I) = XT(I)
            GO TO 44
         ENDIF
         GRADC(I) = CUNA(1,XT1) - CUNA(1,XT)
         XT1(I) = XT(I)
 44   CONTINUE

      GO TO 100

*           *** SET GRADIENT OF CONSTRAINT NO. 6 USING ***
*           *** APPROXIMATELY COMPUTED GRADIENT VALUES ***

 6    DO 46  I = 1, DIMNSN
         XT1(I) = XT(I) + 1
         CALL XPLSIT(XT1)
         IF (XT1(I).EQ.XT(I)) THEN
            XT1(I) = XT(I) - 1
            GRADC(I) = CUNA(2,XT) - CUNA(2,XT1)
            XT1(I) = XT(I)
            GO TO 46
         ENDIF
         GRADC(I) = CUNA(2,XT1) - CUNA(2,XT)
         XT1(I) = XT(I)
 46   CONTINUE


*           *** NORMALIZE GRADC ***

 100  NORMC = 0.0
      DO 35  I = 1, DIMNSN
         NORMC = NORMC + GRADC(I)**2
 35   CONTINUE
      NORMC = NORMC**0.5
      DO 36  I = 1, DIMNSN
         GRADC(I) = GRADC(I)/NORMC
 36   CONTINUE

      RETURN
      END
      SUBROUTINE ECGRAD(XT,FLAG,GRADC)

*           *********************************************
*           *                                           *
*           *  THIS SUBROUTINE COMPUTES THE GRADIENT OF *
*           *  THE EXPLICIT CONSTRAINTS AS NECESSARY.   *
*           *                                           *
*           *********************************************
```

```
      INTEGER D(2), DESIGN(2), FLAG1, IC, NREALS, RMAX(2,2), S(2),
     $ SH(2), XB(6), XBEST(6), X1(6), X2(6),     FLAG, XT(6)
      REAL BCAT(2), BOP, BPR, BUNA(2), C(2), GEE(6), INT, LAMBDA(6),
     $ MAXG1, MAXG2, NORM1(2), O(2,2), P(2,2), PRHO(6), RF(2),
     $ RHO(2,2,40), RO(2), RP(2), RR(2), ZB, ZBEST, Z1, Z2,
     $ GRADC(6)
      COMMON /NLP/ BCAT, BOP, BPR, BUNA, C, D, DESIGN, FLAG1, GEE, IC,
     $ INT, LAMBDA, MAXG1, MAXG2, NORM1, NREALS, O, P, PRHO, RF, RHO,
     $ RMAX, RO, RP, RR, S, SH, XB, XBEST, X1, X2, ZB, ZBEST, Z1, Z2
      INTEGER DIMNSN
      COMMON /OTHER/ DIMNSN

*           *** INITIALIZE GRADC TO ZERO ***

      DO 30  I = 1, DIMNSN
         GRADC(I) = 0.0
 30   CONTINUE

*          *** GO TO THE APPROPRIATE CONSTRAINT ***

      GO TO (1,2,3,4,5,6,7,8,9,10,11,12) FLAG
      RETURN

*          *** SET GRADIENT OF CONSTRAINT NO. 1 ***

  1   GRADC(1) = -1.0
      RETURN

*          *** SET GRADIENT OF CONSTRAINT NO. 2 ***

  2   GRADC(2) = -1.0
      RETURN

*          *** SET GRADIENT OF CONSTRAINT NO. 3 ***

  3   GRADC(3) = -1.0
      RETURN

*          *** SET GRADIENT OF CONSTRAINT NO. 4 ***

  4   GRADC(4) = -1.0
      RETURN

*          *** SET GRADIENT OF CONSTRAINT NO. 5 ***

  5   GRADC(5) = -1.0
      RETURN

*          *** SET GRADIENT OF CONSTRAINT NO. 6 ***

  6   GRADC(6) = -1.0
      RETURN

*          *** SET GRADIENT OF CONSTRAINT NO. 7 ***

  7   GRADC(1) = 1.0
      RETURN

*          *** SET GRADIENT OF CONSTRAINT NO. 8 ***

  8   GRADC(1) = -1.0
      GRADC(2) = 1.0
      RETURN

*          *** SET GRADIENT OF CONSTRAINT NO. 9 ***

  9   GRADC(3) = 1.0
```

```
      RETURN
*           *** SET GRADIENT OF CONSTRAINT NO. 10 ***
 10   GRADC(4) = 1.0
      RETURN
*           *** SET GRADIENT OF CONSTRAINT NO. 11 ***
 11   GRADC(5) = -1.0
      GRADC(6) = 1.0
      RETURN
*           *** SET GRADIENT OF CONSTRAINT NO. 12 ***
 12   GRADC(6) = 1.0
      RETURN
      END
```

Appendix D

SAMPLE ENUMERATION OUTPUT

```
******************************************************************
*                                                                *
*                     MREAL1 ENUMERATOR                          *
*                                                                *
*           COPYRIGHT 1986 - THOMAS PRESTON MOORE                *
*                                                                *
*           WRITTEN IN FORTRAN 77 (IBM VS-FORTRAN)               *
*                                                                *
*                  LINE PRINTER OUTPUT FORMAT                    *
*                                                                *
******************************************************************
```

FROM ENUM: INITIAL BASE POINT: 12 2 20 16 1 30
ALL ADJACENT POINTS TO THIS BASE POINT ARE ENUMERATED BELOW:

|<******* DECISION VARIABLES ******>| |<************* SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ************>|

N1	M1	R1	N2	M2	R2	DESIGN(1)	DESIGN(2)	TOTAL COST	FLAG1	APRC	AOPC	ASUNA(1)	ASUNA(2)	ACUNA(1)	ACUNA(2)
11	1	19	15	1	29	2	2	.3417453E+07	5	290127.	815000.	0.439091E+08	0.826173E+07	-.151118E+08	0.999492E+05
11	1	19	15	1	30	2	2	.3423373E+07	5	297368.	815000.	0.439091E+08	0.815705E+07	-.151118E+08	0.999325E+05
11	1	20	15	1	29	2	2	.3438673E+07	5	294759.	815000.	0.426393E+08	0.826173E+07	-.216254E+08	0.999492E+05
11	1	20	15	1	30	2	2	.3444593E+07	5	302000.	815000.	0.426393E+08	0.815705E+07	-.216254E+08	0.999325E+05
11	1	19	15	2	29	2	2	.3606051E+07	5	290127.	685000.	0.439091E+08	0.830266E+07	-.151118E+08	0.999968E+05
11	1	19	15	2	30	2	2	.3611562E+07	5	297368.	685000.	0.439091E+08	0.820297E+07	-.151118E+08	0.999958E+05
11	1	20	15	2	29	2	2	.3627271E+07	5	294759.	685000.	0.426393E+08	0.830266E+07	-.216254E+08	0.999968E+05
11	1	20	15	2	30	2	2	.3632782E+07	5	302000.	685000.	0.426393E+08	0.820297E+07	-.216254E+08	0.999958E+05
11	2	19	15	1	29	2	2	.3479264E+07	5	290127.	725000.	0.474769E+08	0.826173E+07	-.906089E+06	0.999492E+05
11	2	19	15	1	30	2	2	.3485184E+07	5	297368.	725000.	0.474769E+08	0.815705E+07	-.906089E+06	0.999325E+05
11	2	20	15	1	29	2	2	.3486011E+07	5	294759.	725000.	0.470002E+08	0.826173E+07	-.166931E+07	0.999492E+05
11	2	20	15	1	30	2	2	.3491931E+07	5	302000.	725000.	0.470002E+08	0.815705E+07	-.166931E+07	0.999325E+05
11	2	19	15	2	29	2	2	.3667862E+07	5	290127.	595000.	0.474769E+08	0.830266E+07	-.906089E+06	0.999968E+05
11	2	19	15	2	30	2	2	.3673373E+07	5	297368.	595000.	0.474769E+08	0.820297E+07	-.906089E+06	0.999958E+05
11	2	20	15	2	29	2	2	.3674609E+07	5	294759.	595000.	0.470002E+08	0.830266E+07	-.166931E+07	0.999968E+05
11	2	20	15	2	30	2	2	.3680120E+07	5	302000.	595000.	0.470002E+08	0.820297E+07	-.166931E+07	0.999958E+05
11	3	19	15	1	29	2	2	.3603150E+07	0	290127.	635000.	0.476433E+08	0.826173E+07	0.174453E+06	0.999492E+05
11	3	19	15	1	30	2	2	.3609070E+07	0	297368.	635000.	0.476433E+08	0.815705E+07	0.174453E+06	0.999325E+05
11	3	20	15	1	29	2	2	.3608994E+07	5	294759.	635000.	0.472162E+08	0.826173E+07	-.153408E+06	0.999492E+05
11	3	20	15	1	30	2	2	.3614914E+07	5	302000.	635000.	0.472162E+08	0.815705E+07	-.153408E+06	0.999325E+05
11	3	19	15	2	29	2	2	.3791748E+07	0	290127.	505000.	0.476433E+08	0.830266E+07	0.174453E+06	0.999968E+05
11	3	19	15	2	30	2	2	.3797259E+07	0	297368.	505000.	0.476433E+08	0.820297E+07	0.174453E+06	0.999958E+05
11	3	20	15	2	29	2	2	.3797592E+07	5	294759.	505000.	0.472162E+08	0.830266E+07	-.153408E+06	0.999968E+05
11	3	20	15	2	30	2	2	.3803103E+07	5	302000.	505000.	0.472162E+08	0.820297E+07	-.153408E+06	0.999958E+05
11	1	19	16	1	29	2	2	.3388306E+07	5	275644.	750000.	0.439091E+08	0.995280E+07	-.151118E+08	0.999986E+05
11	1	19	16	1	30	2	2	.3385925E+07	5	283368.	750000.	0.439091E+08	0.994702E+07	-.151118E+08	0.999981E+05
11	1	20	16	1	29	2	2	.3409526E+07	5	280276.	750000.	0.426393E+08	0.995280E+07	-.216254E+08	0.999986E+05
11	1	20	16	1	30	2	2	.3407145E+07	5	288000.	750000.	0.426393E+08	0.994702E+07	-.216254E+08	0.999981E+05
11	1	19	16	2	29	2	2	.3578282E+07	5	275644.	620000.	0.439091E+08	0.997696E+07	-.151118E+08	0.999999E+05
11	1	19	16	2	30	2	2	.3575656E+07	5	283368.	620000.	0.439091E+08	0.997417E+07	-.151118E+08	0.999999E+05
11	1	20	16	2	29	2	2	.3599502E+07	5	280276.	620000.	0.426393E+08	0.997696E+07	-.216254E+08	0.999999E+05
11	1	20	16	2	30	2	2	.3596876E+07	5	288000.	620000.	0.426393E+08	0.997417E+07	-.216254E+08	0.999999E+05
11	2	19	16	1	29	2	2	.3450117E+07	5	275644.	660000.	0.474769E+08	0.995280E+07	-.906089E+06	0.999986E+05
11	2	19	16	1	30	2	2	.3447736E+07	5	283368.	660000.	0.474769E+08	0.994702E+07	-.906089E+06	0.999981E+05
11	2	20	16	1	29	2	2	.3456864E+07	5	280276.	660000.	0.470002E+08	0.995280E+07	-.166931E+07	0.999986E+05
11	2	20	16	1	30	2	2	.3454483E+07	5	288000.	660000.	0.470002E+08	0.994702E+07	-.166931E+07	0.999981E+05
11	2	19	16	2	29	2	2	.3640093E+07	5	275644.	530000.	0.474769E+08	0.997696E+07	-.906089E+06	0.999999E+05
11	2	19	16	2	30	2	2	.3637467E+07	5	283368.	530000.	0.474769E+08	0.997417E+07	-.906089E+06	0.999999E+05
11	2	20	16	2	29	2	2	.3646840E+07	5	280276.	530000.	0.470002E+08	0.997696E+07	-.166931E+07	0.999999E+05
11	2	20	16	2	30	2	2	.3644214E+07	5	288000.	530000.	0.470002E+08	0.997417E+07	-.166931E+07	0.999999E+05
11	3	19	16	1	29	2	2	.3574003E+07	0	275644.	570000.	0.476433E+08	0.995280E+07	0.174453E+06	0.999986E+05
11	3	19	16	1	30	2	2	.3571622E+07	0	283368.	570000.	0.476433E+08	0.994702E+07	0.174453E+06	0.999981E+05
11	3	20	16	1	29	2	2	.3579847E+07	5	280276.	570000.	0.472162E+08	0.995280E+07	-.153408E+06	0.999986E+05
11	3	20	16	1	30	2	2	.3577466E+07	5	288000.	570000.	0.472162E+08	0.994702E+07	-.153408E+06	0.999981E+05
11	3	19	16	2	29	2	2	.3763979E+07	0	275644.	440000.	0.476433E+08	0.997696E+07	0.174453E+06	0.999999E+05
11	3	19	16	2	30	2	2	.3761353E+07	0	283368.	440000.	0.476433E+08	0.997417E+07	0.174453E+06	0.999999E+05
11	3	20	16	2	29	2	2	.3769823E+07	5	280276.	440000.	0.472162E+08	0.997696E+07	-.153408E+06	0.999999E+05
11	3	20	16	2	30	2	2	.3767197E+07	5	288000.	440000.	0.472162E+08	0.997417E+07	-.153408E+06	0.999999E+05
11	1	19	17	1	29	2	2	.3494272E+07	5	261162.	685000.	0.439091E+08	0.999864E+07	-.151118E+08	0.999999E+05
11	1	19	17	1	30	2	2	.3491260E+07	5	269368.	685000.	0.439091E+08	0.999838E+07	-.151118E+08	0.999999E+05
11	1	20	17	1	29	2	2	.3515492E+07	5	265793.	685000.	0.426393E+08	0.999864E+07	-.216254E+08	0.999999E+05
11	1	20	17	1	30	2	2	.3512480E+07	5	274000.	685000.	0.426393E+08	0.999838E+07	-.216254E+08	0.999999E+05
11	1	19	17	2	29	2	2	.3686148E+07	5	261162.	555000.	0.439091E+08	0.999967E+07	-.151118E+08	0.100000E+06
11	1	19	17	2	30	2	2	.3683120E+07	5	269368.	555000.	0.439091E+08	0.999961E+07	-.151118E+08	0.100000E+06
11	1	20	17	2	29	2	2	.3707368E+07	5	265793.	555000.	0.426393E+08	0.999967E+07	-.216254E+08	0.100000E+06

11	1	20	17	2	30	2	2	.3704340E+07	5	274000.	555000.	0.426393E+08	0.999961E+07	-.216254E+08	0.100000E+06
11	2	19	17	1	29	2	2	.3556083E+07	5	261162.	595000.	0.474769E+08	0.999864E+07	-.906089E+06	0.999999E+05
11	2	19	17	1	30	2	2	.3553071E+07	5	269368.	595000.	0.474769E+08	0.999838E+07	-.906089E+06	0.999999E+05
11	2	20	17	1	29	2	2	.3562830E+07	5	265793.	595000.	0.470002E+08	0.999864E+07	-.166931E+07	0.999999E+05
11	2	20	17	1	30	2	2	.3559818E+07	5	274000.	595000.	0.470002E+08	0.999838E+07	-.166931E+07	0.999999E+05
11	2	19	17	2	29	2	2	.3747959E+07	5	261162.	465000.	0.474769E+08	0.999967E+07	-.906089E+06	0.100000E+06
11	2	19	17	2	30	2	2	.3744931E+07	5	269368.	465000.	0.474769E+08	0.999961E+07	-.906089E+06	0.100000E+06
11	2	20	17	2	29	2	2	.3754706E+07	5	265793.	465000.	0.470002E+08	0.999967E+07	-.166931E+07	0.100000E+06
11	2	20	17	2	30	2	2	.3751678E+07	5	274000.	465000.	0.470002E+08	0.999961E+07	-.166931E+07	0.100000E+06
11	3	19	17	1	29	2	2	.3679969E+07	0	261162.	505000.	0.476433E+08	0.999864E+07	0.174453E+06	0.999999E+05
11	3	19	17	1	30	2	2	.3676957E+07	0	269368.	505000.	0.476433E+08	0.999838E+07	0.174453E+06	0.999999E+05
11	3	20	17	1	29	2	2	.3685813E+07	5	265793.	505000.	0.472162E+08	0.999864E+07	-.153408E+06	0.999999E+05
11	3	20	17	1	30	2	2	.3682801E+07	5	274000.	505000.	0.472162E+08	0.999838E+07	-.153408E+06	0.999999E+05
11	3	19	17	2	29	2	2	.3871845E+07	0	261162.	375000.	0.476433E+08	0.999967E+07	0.174453E+06	0.100000E+06
11	3	19	17	2	30	2	2	.3868817E+07	0	269368.	375000.	0.476433E+08	0.999961E+07	0.174453E+06	0.100000E+06
11	3	20	17	2	29	2	2	.3877689E+07	5	265793.	375000.	0.472162E+08	0.999967E+07	-.153408E+06	0.100000E+06
11	3	20	17	2	30	2	2	.3874661E+07	5	274000.	375000.	0.472162E+08	0.999961E+07	-.153408E+06	0.100000E+06
12	1	19	15	1	29	2	2	.3443546E+07	5	281706.	725000.	0.484503E+08	0.826173E+07	-.309918E+07	0.999492E+05
12	1	19	15	1	30	2	2	.3449466E+07	5	288947.	725000.	0.484503E+08	0.815705E+07	-.309918E+07	0.999325E+05
12	1	20	15	1	29	2	2	.3450324E+07	5	286759.	725000.	0.479622E+08	0.826173E+07	-.526389E+07	0.999492E+05
12	1	20	15	1	30	2	2	.3456244E+07	5	294000.	725000.	0.479622E+08	0.815705E+07	-.526389E+07	0.999325E+05
12	1	19	15	2	29	2	2	.3632144E+07	5	281706.	595000.	0.484503E+08	0.830266E+07	-.309918E+07	0.999968E+05
12	1	19	15	2	30	2	2	.3637655E+07	5	288947.	595000.	0.484503E+08	0.820297E+07	-.309918E+07	0.999958E+05
12	1	20	15	2	29	2	2	.3638922E+07	5	286759.	595000.	0.479622E+08	0.830266E+07	-.526389E+07	0.999968E+05
12	1	20	15	2	30	2	2	.3644433E+07	5	294000.	595000.	0.479622E+08	0.820297E+07	-.526389E+07	0.999958E+05
12	2	19	15	1	29	2	2	.3548081E+07	0	281706.	635000.	0.496771E+08	0.826173E+07	0.756068E+06	0.999492E+05
12	2	19	15	1	30	2	2	.3554001E+07	0	288947.	635000.	0.496771E+08	0.815705E+07	0.756068E+06	0.999325E+05
12	2	20	15	1	29	2	2	.3547691E+07	0	286759.	635000.	0.495817E+08	0.826173E+07	0.627813E+06	0.999492E+05
12	2	20	15	1	30	2	2	.3553611E+07	0	294000.	635000.	0.495817E+08	0.815705E+07	0.627813E+06	0.999325E+05
12	2	19	15	2	29	2	2	.3736679E+07	0	281706.	505000.	0.496771E+08	0.830266E+07	0.756068E+06	0.999968E+05
12	2	19	15	2	30	2	2	.3742190E+07	0	288947.	505000.	0.496771E+08	0.820297E+07	0.756068E+06	0.999958E+05
12	2	20	15	2	29	2	2	.3736289E+07	0	286759.	505000.	0.495817E+08	0.830266E+07	0.627813E+06	0.999968E+05
12	2	20	15	2	30	2	2	.3741800E+07	0	294000.	505000.	0.495817E+08	0.820297E+07	0.627813E+06	0.999958E+05
12	3	19	15	1	29	2	2	.3672783E+07	0	281706.	545000.	0.497989E+08	0.826173E+07	0.929549E+06	0.999492E+05
12	3	19	15	1	30	2	2	.3678703E+07	0	288947.	545000.	0.497989E+08	0.815705E+07	0.929549E+06	0.999325E+05
12	3	20	15	1	29	2	2	.3671706E+07	0	286759.	545000.	0.497411E+08	0.826173E+07	0.892750E+06	0.999492E+05
12	3	20	15	1	30	2	2	.3677626E+07	0	294000.	545000.	0.497411E+08	0.815705E+07	0.892750E+06	0.999325E+05
12	3	19	15	2	29	2	2	.3861381E+07	0	281706.	415000.	0.497989E+08	0.830266E+07	0.929549E+06	0.999968E+05
12	3	19	15	2	30	2	2	.3866892E+07	0	288947.	415000.	0.497989E+08	0.820297E+07	0.929549E+06	0.999958E+05
12	3	20	15	2	29	2	2	.3860304E+07	0	286759.	415000.	0.497411E+08	0.830266E+07	0.892750E+06	0.999968E+05
12	3	20	15	2	30	2	2	.3865815E+07	0	294000.	415000.	0.497411E+08	0.820297E+07	0.892750E+06	0.999958E+05
12	1	19	16	1	29	2	2	.3414399E+07	5	267223.	660000.	0.484503E+08	0.995280E+07	-.309918E+07	0.999986E+05
12	1	19	16	1	30	2	2	.3412018E+07	5	274947.	660000.	0.484503E+08	0.994702E+07	-.309918E+07	0.999981E+05
12	1	20	16	1	29	2	2	.3421177E+07	5	272276.	660000.	0.479622E+08	0.995280E+07	-.526389E+07	0.999986E+05
12	1	20	16	1	30	2	2	.3418796E+07	5	280000.	660000.	0.479622E+08	0.994702E+07	-.526389E+07	0.999981E+05
12	1	19	16	2	29	2	2	.3604375E+07	5	267223.	530000.	0.484503E+08	0.997696E+07	-.309918E+07	0.999999E+05
12	1	19	16	2	30	2	2	.3601749E+07	5	274947.	530000.	0.484503E+08	0.997417E+07	-.309918E+07	0.999999E+05
12	1	20	16	2	29	2	2	.3611153E+07	5	272276.	530000.	0.479622E+08	0.997696E+07	-.526389E+07	0.999999E+05
12	1	20	16	2	30	2	2	.3608527E+07	5	280000.	530000.	0.479622E+08	0.997417E+07	-.526389E+07	0.999999E+05
12	2	19	16	1	29	2	2	.3518934E+07	0	267223.	570000.	0.496771E+08	0.995280E+07	0.756068E+06	0.999986E+05
12	2	19	16	1	30	2	2	.3516553E+07	0	274947.	570000.	0.496771E+08	0.994702E+07	0.756068E+06	0.999981E+05
12	2	20	16	1	29	2	2	.3518544E+07	0	272276.	570000.	0.495817E+08	0.995280E+07	0.627813E+06	0.999986E+05
12	2	20	16	1	30	2	2	.3516163E+07	0	280000.	570000.	0.495817E+08	0.994702E+07	0.627813E+06	0.999981E+05
12	2	19	16	2	29	2	2	.3708910E+07	0	267223.	440000.	0.496771E+08	0.997696E+07	0.756068E+06	0.999999E+05
12	2	19	16	2	30	2	2	.3706284E+07	0	274947.	440000.	0.496771E+08	0.997417E+07	0.756068E+06	0.999999E+05
12	2	20	16	2	29	2	2	.3708520E+07	0	272276.	440000.	0.495817E+08	0.997696E+07	0.627813E+06	0.999999E+05
12	2	20	16	2	30	2	2	.3705894E+07	0	280000.	440000.	0.495817E+08	0.997417E+07	0.627813E+06	0.999999E+05
12	3	19	16	1	29	2	2	.3643636E+07	0	267223.	480000.	0.497989E+08	0.995280E+07	0.929549E+06	0.999986E+05
12	3	19	16	1	30	2	2	.3641255E+07	0	274947.	480000.	0.497989E+08	0.994702E+07	0.929549E+06	0.999981E+05
12	3	20	16	1	29	2	2	.3642559E+07	0	272276.	480000.	0.497411E+08	0.995280E+07	0.892750E+06	0.999986E+05
12	3	20	16	1	30	2	2	.3640178E+07	0	280000.	480000.	0.497411E+08	0.994702E+07	0.892750E+06	0.999981E+05
12	3	19	16	2	29	2	2	.3833612E+07	0	267223.	350000.	0.497989E+08	0.997696E+07	0.929549E+06	0.999999E+05
12	3	19	16	2	30	2	2	.3830986E+07	0	274947.	350000.	0.497989E+08	0.997417E+07	0.929549E+06	0.999999E+05
12	3	20	16	2	29	2	2	.3832535E+07	0	272276.	350000.	0.497411E+08	0.997696E+07	0.892750E+06	0.999999E+05
12	3	20	16	2	30	2	2	.3829909E+07	0	280000.	350000.	0.497411E+08	0.997417E+07	0.892750E+06	0.999999E+05
12	1	19	17	1	29	2	2	.3520365E+07	5	252741.	595000.	0.484503E+08	0.999864E+07	-.309918E+07	0.999999E+05
12	1	19	17	1	30	2	2	.3517353E+07	5	260947.	595000.	0.484503E+08	0.999838E+07	-.309918E+07	0.999999E+05
12	1	20	17	1	29	2	2	.3527143E+07	5	257793.	595000.	0.479622E+08	0.999864E+07	-.526389E+07	0.999999E+05
12	1	20	17	1	30	2	2	.3524131E+07	5	266000.	595000.	0.479622E+08	0.999838E+07	-.526389E+07	0.999999E+05
12	1	19	17	2	29	2	2	.3712241E+07	5	252741.	465000.	0.484503E+08	0.999967E+07	-.309918E+07	0.100000E+06
12	1	19	17	2	30	2	2	.3709213E+07	5	260947.	465000.	0.484503E+08	0.999961E+07	-.309918E+07	0.100000E+06
12	1	20	17	2	29	2	2	.3719019E+07	5	257793.	465000.	0.479622E+08	0.999967E+07	-.526389E+07	0.100000E+06
12	1	20	17	2	30	2	2	.3715991E+07	5	266000.	465000.	0.479622E+08	0.999961E+07	-.526389E+07	0.100000E+06

12	2	19	17	1	29	2	2	.3624900E+07	0	252741.	505000.	0.496771E+08	0.999864E+07	0.756068E+06	0.999999E+05
12	2	19	17	1	30	2	2	.3621888E+07	0	260947.	505000.	0.496771E+08	0.999838E+07	0.756068E+06	0.999999E+05
12	2	20	17	1	29	2	2	.3624510E+07	0	257793.	505000.	0.495817E+08	0.999864E+07	0.627813E+06	0.999999E+05
12	2	20	17	1	30	2	2	.3621498E+07	0	266000.	505000.	0.495817E+08	0.999838E+07	0.627813E+06	0.999999E+05
12	2	19	17	2	29	2	2	.3816776E+07	0	252741.	375000.	0.496771E+08	0.999967E+07	0.756068E+06	0.100000E+06
12	2	19	17	2	30	2	2	.3813748E+07	0	260947.	375000.	0.496771E+08	0.999961E+07	0.756068E+06	0.100000E+06
12	2	20	17	2	29	2	2	.3816386E+07	0	257793.	375000.	0.495817E+08	0.999967E+07	0.627813E+06	0.100000E+06
12	2	20	17	2	30	2	2	.3813358E+07	0	266000.	375000.	0.495817E+08	0.999961E+07	0.627813E+06	0.100000E+06
12	3	19	17	1	29	2	2	.3749602E+07	0	252741.	415000.	0.497989E+08	0.999864E+07	0.929549E+06	0.999999E+05
12	3	19	17	1	30	2	2	.3746590E+07	0	260947.	415000.	0.497989E+08	0.999838E+07	0.929549E+06	0.999999E+05
12	3	20	17	1	29	2	2	.3748525E+07	0	257793.	415000.	0.497411E+08	0.999864E+07	0.892750E+06	0.999999E+05
12	3	20	17	1	30	2	2	.3745513E+07	0	266000.	415000.	0.497411E+08	0.999838E+07	0.892750E+06	0.999999E+05
12	3	19	17	2	29	2	2	.3941478E+07	0	252741.	285000.	0.497989E+08	0.999967E+07	0.929549E+06	0.100000E+06
12	3	19	17	2	30	2	2	.3938450E+07	0	260947.	285000.	0.497989E+08	0.999961E+07	0.929549E+06	0.100000E+06
12	3	20	17	2	29	2	2	.3940401E+07	0	257793.	285000.	0.497411E+08	0.999967E+07	0.892750E+06	0.100000E+06
12	3	20	17	2	30	2	2	.3937373E+07	0	266000.	285000.	0.497411E+08	0.999961E+07	0.892750E+06	0.100000E+06
13	1	19	15	1	29	2	2	.3532025E+07	5	273285.	635000.	0.495732E+08	0.826173E+07	-.128991E+06	0.999492E+05
13	1	19	15	1	30	2	2	.3537945E+07	5	280526.	635000.	0.495732E+08	0.815705E+07	-.128991E+06	0.999325E+05
13	1	20	15	1	29	2	2	.3533071E+07	5	278759.	635000.	0.493894E+08	0.826173E+07	-.876957E+06	0.999492E+05
13	1	20	15	1	30	2	2	.3538991E+07	5	286000.	635000.	0.493894E+08	0.815705E+07	-.876957E+06	0.999325E+05
13	1	19	15	2	29	2	2	.3720623E+07	5	273285.	505000.	0.495732E+08	0.830266E+07	-.128991E+06	0.999968E+05
13	1	19	15	2	30	2	2	.3726134E+07	5	280526.	505000.	0.495732E+08	0.820297E+07	-.128991E+06	0.999958E+05
13	1	20	15	2	29	2	2	.3721669E+07	5	278759.	505000.	0.493894E+08	0.830266E+07	-.876957E+06	0.999968E+05
13	1	20	15	2	30	2	2	.3727180E+07	5	286000.	505000.	0.493894E+08	0.820297E+07	-.876957E+06	0.999958E+05
13	2	19	15	1	29	2	2	.3651976E+07	0	273285.	545000.	0.499552E+08	0.826173E+07	0.966183E+06	0.999492E+05
13	2	19	15	1	30	2	2	.3657896E+07	0	280526.	545000.	0.499552E+08	0.815705E+07	0.966183E+06	0.999325E+05
13	2	20	15	1	29	2	2	.3650004E+07	0	278759.	545000.	0.499368E+08	0.826173E+07	0.943785E+06	0.999492E+05
13	2	20	15	1	30	2	2	.3655924E+07	0	286000.	545000.	0.499368E+08	0.815705E+07	0.943785E+06	0.999325E+05
13	2	19	15	2	29	2	2	.3840574E+07	0	273285.	415000.	0.499552E+08	0.830266E+07	0.966183E+06	0.999968E+05
13	2	19	15	2	30	2	2	.3846085E+07	0	280526.	415000.	0.499552E+08	0.820297E+07	0.966183E+06	0.999958E+05
13	2	20	15	2	29	2	2	.3838602E+07	0	278759.	415000.	0.499368E+08	0.830266E+07	0.943785E+06	0.999968E+05
13	2	20	15	2	30	2	2	.3844113E+07	0	286000.	415000.	0.499368E+08	0.820297E+07	0.943785E+06	0.999958E+05
13	3	19	15	1	29	2	2	.3778423E+07	0	273285.	455000.	0.499814E+08	0.826173E+07	0.993487E+06	0.999492E+05
13	3	19	15	1	30	2	2	.3784343E+07	0	280526.	455000.	0.499814E+08	0.815705E+07	0.993487E+06	0.999325E+05
13	3	20	15	1	29	2	2	.3776251E+07	0	278759.	455000.	0.499739E+08	0.826173E+07	0.989196E+06	0.999492E+05
13	3	20	15	1	30	2	2	.3782171E+07	0	286000.	455000.	0.499739E+08	0.815705E+07	0.989196E+06	0.999325E+05
13	3	19	15	2	29	2	2	.3967021E+07	0	273285.	325000.	0.499814E+08	0.830266E+07	0.993487E+06	0.999968E+05
13	3	19	15	2	30	2	2	.3972532E+07	0	280526.	325000.	0.499814E+08	0.820297E+07	0.993487E+06	0.999958E+05
13	3	20	15	2	29	2	2	.3964849E+07	0	278759.	325000.	0.499739E+08	0.830266E+07	0.989196E+06	0.999968E+05
13	3	20	15	2	30	2	2	.3970360E+07	0	286000.	325000.	0.499739E+08	0.820297E+07	0.989196E+06	0.999958E+05
13	1	19	16	1	29	2	2	.3502878E+07	5	258802.	570000.	0.495732E+08	0.995280E+07	-.128991E+06	0.999986E+05
13	1	19	16	1	30	2	2	.3500497E+07	5	266526.	570000.	0.495732E+08	0.994702E+07	-.128991E+06	0.999981E+05
13	1	20	16	1	29	2	2	.3503924E+07	5	264276.	570000.	0.493894E+08	0.995280E+07	-.876957E+06	0.999986E+05
13	1	20	16	1	30	2	2	.3501543E+07	5	272000.	570000.	0.493894E+08	0.994702E+07	-.876957E+06	0.999981E+05
13	1	19	16	2	29	2	2	.3692854E+07	5	258802.	440000.	0.495732E+08	0.997696E+07	-.128991E+06	0.999999E+05
13	1	19	16	2	30	2	2	.3690228E+07	5	266526.	440000.	0.495732E+08	0.997417E+07	-.128991E+06	0.999999E+05
13	1	20	16	2	29	2	2	.3693900E+07	5	264276.	440000.	0.493894E+08	0.997696E+07	-.876957E+06	0.999999E+05
13	1	20	16	2	30	2	2	.3691274E+07	5	272000.	440000.	0.493894E+08	0.997417E+07	-.876957E+06	0.999999E+05
13	2	19	16	1	29	2	2	.3622829E+07	0	258802.	480000.	0.499552E+08	0.995280E+07	0.966183E+06	0.999986E+05
13	2	19	16	1	30	2	2	.3620448E+07	0	266526.	480000.	0.499552E+08	0.994702E+07	0.966183E+06	0.999981E+05
13	2	20	16	1	29	2	2	.3620857E+07	0	264276.	480000.	0.499368E+08	0.995280E+07	0.943785E+06	0.999986E+05
13	2	20	16	1	30	2	2	.3618476E+07	0	272000.	480000.	0.499368E+08	0.994702E+07	0.943785E+06	0.999981E+05
13	2	19	16	2	29	2	2	.3812805E+07	0	258802.	350000.	0.499552E+08	0.997696E+07	0.966183E+06	0.999999E+05
13	2	19	16	2	30	2	2	.3810179E+07	0	266526.	350000.	0.499552E+08	0.997417E+07	0.966183E+06	0.999999E+05
13	2	20	16	2	29	2	2	.3810833E+07	0	264276.	350000.	0.499368E+08	0.997696E+07	0.943785E+06	0.999999E+05
13	2	20	16	2	30	2	2	.3808207E+07	0	272000.	350000.	0.499368E+08	0.997417E+07	0.943785E+06	0.999999E+05
13	3	19	16	1	29	2	2	.3749276E+07	0	258802.	390000.	0.499814E+08	0.995280E+07	0.993487E+06	0.999986E+05
13	3	19	16	1	30	2	2	.3746895E+07	0	266526.	390000.	0.499814E+08	0.994702E+07	0.993487E+06	0.999981E+05
13	3	20	16	1	29	2	2	.3747104E+07	0	264276.	390000.	0.499739E+08	0.995280E+07	0.989196E+06	0.999986E+05
13	3	20	16	1	30	2	2	.3744723E+07	0	272000.	390000.	0.499739E+08	0.994702E+07	0.989196E+06	0.999981E+05
13	3	19	16	2	29	2	2	.3939252E+07	0	258802.	260000.	0.499814E+08	0.997696E+07	0.993487E+06	0.999999E+05
13	3	19	16	2	30	2	2	.3936626E+07	0	266526.	260000.	0.499814E+08	0.997417E+07	0.993487E+06	0.999999E+05
13	3	20	16	2	29	2	2	.3937080E+07	0	264276.	260000.	0.499739E+08	0.997696E+07	0.989196E+06	0.999999E+05
13	3	20	16	2	30	2	2	.3934454E+07	0	272000.	260000.	0.499739E+08	0.997417E+07	0.989196E+06	0.999999E+05
13	1	19	17	1	29	2	2	.3608844E+07	5	244320.	505000.	0.495732E+08	0.999864E+07	-.128991E+06	0.999999E+05
13	1	19	17	1	30	2	2	.3605832E+07	5	252526.	505000.	0.495732E+08	0.999838E+07	-.128991E+06	0.999999E+05
13	1	20	17	1	29	2	2	.3609890E+07	5	249793.	505000.	0.493894E+08	0.999864E+07	-.876957E+06	0.999999E+05
13	1	20	17	1	30	2	2	.3606878E+07	5	258000.	505000.	0.493894E+08	0.999838E+07	-.876957E+06	0.999999E+05
13	1	19	17	2	29	2	2	.3800720E+07	5	244320.	375000.	0.495732E+08	0.999967E+07	-.128991E+06	0.100000E+06
13	1	19	17	2	30	2	2	.3797692E+07	5	252526.	375000.	0.495732E+08	0.999961E+07	-.128991E+06	0.100000E+06
13	1	20	17	2	29	2	2	.3801766E+07	5	249793.	375000.	0.493894E+08	0.999967E+07	-.876957E+06	0.100000E+06
13	1	20	17	2	30	2	2	.3798738E+07	5	258000.	375000.	0.493894E+08	0.999961E+07	-.876957E+06	0.100000E+06
13	2	19	17	1	29	2	2	.3728795E+07	0	244320.	415000.	0.499552E+08	0.999864E+07	0.966183E+06	0.999999E+05

13	2	19	17	1	30	2	2	.3725783E+07	0	252526.	415000.	0.499552E+08	0.999838E+07	0.966183E+06	0.999999E+05
13	2	20	17	1	29	2	2	.3726823E+07	0	249793.	415000.	0.499368E+08	0.999864E+07	0.943785E+06	0.999999E+05
13	2	20	17	1	30	2	2	.3723811E+07	0	258000.	415000.	0.499368E+08	0.999838E+07	0.943785E+06	0.999999E+05
13	2	19	17	2	29	2	2	.3920671E+07	0	244320.	285000.	0.499552E+08	0.999967E+07	0.966183E+06	0.100000E+06
13	2	19	17	2	30	2	2	.3917643E+07	0	252526.	285000.	0.499552E+08	0.999961E+07	0.966183E+06	0.100000E+06
13	2	20	17	2	29	2	2	.3918699E+07	0	249793.	285000.	0.499368E+08	0.999967E+07	0.943785E+06	0.100000E+06
13	2	20	17	2	30	2	2	.3915671E+07	0	258000.	285000.	0.499368E+08	0.999961E+07	0.943785E+06	0.100000E+06
13	3	19	17	1	29	2	2	.3855242E+07	0	244320.	325000.	0.499814E+08	0.999864E+07	0.993487E+06	0.999999E+05
13	3	19	17	1	30	2	2	.3852230E+07	0	252526.	325000.	0.499814E+08	0.999838E+07	0.993487E+06	0.999999E+05
13	3	20	17	1	29	2	2	.3853070E+07	0	249793.	325000.	0.499739E+08	0.999864E+07	0.989196E+06	0.999999E+05
13	3	20	17	1	30	2	2	.3850058E+07	0	258000.	325000.	0.499739E+08	0.999838E+07	0.989196E+06	0.999999E+05
13	3	19	17	2	29	2	2	.4047118E+07	0	244320.	195000.	0.499814E+08	0.999967E+07	0.993487E+06	0.100000E+06
13	3	19	17	2	30	2	2	.4044090E+07	0	252526.	195000.	0.499814E+08	0.999961E+07	0.993487E+06	0.100000E+06
13	3	20	17	2	29	2	2	.4044946E+07	0	249793.	195000.	0.499739E+08	0.999967E+07	0.989196E+06	0.100000E+06
13	3	20	17	2	30	2	2	.4041918E+07	0	258000.	195000.	0.499739E+08	0.999961E+07	0.989196E+06	0.100000E+06

N1	M1	R1	N2	M2	R2	DESIGN(1)	DESIGN(2)	BEST COST	APRC	AOPC	ASUNA(1)	ASUNA(2)	ACUNA(1)	ACUNA(2)
11	1	19	16	1	30	2	2	3385925.	283368.	750000.	0.439091E+08	0.994702E+07	-.151118E+08	0.999981E+05

N1	M1	R1	N2	M2	R2	DESIGN(1)	DESIGN(2)	BEST FEAS. COST	APRC	AOPC	ASUNA(1)	ASUNA(2)	ACUNA(1)	ACUNA(2)
12	2	20	16	1	30	2	2	3516163.	280000.	570000.	0.495817E+08	0.994702E+07	0.627813E+06	0.999981E+05

Appendix E

SAMPLE OPTIMIZATION OUTPUT

```
*******************************************************************
*                                                                 *
*                       MREAL1 OPTIMIZER                          *
*                                                                 *
*            COPYRIGHT 1986 - THOMAS PRESTON MOORE                *
*                                                                 *
*           WRITTEN IN FORTRAN 77 (IBM VS-FORTRAN)                *
*                                                                 *
*                  LINE PRINTER OUTPUT FORMAT                     *
*                                                                 *
*******************************************************************

                    DESIGN COMBINATION:   EQUIPMENT 1:  1      EQUIPMENT 2:  1

FROM INIT: INITIAL BASE POINT:                11    2   11   16    2   19
FROM INIT: INITIAL PENALTY PARAMETERS:       1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT: INITIAL LAGRANGE MULTIPLIERS:     0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2  TOTAL COST  FLAGC      APRC      AOPC       ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   1       1    11  2 11 16  2 19  .3743403E+07 000000   165263.   340000.  0.485967E+08 0.997696E+07 0.395368E+06 0.999999E+05
   1       1    12  2 11 16  2 19  .3839782E+07 000000   152536.   240000.  0.498658E+08 0.997696E+07 0.942187E+06 0.999999E+05
   1       1    10  2 11 16  2 19  .1739465E+14 000010   177991.   440000.  0.339902E+08 0.997696E+07 -.589825E+07 0.999999E+05
   1       1    11  3 11 16  2 19  .3869072E+07 000000   165263.   250000.  0.486655E+08 0.997696E+07 0.736389E+06 0.999999E+05
   1       1    11  1 11 16  2 19  .8244067E+13 000010   165263.   430000.  0.467668E+08 0.997696E+07 -.406056E+07 0.999999E+05
   1       1    11  2 12 16  2 19  .3743702E+07 000000   176930.   340000.  0.483978E+08 0.997696E+07 0.215700E+06 0.999999E+05
   1       1    11  2 10 16  2 19  .3744836E+07 000000   151263.   340000.  0.487087E+08 0.997696E+07 0.486508E+06 0.999999E+05
   1       1    11  2 11 17  2 19  .3851737E+07 000000   146842.   270000.  0.485967E+08 0.999967E+07 0.395368E+06 0.100000E+06
   1       1    11  2 11 15  2 19  .3770704E+07 000000   183684.   410000.  0.485967E+08 0.830266E+07 0.395368E+06 0.999968E+05
   1       1    11  2 11 16  3 19  .3935348E+07 000000   165263.   210000.  0.485967E+08 0.997714E+07 0.395368E+06 0.999999E+05
   1       1    11  2 11 16  1 19  .3553427E+07 000000   165263.   470000.  0.485967E+08 0.995280E+07 0.395368E+06 0.999986E+05
   1       1    11  2 11 16  2 20  .3738660E+07 000000   180000.   340000.  0.485967E+08 0.997417E+07 0.395368E+06 0.999999E+05
   1       1    11  2 11 16  1 19  .3553427E+07 000000   165263.   470000.  0.485967E+08 0.995280E+07 0.395368E+06 0.999986E+05
   1       1    11  2 11 16  1 19  .4053427E+07 000000   165263.   470000.  0.485967E+08 0.995280E+07 0.395368E+06 0.999986E+05
   1       1    12  2 11 16  1 19  .3649806E+07 000000   152536.   370000.  0.498658E+08 0.995280E+07 0.942187E+06 0.999986E+05
   1       1    10  2 11 16  1 19  .1739465E+14 000010   177991.   570000.  0.339902E+08 0.995280E+07 -.589825E+07 0.999986E+05
   1       1    11  3 11 16  1 19  .3679096E+07 000000   165263.   380000.  0.486655E+08 0.995280E+07 0.736389E+06 0.999986E+05
   1       1    11  1 11 16  1 19  .8244067E+13 000010   165263.   560000.  0.467668E+08 0.995280E+07 -.406056E+07 0.999986E+05
   1       1    11  2 12 16  1 19  .3553726E+07 000000   176930.   470000.  0.483978E+08 0.995280E+07 0.215700E+06 0.999986E+05
   1       1    11  2 10 16  1 19  .3554860E+07 000000   151263.   470000.  0.487087E+08 0.995280E+07 0.486508E+06 0.999986E+05
   1       1    11  2 11 17  1 19  .3659861E+07 000000   146842.   400000.  0.485967E+08 0.999864E+07 0.395368E+06 0.999999E+05
   1       1    11  2 11 15  1 19  .3582106E+07 000000   183684.   540000.  0.485967E+08 0.826173E+07 0.395368E+06 0.999492E+05
   1       1    11  2 11 16  2 19  .3743403E+07 000000   165263.   340000.  0.485967E+08 0.997696E+07 0.395368E+06 0.999999E+05
   1       1    11  2 11 16  1 19  .4053427E+07 000000   165263.   470000.  0.485967E+08 0.995280E+07 0.395368E+06 0.999986E+05
   1       1    11  2 11 16  1 20  .3548929E+07 000000   180000.   470000.  0.485967E+08 0.994702E+07 0.395368E+06 0.999981E+05
   1       1    11  2 11 16  1 21  .3545797E+07 000000   193333.   470000.  0.485967E+08 0.992763E+07 0.395368E+06 0.999951E+05
   1       1    11  2 11 16  1 22  .3542522E+07 000000   205455.   470000.  0.485967E+08 0.991297E+07 0.395368E+06 0.999916E+05
   1       1    11  2 11 16  1 24  .3537764E+07 000000   226667.   470000.  0.485967E+08 0.987023E+07 0.395368E+06 0.999727E+05
   1       1    11  2 11 16  1 25  .5535563E+07 000000   236000.   470000.  0.485967E+08 0.985067E+07 0.395368E+06 0.999587E+05
   1       1    11  2 11 16  1 25  .3535563E+07 000000   236000.   470000.  0.485967E+08 0.985067E+07 0.395368E+06 0.999587E+05
   1       1    11  2 11 16  1 25  .4035563E+07 000000   236000.   470000.  0.485967E+08 0.985067E+07 0.395368E+06 0.999587E+05
   1       1    12  2 11 16  1 25  .3631942E+07 000000   223273.   370000.  0.498658E+08 0.985067E+07 0.942187E+06 0.999587E+05
   1       1    10  2 11 16  1 25  .1739465E+14 000010   248727.   570000.  0.339902E+08 0.985067E+07 -.589825E+07 0.999587E+05
   1       1    11  3 11 16  1 25  .3661232E+07 000000   236000.   380000.  0.486655E+08 0.985067E+07 0.736389E+06 0.999587E+05
   1       1    11  1 11 16  1 25  .8244067E+13 000010   236000.   560000.  0.467668E+08 0.985067E+07 -.406056E+07 0.999587E+05
   1       1    11  2 12 16  1 25  .3535862E+07 000000   247667.   470000.  0.483978E+08 0.985067E+07 0.215700E+06 0.999587E+05
   1       1    11  2 10 16  1 25  .3536996E+07 000000   222000.   470000.  0.487087E+08 0.985067E+07 0.486508E+06 0.999587E+05
   1       1    11  2 11 17  1 25  .3632480E+07 000000   222000.   400000.  0.485967E+08 0.999241E+07 0.395368E+06 0.999979E+05
   1       1    11  2 11 15  1 25  .3671044E+07 000000   250000.   540000.  0.485967E+08 0.687910E+07 0.395368E+06 0.991378E+05
   1       1    11  2 11 16  2 25  .3721139E+07 000000   236000.   340000.  0.485967E+08 0.992840E+07 0.395368E+06 0.999987E+05
   1       1    11  2 11 16  1 25  .4035563E+07 000000   236000.   470000.  0.485967E+08 0.985067E+07 0.395368E+06 0.999587E+05
   1       1    11  2 11 16  1 25  .4035563E+07 000000   236000.   470000.  0.485967E+08 0.985067E+07 0.395368E+06 0.999587E+05
   1       1    11  2 11 16  1 24  .3537764E+07 000000   226667.   470000.  0.485967E+08 0.987023E+07 0.395368E+06 0.999727E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                11  2 11 16  1 25       3535563.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 12  1 11 17  1 25  FLAGC =000000
FROM LOCMIN: NEXT GRAD SEARCH POINT: 13  1 11 18  1 25  FLAGC =000000
FROM LOCMIN: THERE ARE  0 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  2 11 17  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  13  2 11 18  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  14  2 11 19  1 25   FLAG1= 2
```

```
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  15  2 11 20  1 25    FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  16  2 11 21  1 25    FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  17  2 11 22  1 25    FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  18  2 11 22  1 25    FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  19  2 11 23  1 25    FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  20  2 11 24  1 25    FLAG1= 2
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   1       1      11  2 11 16  1 25      3535563.              236000.     470000.  0.485967E+08 0.985067E+07 0.395368E+06 0.999587E+05

                                        DESIGN COMBINATION:   EQUIPMENT 1:  1      EQUIPMENT 2:  1

FROM INIT1: INITIAL BASE POINT:                 13    1    8   18    1   17
FROM INIT1: INITIAL PENALTY PARAMETERS:     1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT1: INITIAL LAGRANGE MULTIPLIERS:  0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2    TOTAL COST  FLAGC          APRC         AOPC         ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   1       1      13  1  8 18  1 17  .3882849E+07 000000         1912.      220000.  0.499218E+08 0.999998E+07 0.901587E+06 0.100000E+06
   1       1      14  1  8 18  1 17  .1254980E+09 100000       -15588.      120000.  0.499845E+08 0.999998E+07 0.980548E+06 0.100000E+06
   1       1      12  1  8 18  1 17  .3768663E+07 000000        19412.      320000.  0.495743E+08 0.999998E+07 0.463961E+06 0.100000E+06
   1       1      13  2  8 18  1 17  .4008505E+07 000000         1912.      130000.  0.499913E+08 0.999998E+07 0.997034E+06 0.100000E+06
   1       1      13  1  9 18  1 17  .3878659E+07 000000        27190.      220000.  0.499068E+08 0.999998E+07 0.873112E+06 0.100000E+06
   1       1      13  1  8 19  1 17  .1783983E+09 100000       -18676.      150000.  0.499218E+08 0.100000E+08 0.901587E+06 0.100000E+06
   1       1      13  1  8 17  1 17  .3772040E+07 000000        22500.      290000.  0.499218E+08 0.999924E+07 0.901587E+06 0.999999E+05
   1       1      13  1  8 18  2 17  .4074808E+07 000000         1912.       90000.  0.499218E+08 0.100000E+08 0.901587E+06 0.100000E+06
   1       1      13  1  8 18  1 18  .3876663E+07 000000        22500.      220000.  0.499218E+08 0.999997E+07 0.901587E+06 0.100000E+06
   1       1      12  1  8 17  1 17  .3657854E+07 000000        40000.      390000.  0.495743E+08 0.999924E+07 0.463961E+06 0.999999E+05
   1       1      11  1  8 16  1 17  .2337456E+13 000010        78088.      560000.  0.474886E+08 0.996811E+07 -.216215E+07 0.999996E+05
   1       1      13  1  8 18  1 17  .3882849E+07 000000         1912.      220000.  0.499218E+08 0.999998E+07 0.901587E+06 0.100000E+06
   1       1      11  1  8 16  1 17  .2337456E+13 000010        78088.      560000.  0.474886E+08 0.996811E+07 -.216215E+07 0.999996E+05
   1       1      12  2  8 17  1 17  .3778725E+07 000000        40000.      300000.  0.499059E+08 0.999924E+07 0.967772E+06 0.999999E+05
   1       1      12  1  8 17  1 17  .4157854E+07 000000        40000.      390000.  0.495743E+08 0.999924E+07 0.463961E+06 0.999999E+05
   1       1      12  1  9 17  1 17  .3654840E+07 000000        63333.      390000.  0.495136E+08 0.999924E+07 0.337480E+06 0.999999E+05
   1       1      12  1 10 17  1 17  .3651887E+07 000000        82000.      390000.  0.494587E+08 0.999924E+07 0.214830E+06 0.999999E+05
   1       1      12  1 11 17  1 17  .3649466E+07 000000        97273.      390000.  0.493837E+08 0.999924E+07 0.353311E+05 0.999999E+05
   1       1      12  1 13 17  1 17  .4974367E+12 000010       120769.      390000.  0.490220E+08 0.999924E+07 -.997430E+06 0.999999E+05
   1       1      12  1 12 17  1 17  .5745089E+11 000010       110000.      390000.  0.492416E+08 0.999924E+07 -.338961E+06 0.999999E+05
   1       1      11  1 11 18  1 17  .8244067E+13 000010        89412.      420000.  0.467668E+08 0.999998E+07 -.406056E+07 0.100000E+06
   1       1      13  1 11 16  1 17  .3651766E+07 000000       105134.      360000.  0.498728E+08 0.996811E+07 0.800858E+06 0.999996E+05
   1       1      12  1 11 17  2 17  .3841379E+07 000000        97273.      260000.  0.493837E+08 0.999981E+07 0.353311E+05 0.100000E+06
   1       1      12  1 11 17  1 17  .4149466E+07 000000        97273.      390000.  0.493837E+08 0.999924E+07 0.353311E+05 0.999999E+05
   1       1      12  1 11 17  1 18  .3643654E+07 000000       116717.      390000.  0.493837E+08 0.999887E+07 0.353311E+05 0.999999E+05
   1       1      12  1 11 17  1 19  .3638116E+07 000000       134115.      390000.  0.493837E+08 0.999864E+07 0.353311E+05 0.999999E+05
   1       1      12  1 11 17  1 20  .3632854E+07 000000       149773.      390000.  0.493837E+08 0.999838E+07 0.353311E+05 0.999999E+05
   1       1      12  1 11 17  1 22  .3623221E+07 000000       176818.      390000.  0.493837E+08 0.999661E+07 0.353311E+05 0.999996E+05
   1       1      12  1 11 17  1 25  .3610735E+07 000000       209273.      390000.  0.493837E+08 0.999241E+07 0.353311E+05 0.999979E+05
   1       1      12  1 11 17  1 25  .1611074E+08 000000       209273.      390000.  0.493837E+08 0.999241E+07 0.353311E+05 0.999979E+05
   1       1      12  1 11 17  1 25  .5610735E+07 000000       209273.      390000.  0.493837E+08 0.999241E+07 0.353311E+05 0.999979E+05
   1       1      12  1 11 17  1 24  .3614663E+07 000000       199356.      390000.  0.493837E+08 0.999384E+07 0.353311E+05 0.999987E+05
   1       1      12  1 11 17  1 25  .4110735E+07 000000       209273.      390000.  0.493837E+08 0.999241E+07 0.353311E+05 0.999979E+05
   1       1      13  1 11 18  1 25  .3829317E+07 000000       182546.      220000.  0.498728E+08 0.999959E+07 0.800858E+06 0.999999E+05
   1       1      11  1 11 16  1 25  .8244067E+13 000010       236000.      560000.  0.467668E+08 0.985067E+07 -.406056E+07 0.999587E+05
   1       1      12  2 11 17  1 25  .3728859E+07 000000       209273.      300000.  0.498658E+08 0.999241E+07 0.942187E+06 0.999979E+05
   1       1      12  1 11 17  1 25  .4110735E+07 000000       209273.      390000.  0.493837E+08 0.999241E+07 0.353311E+05 0.999979E+05
   1       1      12  1 11 17  1 25  .4110735E+07 000000       209273.      390000.  0.493837E+08 0.999241E+07 0.353311E+05 0.999979E+05
   1       1      12  1 11 17  1 24  .3614663E+07 000000       199356.      390000.  0.493837E+08 0.999384E+07 0.353311E+05 0.999987E+05
   1       1      11  1 11 18  1 25  .8244067E+13 000010       208000.      420000.  0.467668E+08 0.999959E+07 -.406056E+07 0.999999E+05
   1       1      13  1 11 16  1 25  .3624431E+07 000000       210546.      360000.  0.498728E+08 0.985067E+07 0.800858E+06 0.999587E+05
   1       1      12  1 11 17  2 25  .3802221E+07 000000       209273.      260000.  0.493837E+08 0.999818E+07 0.353311E+05 0.999999E+05
   1       1      12  1 11 17  1 25  .4110735E+07 000000       209273.      390000.  0.493837E+08 0.999241E+07 0.353311E+05 0.999979E+05
   1       1      12  1 10 17  1 25  .3613156E+07 000000       194000.      390000.  0.494587E+08 0.999241E+07 0.214830E+06 0.999979E+05
   1       1      12  1 12 17  1 25  .5745085E+11 000010       222000.      390000.  0.492416E+08 0.999241E+07 -.338961E+06 0.999979E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                  12  1 11 17  1 25      3610735.
FROM LOCMIN: RETURN 3
   1       1      12  1 11 16  1 25  .3513818E+07 000000       223273.      460000.  0.493837E+08 0.985067E+07 0.353311E+05 0.999587E+05
   1       1      13  1 11 16  1 25  .3624431E+07 000000       210546.      360000.  0.498728E+08 0.985067E+07 0.800858E+06 0.999587E+05
   1       1      11  1 11 16  1 25  .8244067E+13 000010       236000.      560000.  0.467668E+08 0.985067E+07 -.406056E+07 0.999587E+05
   1       1      12  2 11 16  1 25  .3631942E+07 000000       223273.      370000.  0.498658E+08 0.985067E+07 0.942187E+06 0.999587E+05
   1       1      12  1 12 16  1 25  .5745075E+11 000010       236000.      460000.  0.492416E+08 0.985067E+07 -.338961E+06 0.999587E+05
```

```
   1       1     12  1 10 16  1 25  .3516239E+07 000000      208000.      460000.  0.494587E+08 0.985067E+07 0.214830E+06 0.999587E+05
   1       1     12  1 11 17  1 25  .3610735E+07 000000      209273.      390000.  0.493837E+08 0.999241E+07 0.353311E+05 0.999979E+05
   1       1     12  1 11 15  1 25  .3649299E+07 000000      237273.      530000.  0.493837E+08 0.687910E+07 0.353311E+05 0.991378E+05
   1       1     12  1 11 16  2 25  .3699394E+07 000000      223273.      330000.  0.493837E+08 0.992840E+07 0.353311E+05 0.999987E+05
   1       1     12  1 11 16  1 24  .3516019E+07 000000      213939.      460000.  0.493837E+08 0.987023E+07 0.353311E+05 0.999727E+05
FROM INNER: RESOLUTION VALLEY OR LOCAL MINIMUM ENCOUNTERED IN GRADIENT APPOXIMATION SECTION - RETURN TO DUNLOP.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 11 17  1 25  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  1 11 17  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 11 18  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 11 19  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 10 20  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 10 21  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 10 22  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 23  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 24  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 25  1 25   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   1       1     12  1 11 16  1 25      3513818.             223273.      460000.  0.493837E+08 0.985067E+07 0.353311E+05 0.999587E+05


                          DESIGN COMBINATION:   EQUIPMENT 1:  1     EQUIPMENT 2:  1

FROM INIT2: INITIAL BASE POINT:              10     3    14    15     2    25
FROM INIT2: INITIAL PENALTY PARAMETERS:     1.0   1.0   1.0   1.0   1.0   1.0
FROM INIT2: INITIAL LAGRANGE MULTIPLIERS:   0.0   0.0   0.0   0.0   0.0   0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC       APRC        AOPC         ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   1       1     10  3 14 15  2 25  .2891937E+14 000010      290000.      420000.  0.301084E+08 0.700776E+07 -.760518E+07 0.999469E+05
   1       1     11  3 14 15  2 25  .3981703E+07 000000      280000.      320000.  0.479443E+08 0.700776E+07 0.372508E+06 0.999469E+05
   1       1     10  4 14 15  2 25  .2813247E+14 000010      290000.      330000.  0.301124E+08 0.700776E+07 -.750100E+07 0.999469E+05
   1       1     10  3 15 15  2 25  .4831768E+14 000010      296667.      420000.  0.284679E+08 0.700776E+07 -.983033E+07 0.999469E+05
   1       1     10  3 13 15  2 25  .1752401E+14 000010      282308.      420000.  0.315438E+08 0.700776E+07 -.592014E+07 0.999469E+05
   1       1     10  3 14 16  2 25  .2891937E+14 000010      276000.      350000.  0.301084E+08 0.992840E+07 -.760518E+07 0.999987E+05
   1       1     10  3 14 15  3 25  .2891937E+14 000010      290000.      290000.  0.301084E+08 0.700896E+07 -.760518E+07 0.999843E+05
   1       1     10  3 14 15  1 25  .2891937E+14 000010      290000.      550000.  0.301084E+08 0.687910E+07 -.760518E+07 0.991378E+05
   1       1     10  3 14 15  2 24  .2891937E+14 000010      281250.      420000.  0.301084E+08 0.720683E+07 -.760518E+07 0.999622E+05
   1       1     11  3 14 15  2 25  .3981703E+07 000000      280000.      320000.  0.479443E+08 0.700776E+07 0.372508E+06 0.999469E+05
   1       1     12  3 13 15  2 25  .4068576E+07 000000      260769.      220000.  0.498688E+08 0.700776E+07 0.965440E+06 0.999469E+05
   1       1     12  3 14 15  2 25  .4065843E+07 000000      270000.      220000.  0.498365E+08 0.700776E+07 0.950083E+06 0.999469E+05
   1       1     10  3 14 15  2 25  .2891937E+14 000010      290000.      420000.  0.301084E+08 0.700776E+07 -.760518E+07 0.999469E+05
   1       1     11  4 14 15  2 25  .4108538E+07 000000      280000.      230000.  0.479493E+08 0.700776E+07 0.535076E+06 0.999469E+05
   1       1     11  2 14 15  2 25  .9956858E+11 000010      280000.      410000.  0.478098E+08 0.700776E+07 -.446240E+06 0.999469E+05
   1       1     11  3 15 15  2 25  .3985041E+07 000000      287333.      320000.  0.476021E+08 0.700776E+07 0.145167E+06 0.999469E+05
   1       1     11  3 13 15  2 25  .3979659E+07 000000      271539.      320000.  0.482233E+08 0.700776E+07 0.531777E+06 0.999469E+05
   1       1     11  3 12 15  2 25  .3978127E+07 000000      261667.      320000.  0.484818E+08 0.700776E+07 0.658512E+06 0.999469E+05
   1       1     11  3 11 15  2 25  .3978106E+07 000000      250000.      320000.  0.486655E+08 0.700776E+07 0.736389E+06 0.999469E+05
   1       1     11  3  9 15  2 25  .3981859E+07 000000      218889.      320000.  0.488489E+08 0.700776E+07 0.804079E+06 0.999469E+05
   1       1     11  3 10 15  2 25  .3979687E+07 000000      236000.      320000.  0.487694E+08 0.700776E+07 0.775947E+06 0.999469E+05
   1       1     11  3 11 16  2 25  .3846808E+07 000000      236000.      250000.  0.486655E+08 0.992840E+07 0.736389E+06 0.999987E+05
   1       1     11  3 11 17  2 25  .3949635E+07 000000      222000.      180000.  0.486655E+08 0.999818E+07 0.736389E+06 0.999999E+05
   1       1     11  3 11 16  3 25  .4038685E+07 000000      236000.      120000.  0.486655E+08 0.992941E+07 0.736389E+06 0.999997E+05
   1       1     11  3 11 16  1 25  .3661232E+07 000000      236000.      380000.  0.486655E+08 0.985067E+07 0.736389E+06 0.999587E+05
   1       1     11  3 11 16  1 25  .4161232E+07 000000      236000.      380000.  0.486655E+08 0.985067E+07 0.736389E+06 0.999587E+05
   1       1     11  3 11 16  1 25  .4161232E+07 000000      236000.      380000.  0.486655E+08 0.985067E+07 0.736389E+06 0.999587E+05
   1       1     11  3 11 16  1 24  .3663433E+07 000000      226667.      380000.  0.486655E+08 0.987023E+07 0.736389E+06 0.999727E+05
   1       1     12  3 11 16  1 25  .3757970E+07 000000      223273.      280000.  0.499149E+08 0.985067E+07 0.983194E+06 0.999587E+05
   1       1     10  3 11 16  1 25  .6163214E+13 000010      248727.      480000.  0.340604E+08 0.985067E+07 -.351090E+07 0.999587E+05
   1       1     11  4 11 16  1 25  .3788119E+07 000000      236000.      290000.  0.486676E+08 0.985067E+07 0.804194E+06 0.999587E+05
   1       1     11  2 11 16  1 25  .3535563E+07 000000      236000.      470000.  0.485967E+08 0.985067E+07 0.395368E+06 0.999587E+05
   1       1     11  1 11 16  1 25  .8244067E+13 000010      236000.      560000.  0.467668E+08 0.985067E+07 -.406056E+07 0.999587E+05
   1       1     11  2 12 16  1 25  .3535862E+07 000000      247667.      470000.  0.483978E+08 0.985067E+07 0.215700E+06 0.999587E+05
   1       1     11  2 10 16  1 25  .3536996E+07 000000      222000.      470000.  0.487087E+08 0.985067E+07 0.486508E+06 0.999587E+05
   1       1     11  2 10 17  2 25  .3825399E+07 000000      208000.      270000.  0.487087E+08 0.999818E+07 0.486508E+06 0.999999E+05
   1       1     11  2 12 15  1 25  .4171343E+07 000000      261667.      540000.  0.483978E+08 0.687910E+07 0.215700E+06 0.991378E+05
   1       1     11  2 11 15  2 25  .3852437E+07 000000      250000.      410000.  0.485967E+08 0.700776E+07 0.395368E+06 0.999469E+05
   1       1     11  2 11 17  1 25  .4132480E+07 000000      222000.      400000.  0.485967E+08 0.999241E+07 0.395368E+06 0.999979E+05
   1       1     11  2 11 16  1 25  .4035563E+07 000000      236000.      470000.  0.485967E+08 0.985067E+07 0.395368E+06 0.999587E+05
   1       1     11  2 11 16  1 24  .3537764E+07 000000      226667.      470000.  0.485967E+08 0.987023E+07 0.395368E+06 0.999727E+05
```

```
   1        1      12  2 11 16  1 25  .3631942E+07 000000     223273.     370000.  0.498658E+08 0.985067E+07 0.942187E+06 0.999587E+05
   1        1      10  2 11 16  1 25  .1739465E+14 000010     248727.     570000.  0.339902E+08 0.985067E+07 -.589825E+07 0.999587E+05
   1        1      11  3 11 16  1 25  .3661232E+07 000000     236000.     380000.  0.486655E+08 0.985067E+07 0.736389E+06 0.999587E+05
   1        1      11  1 11 16  1 25  .8244067E+13 000010     236000.     560000.  0.467668E+08 0.985067E+07 -.406056E+07 0.999587E+05
   1        1      11  2 12 16  1 25  .3535862E+07 000000     247667.     470000.  0.483978E+08 0.985067E+07 0.215700E+06 0.999587E+05
   1        1      11  2 10 16  1 25  .3536996E+07 000000     222000.     470000.  0.487087E+08 0.985067E+07 0.486508E+06 0.999587E+05
   1        1      11  2 10 17  2 25  .3825399E+07 000000     208000.     270000.  0.487087E+08 0.999818E+07 0.486508E+06 0.999999E+05
   1        1      11  2 12 15  1 25  .4171343E+07 000000     261667.     540000.  0.483978E+08 0.687910E+07 0.215700E+06 0.991378E+05
   1        1      11  2 11 15  2 25  .3852437E+07 000000     250000.     410000.  0.485967E+08 0.700776E+07 0.395368E+06 0.999469E+05
   1        1      11  2 11 17  1 25  .4132480E+07 000000     222000.     400000.  0.485967E+08 0.999241E+07 0.395368E+06 0.999979E+05
   1        1      11  2 11 16  1 25  .4035563E+07 000000     236000.     470000.  0.485967E+08 0.985067E+07 0.395368E+06 0.999587E+05
   1        1      11  2 11 16  1 24  .3537764E+07 000000     226667.     470000.  0.485967E+08 0.987023E+07 0.395368E+06 0.999727E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                   11  2 11 16  1 25        3535563.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 12  1 11 17  1 25  FLAGC =000000
FROM LOCMIN: NEXT GRAD SEARCH POINT: 13  1 11 18  1 25  FLAGC =000000
FROM LOCMIN: THERE ARE  0 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  2 11 17  1 25    FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  13  2 11 18  1 25    FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  14  2 11 19  1 25    FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  15  2 11 20  1 25    FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  16  2 11 21  1 25    FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  17  2 11 22  1 25    FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  18  2 11 22  1 25    FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  19  2 11 23  1 25    FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  20  2 11 24  1 25    FLAG1= 2
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   1        1      11  2 11 16  1 25        3535563.           236000.     470000.  0.485967E+08 0.985067E+07 0.395368E+06 0.999587E+05

                                        DESIGN COMBINATION:   EQUIPMENT 1:  1      EQUIPMENT 2:  1

FROM INIT1: INITIAL BASE POINT:              14    1    8   18    1   17
FROM INIT1: INITIAL PENALTY PARAMETERS:     1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT1: INITIAL LAGRANGE MULTIPLIERS:   0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC        APRC        AOPC        ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   1        1      14  1  8 18  1 17  .1254980E+09 100000     -15588.     120000.  0.499845E+08 0.999998E+07 0.980548E+06 0.100000E+06
   1        1      15  1  8 18  1 17  .5515359E+09 100000     -33088.      20000.  0.499967E+08 0.999998E+07 0.995882E+06 0.100000E+06
   1        1      13  1  8 18  1 17  .3882849E+07 000000       1912.     220000.  0.499218E+08 0.999998E+07 0.901587E+06 0.100000E+06
   1        1      14  2  8 18  1 17  .1256246E+09 100000     -15588.      30000.  0.499991E+08 0.999998E+07 0.999706E+06 0.100000E+06
   1        1      14  1  9 18  1 17  .3997496E+07 000000      11634.     120000.  0.499808E+08 0.999998E+07 0.973837E+06 0.100000E+06
   1        1      14  1  8 19  1 17  .6584796E+09 100000     -36176.      50000.  0.499845E+08 0.100000E+08 0.980548E+06 0.100000E+06
   1        1      14  1  8 17  1 17  .3891424E+07 000000       5000.     190000.  0.499845E+08 0.999924E+07 0.980548E+06 0.999999E+05
   1        1      14  1  8 18  2 17  .1756898E+09 110000     -15588.     -10000.  0.499845E+08 0.100000E+08 0.980548E+06 0.100000E+06
   1        1      14  1  8 18  1 18  .3996047E+07 000000       5000.     120000.  0.499845E+08 0.999997E+07 0.980548E+06 0.100000E+06
   1        1      13  1  9 17  1 18  .3762038E+07 000000      67222.     290000.  0.499068E+08 0.999887E+07 0.873112E+06 0.999999E+05
   1        1      12  1 10 16  1 19  .3534103E+07 000000     137263.     460000.  0.494587E+08 0.995280E+07 0.214830E+06 0.999986E+05
   1        1      11  1 11 15  1 20  .8244067E+13 000010     197500.     630000.  0.467668E+08 0.815705E+07 -.406056E+07 0.999325E+05
   1        1      13  1 11 17  1 20  .3743467E+07 000000     137046.     290000.  0.498728E+08 0.999838E+07 0.800858E+06 0.999999E+05
   1        1      11  1  9 15  1 18  .3771212E+13 000010     137222.     630000.  0.472498E+08 0.836614E+07 -.274635E+07 0.999625E+05
   1        1      12  2 10 16  1 19  .3653311E+07 000000     137263.     370000.  0.498815E+08 0.995280E+07 0.952892E+06 0.999986E+05
   1        1      12  1 10 16  1 19  .4034103E+07 000000     137263.     460000.  0.494587E+08 0.995280E+07 0.214830E+06 0.999986E+05
   1        1      11  1 10 16  1 19  .5417998E+13 000010     151263.     560000.  0.470415E+08 0.995280E+07 -.329181E+07 0.999986E+05
   1        1      13  1 10 16  1 19  .3646039E+07 000000     123263.     360000.  0.498928E+08 0.995280E+07 0.844437E+06 0.999986E+05
   1        1      12  1  9 17  1 20  .3638228E+07 000000     115833.     390000.  0.495136E+08 0.999838E+07 0.337480E+06 0.999999E+05
   1        1      12  1 11 16  1 19  .3531682E+07 000000     152536.     460000.  0.493837E+08 0.995280E+07 0.353311E+05 0.999986E+05
   1        1      12  1 12 15  1 18  .5745079E+11 000010     168333.     530000.  0.492416E+08 0.836614E+07 -.338961E+06 0.999625E+05
   1        1      12  1 11 16  2 19  .3721658E+07 000000     152536.     330000.  0.493837E+08 0.997696E+07 0.353311E+05 0.999999E+05
   1        1      12  1 11 16  1 19  .4031682E+07 000000     152536.     460000.  0.493837E+08 0.995280E+07 0.353311E+05 0.999986E+05
   1        1      12  1 11 15  1 20  .3564297E+07 000000     184773.     530000.  0.493837E+08 0.815705E+07 0.353311E+05 0.999325E+05
   1        1      12  1 11 17  1 18  .3643654E+07 000000     116717.     390000.  0.493837E+08 0.999887E+07 0.353311E+05 0.999999E+05
   1        1      13  1 12 17  1 20  .3740260E+07 000000     150833.     290000.  0.498328E+08 0.999838E+07 0.704910E+06 0.999999E+05
   1        1      11  1 10 15  1 18  .5417998E+13 000010     154333.     630000.  0.470415E+08 0.836614E+07 -.329181E+07 0.999625E+05
   1        1      12  2 11 16  1 19  .3649806E+07 000000     152536.     370000.  0.498658E+08 0.995280E+07 0.942187E+06 0.999986E+05
   1        1      12  1 11 16  1 19  .4031682E+07 000000     152536.     460000.  0.493837E+08 0.995280E+07 0.353311E+05 0.999986E+05
   1        1      11  1 11 16  1 19  .8244067E+13 000010     165263.     560000.  0.467668E+08 0.995280E+07 -.406056E+07 0.999986E+05
   1        1      13  1 11 16  1 19  .3642295E+07 000000     139809.     360000.  0.498728E+08 0.995280E+07 0.800858E+06 0.999986E+05
   1        1      12  1 10 16  1 19  .3534103E+07 000000     137263.     460000.  0.494587E+08 0.995280E+07 0.214830E+06 0.999986E+05
```

```
   1        1      12  1 12 16  1 19  .5745077E+11 000010     165263.     460000.  0.492416E+08 0.995280E+07 -.338961E+06 0.999986E+05
   1        1      12  1 11 16  2 19  .3721658E+07 000000     152536.     330000.  0.493837E+08 0.997696E+07 0.353311E+05 0.999999E+05
   1        1      12  1 11 16  1 19  .4031682E+07 000000     152536.     460000.  0.493837E+08 0.995280E+07 0.353311E+05 0.999986E+05
   1        1      12  1 11 15  1 20  .3564297E+07 000000     184773.     530000.  0.493837E+08 0.815705E+07 0.353311E+05 0.999325E+05
   1        1      12  1 11 17  1 18  .3643654E+07 000000     116717.     390000.  0.493837E+08 0.999887E+07 0.353311E+05 0.999999E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                   12  1 11 16  1 19        3531682.
FROM LOCMIN: RETURN 2
   1        1      12  1 11 16  1 20  .3527184E+07 000000     167273.     460000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
   1        1      13  1 11 16  1 20  .3637797E+07 000000     154546.     360000.  0.498728E+08 0.994702E+07 0.800858E+06 0.999981E+05
   1        1      11  1 11 16  1 20  .8244067E+13 000010     180000.     560000.  0.467668E+08 0.994702E+07 -.406056E+07 0.999981E+05
   1        1      12  2 11 16  1 20  .3645308E+07 000000     167273.     370000.  0.498658E+08 0.994702E+07 0.942187E+06 0.999981E+05
   1        1      12  1 12 16  1 20  .5745077E+11 000010     180000.     460000.  0.492416E+08 0.994702E+07 -.338961E+06 0.999981E+05
   1        1      12  1 10 16  1 20  .3529605E+07 000000     152000.     460000.  0.494587E+08 0.994702E+07 0.214830E+06 0.999981E+05
   1        1      12  1 11 17  1 20  .3632854E+07 000000     149773.     390000.  0.493837E+08 0.999838E+07 0.353311E+05 0.999999E+05
   1        1      12  1 11 15  1 20  .3564297E+07 000000     184773.     530000.  0.493837E+08 0.815705E+07 0.353311E+05 0.999325E+05
   1        1      12  1 11 16  2 20  .3716915E+07 000000     167273.     330000.  0.493837E+08 0.997417E+07 0.353311E+05 0.999999E+05
   1        1      12  1 11 16  1 21  .3524052E+07 000000     180606.     460000.  0.493837E+08 0.992763E+07 0.353311E+05 0.999951E+05
   1        1      12  1 11 16  1 21  .3524052E+07 000000     180606.     460000.  0.493837E+08 0.992763E+07 0.353311E+05 0.999951E+05
   1        1      12  1 11 16  1 22  .3520777E+07 000000     192727.     460000.  0.493837E+08 0.991297E+07 0.353311E+05 0.999916E+05
   1        1      12  1 11 16  1 23  .3517848E+07 000000     203795.     460000.  0.493837E+08 0.989693E+07 0.353311E+05 0.999861E+05
   1        1      12  1 11 16  1 25  .3513818E+07 000000     223273.     460000.  0.493837E+08 0.985067E+07 0.353311E+05 0.999587E+05
   1        1      12  1 11 16  1 25  .8013818E+07 000000     223273.     460000.  0.493837E+08 0.985067E+07 0.353311E+05 0.999587E+05
   1        1      12  1 11 16  1 25  .4013818E+07 000000     223273.     460000.  0.493837E+08 0.985067E+07 0.353311E+05 0.999587E+05
   1        1      12  1 11 16  1 24  .3516019E+07 000000     213939.     460000.  0.493837E+08 0.987023E+07 0.353311E+05 0.999727E+05
   1        1      13  1 11 16  1 25  .3624431E+07 000000     210546.     360000.  0.498728E+08 0.985067E+07 0.800858E+06 0.999587E+05
   1        1      11  1 11 16  1 25  .8244067E+13 000010     236000.     560000.  0.467668E+08 0.985067E+07 -.406056E+07 0.999587E+05
   1        1      12  2 11 16  1 25  .3631942E+07 000000     223273.     370000.  0.498658E+08 0.985067E+07 0.942187E+06 0.999587E+05
   1        1      12  1 11 16  1 25  .4013818E+07 000000     223273.     460000.  0.493837E+08 0.985067E+07 0.353311E+05 0.999587E+05
   1        1      12  1 12 16  1 25  .5745075E+11 000010     236000.     460000.  0.492416E+08 0.985067E+07 -.338961E+06 0.999587E+05
   1        1      12  1 10 16  1 25  .3516239E+07 000000     208000.     460000.  0.494587E+08 0.985067E+07 0.214830E+06 0.999587E+05
   1        1      12  1 11 17  1 25  .3610735E+07 000000     209273.     390000.  0.493837E+08 0.999241E+07 0.353311E+05 0.999979E+05
   1        1      12  1 11 15  1 25  .3649299E+07 000000     237273.     530000.  0.493837E+08 0.687910E+07 0.353311E+05 0.991378E+05
   1        1      12  1 11 16  2 25  .3699394E+07 000000     223273.     330000.  0.493837E+08 0.992840E+07 0.353311E+05 0.999987E+05
   1        1      12  1 11 16  1 25  .4013818E+07 000000     223273.     460000.  0.493837E+08 0.985067E+07 0.353311E+05 0.999587E+05
   1        1      12  1 11 16  1 25  .4013818E+07 000000     223273.     460000.  0.493837E+08 0.985067E+07 0.353311E+05 0.999587E+05
   1        1      12  1 11 16  1 24  .3516019E+07 000000     213939.     460000.  0.493837E+08 0.987023E+07 0.353311E+05 0.999727E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                   12  1 11 16  1 25        3513818.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 11 17  1 25  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  1 11 17  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 11 18  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 11 19  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 10 20  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 10 21  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 10 22  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 23  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 24  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 25  1 25   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   1        1      12  1 11 16  1 25        3513818.          223273.     460000.  0.493837E+08 0.985067E+07 0.353311E+05 0.999587E+05


                              DESIGN COMBINATION:   EQUIPMENT 1:  1      EQUIPMENT 2:  1

FROM INIT2: INITIAL BASE POINT:              11    2   14   15    2   25
FROM INIT2: INITIAL PENALTY PARAMETERS:     1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT2: INITIAL LAGRANGE MULTIPLIERS:   0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC       APRC        AOPC          ASUNA(1)     ASUNA(2)      ACUNA(1)     ACUNA(2)
   1        1      11  2 14 15  2 25  .9956858E+11 000010     280000.     410000.  0.478098E+08 0.700776E+07 -.446240E+06 0.999469E+05
   1        1      12  2 14 15  2 25  .3940703E+07 000000     270000.     310000.  0.497387E+08 0.700776E+07 0.827465E+06 0.999469E+05
   1        1      11  3 14 15  2 25  .3981703E+07 000000     280000.     320000.  0.479443E+08 0.700776E+07 0.372508E+06 0.999469E+05
   1        1      11  2 15 15  2 25  .4745071E+12 000010     287333.     410000.  0.474311E+08 0.700776E+07 -.974170E+06 0.999469E+05
   1        1      11  2 13 15  2 25  .2988484E+10 000010     271539.     410000.  0.481161E+08 0.700776E+07 -.772610E+05 0.999469E+05
   1        1      11  2 14 16  2 25  .9956852E+11 000010     266000.     340000.  0.478098E+08 0.992840E+07 -.446240E+06 0.999987E+05
   1        1      11  2 14 15  3 25  .9956878E+11 000010     280000.     280000.  0.478098E+08 0.700896E+07 -.446240E+06 0.999843E+05
   1        1      11  2 14 15  1 25  .9956839E+11 000010     280000.     540000.  0.478098E+08 0.687910E+07 -.446240E+06 0.991378E+05
```

```
   1       1      11  2 14 15  2 24  .9956858E+11 000010      271250.     410000.  0.478098E+08 0.720683E+07 -.446240E+06 0.999622E+05
   1       1      12  3 13 15  2 25  .4068576E+07 000000      260769.     220000.  0.498688E+08 0.700776E+07 0.965440E+06 0.999469E+05
   1       1      13  2 14 15  2 25  .4055221E+07 000000      260000.     210000.  0.499662E+08 0.700776E+07 0.977703E+06 0.999469E+05
   1       1      11  2 14 15  2 25  .9956858E+11 000010      280000.     410000.  0.478098E+08 0.700776E+07 -.446240E+06 0.999469E+05
   1       1      12  3 14 15  2 25  .4065843E+07 000000      270000.     220000.  0.498365E+08 0.700776E+07 0.950083E+06 0.999469E+05
   1       1      12  1 14 15  2 25  .1795590E+13 000010      270000.     400000.  0.487604E+08 0.700776E+07 -.189504E+07 0.999469E+05
   1       1      12  2 15 15  2 25  .3938814E+07 000000      278000.     310000.  0.496683E+08 0.700776E+07 0.745090E+06 0.999469E+05
   1       1      12  2 15 15  2 25  .4438814E+07 000000      278000.     310000.  0.496683E+08 0.700776E+07 0.745090E+06 0.999469E+05
   1       1      12  2 15 16  2 25  .3807516E+07 000000      264000.     240000.  0.496683E+08 0.992840E+07 0.745090E+06 0.999987E+05
   1       1      12  2 15 17  2 25  .3910343E+07 000000      250000.     170000.  0.496683E+08 0.999818E+07 0.745090E+06 0.999999E+05
   1       1      12  2 15 16  3 25  .3999393E+07 000000      264000.     110000.  0.496683E+08 0.992941E+07 0.745090E+06 0.999997E+05
   1       1      12  2 15 16  1 25  .3621940E+07 000000      264000.     370000.  0.496683E+08 0.985067E+07 0.745090E+06 0.999587E+05
   1       1      12  2 15 16  1 25  .4121940E+07 000000      264000.     370000.  0.496683E+08 0.985067E+07 0.745090E+06 0.999587E+05
   1       1      12  2 15 16  1 25  .4121940E+07 000000      264000.     370000.  0.496683E+08 0.985067E+07 0.745090E+06 0.999587E+05
   1       1      12  2 15 16  1 24  .3624141E+07 000000      254667.     370000.  0.496683E+08 0.987023E+07 0.745090E+06 0.999727E+05
   1       1      13  2 15 16  1 25  .3735140E+07 000000      254667.     270000.  0.499536E+08 0.985067E+07 0.964344E+06 0.999587E+05
   1       1      11  2 15 16  1 25  .4745069E+12 000010      273333.     470000.  0.474311E+08 0.985067E+07 -.974170E+06 0.999587E+05
   1       1      12  3 15 16  1 25  .3746579E+07 000000      264000.     280000.  0.497935E+08 0.985067E+07 0.926395E+06 0.999587E+05
   1       1      12  1 15 16  1 25  .5393732E+13 000010      264000.     460000.  0.484056E+08 0.985067E+07 -.328443E+07 0.999587E+05
   1       1      12  2 15 17  2 25  .4410343E+07 000000      250000.     170000.  0.496683E+08 0.999818E+07 0.745090E+06 0.999999E+05
   1       1      12  2 14 15  1 25  .4259310E+07 000000      270000.     440000.  0.497387E+08 0.687910E+07 0.827465E+06 0.991378E+05
   1       1      12  2 14 16  1 25  .3623829E+07 000000      256000.     370000.  0.497387E+08 0.985067E+07 0.827465E+06 0.999587E+05
   1       1      12  2 15 16  1 25  .4121940E+07 000000      264000.     370000.  0.496683E+08 0.985067E+07 0.745090E+06 0.999587E+05
   1       1      12  2 15 15  2 25  .3938814E+07 000000      278000.     310000.  0.496683E+08 0.700776E+07 0.745090E+06 0.999469E+05
   1       1      12  2 15 17  1 25  .4218857E+07 000000      250000.     300000.  0.496683E+08 0.999241E+07 0.745090E+06 0.999979E+05
   1       1      12  2 15 16  1 25  .4121940E+07 000000      264000.     370000.  0.496683E+08 0.985067E+07 0.745090E+06 0.999587E+05
   1       1      12  2 15 16  1 24  .3624141E+07 000000      254667.     370000.  0.496683E+08 0.987023E+07 0.745090E+06 0.999727E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                  12  2 15 16  1 25          3621940.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 15 17  1 25  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  2 15 17  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  2 14 18  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 14 19  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 14 20  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 13 21  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 13 22  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 13 23  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 12 24  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 12 25  1 25   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   1       1      12  2 15 16  1 25      3621940.         264000.     370000.  0.496683E+08 0.985067E+07 0.745090E+06 0.999587E+05


                              DESIGN COMBINATION:   EQUIPMENT 1:  1      EQUIPMENT 2:  2

FROM INIT: INITIAL BASE POINT:               11    2    11    16    2    23
FROM INIT: INITIAL PENALTY PARAMETERS:      1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT: INITIAL LAGRANGE MULTIPLIERS:   0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC        APRC         AOPC         ASUNA(1)      ASUNA(2)      ACUNA(1)      ACUNA(2)
   1       2      11  2 11 16  2 23  .3755883E+07 000000      167826.     420000.  0.485967E+08 0.999034E+07 0.395368E+06 0.100000E+06
   1       2      12  2 11 16  2 23  .3852262E+07 000000      155099.     320000.  0.498658E+08 0.999034E+07 0.942187E+06 0.100000E+06
   1       2      10  2 11 16  2 23  .1739465E+14 000010      180553.     520000.  0.339902E+08 0.999034E+07 -.589825E+07 0.100000E+06
   1       2      11  3 11 16  2 23  .3881552E+07 000000      167826.     330000.  0.486655E+08 0.999034E+07 0.736389E+06 0.100000E+06
   1       2      11  1 11 16  2 23  .8244067E+13 000010      167826.     510000.  0.467668E+08 0.999034E+07 -.406056E+07 0.100000E+06
   1       2      11  2 12 16  2 23  .3756182E+07 000000      179493.     420000.  0.483978E+08 0.999034E+07 0.215700E+06 0.100000E+06
   1       2      11  2 10 16  2 23  .3757316E+07 000000      153826.     420000.  0.487087E+08 0.999034E+07 0.486508E+06 0.100000E+06
   1       2      11  2 11 17  2 23  .3866145E+07 000000      149565.     355000.  0.485967E+08 0.999991E+07 0.395368E+06 0.100000E+06
   1       2      11  2 11 15  2 23  .3734284E+07 000000      186087.     485000.  0.485967E+08 0.890115E+07 0.395368E+06 0.999996E+05
   1       2      11  2 11 16  3 23  .3947839E+07 000000      167826.     290000.  0.485967E+08 0.999039E+07 0.395368E+06 0.100000E+06
   1       2      11  2 11 16  1 23  .3564741E+07 000000      167826.     550000.  0.485967E+08 0.998038E+07 0.395368E+06 0.999999E+05
   1       2      11  2 11 16  2 24  .3752010E+07 000000      180000.     420000.  0.485967E+08 0.998851E+07 0.395368E+06 0.999999E+05
   1       2      11  2 11 16  1 23  .3564741E+07 000000      167826.     550000.  0.485967E+08 0.998038E+07 0.395368E+06 0.999999E+05
   1       2      11  2 11 16  1 23  .4064741E+07 000000      167826.     550000.  0.485967E+08 0.998038E+07 0.395368E+06 0.999999E+05
   1       2      12  2 11 16  1 23  .3661120E+07 000000      155099.     450000.  0.498658E+08 0.998038E+07 0.942187E+06 0.999999E+05
   1       2      10  2 11 16  1 23  .1739465E+14 000010      180553.     650000.  0.339902E+08 0.998038E+07 -.589825E+07 0.999999E+05
   1       2      11  3 11 16  1 23  .3690410E+07 000000      167826.     460000.  0.486655E+08 0.998038E+07 0.736389E+06 0.999999E+05
   1       2      11  1 11 16  1 23  .8244067E+13 000010      167826.     640000.  0.467668E+08 0.998038E+07 -.406056E+07 0.999999E+05
```

```
  1       2      11  2 12 16  1 23  .3565040E+07 000000     179493.     550000.  0.483978E+08 0.998038E+07 0.215700E+06 0.999999E+05
  1       2      11  2 10 16  1 23  .3566174E+07 000000     153826.     550000.  0.487087E+08 0.998038E+07 0.486508E+06 0.999999E+05
  1       2      11  2 11 17  1 23  .3674208E+07 000000     149565.     485000.  0.485967E+08 0.999963E+07 0.395368E+06 0.100000E+06
  1       2      11  2 11 15  1 23  .3543725E+07 000000     186087.     615000.  0.485967E+08 0.888408E+07 0.395368E+06 0.999942E+05
  1       2      11  2 11 15  1 23  .4043725E+07 000000     186087.     615000.  0.485967E+08 0.888408E+07 0.395368E+06 0.999942E+05
  1       2      11  2 11 15  2 23  .3734284E+07 000000     186087.     485000.  0.485967E+08 0.890115E+07 0.395368E+06 0.999996E+05
  1       2      11  2 11 15  1 23  .4043725E+07 000000     186087.     615000.  0.485967E+08 0.888408E+07 0.395368E+06 0.999942E+05
  1       2      11  2 11 15  1 24  .3548417E+07 000000     197500.     615000.  0.485967E+08 0.878104E+07 0.395368E+06 0.999911E+05
  1       2      11  2 11 15  1 22  .3547705E+07 000000     173636.     615000.  0.485967E+08 0.888408E+07 0.395368E+06 0.999942E+05
  1       2      12  2 11 15  1 23  .3640104E+07 000000     173360.     515000.  0.498658E+08 0.888408E+07 0.942187E+06 0.999942E+05
  1       2      10  2 11 15  1 23  .1739465E+14 000010     198814.     715000.  0.339902E+08 0.888408E+07 -.589825E+07 0.999942E+05
  1       2      11  3 11 15  1 23  .3669394E+07 000000     186087.     525000.  0.486655E+08 0.888408E+07 0.736389E+06 0.999942E+05
  1       2      11  1 11 15  1 23  .8244067E+13 000010     186087.     705000.  0.467668E+08 0.888408E+07 -.406056E+07 0.999942E+05
  1       2      11  2 12 15  1 23  .3544024E+07 000000     197754.     615000.  0.483978E+08 0.888408E+07 0.215700E+06 0.999942E+05
  1       2      11  2 10 15  1 23  .3545158E+07 000000     172087.     615000.  0.487087E+08 0.888408E+07 0.486508E+06 0.999942E+05
  1       2      11  2 11 16  1 23  .3564741E+07 000000     167826.     550000.  0.485967E+08 0.998038E+07 0.395368E+06 0.999999E+05
  1       2      11  2 11 15  1 23  .4043725E+07 000000     186087.     615000.  0.485967E+08 0.888408E+07 0.395368E+06 0.999942E+05
  1       2      11  2 11 15  2 23  .3734284E+07 000000     186087.     485000.  0.485967E+08 0.890115E+07 0.395368E+06 0.999996E+05
  1       2      11  2 11 15  1 23  .4043725E+07 000000     186087.     615000.  0.485967E+08 0.888408E+07 0.395368E+06 0.999942E+05
  1       2      11  2 11 15  1 24  .3548417E+07 000000     197500.     615000.  0.485967E+08 0.878104E+07 0.395368E+06 0.999911E+05
  1       2      11  2 11 15  1 22  .3547705E+07 000000     173636.     615000.  0.485967E+08 0.888408E+07 0.395368E+06 0.999942E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                 11  2 11 15  1 23        3543725.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 12  1 11 15  1 23  FLAGC =000000
FROM LOCMIN: RETURN 5
  1       2      12  1 11 15  1 23  .3521980E+07 000000     173360.     605000.  0.493837E+08 0.888408E+07 0.353311E+05 0.999942E+05
  1       2      13  1 11 15  1 23  .3632593E+07 000000     160633.     505000.  0.498728E+08 0.888408E+07 0.800858E+06 0.999942E+05
  1       2      11  1 11 15  1 23  .8244067E+13 000010     186087.     705000.  0.467668E+08 0.888408E+07 -.406056E+07 0.999942E+05
  1       2      12  2 11 15  1 23  .3640104E+07 000000     173360.     515000.  0.498658E+08 0.888408E+07 0.942187E+06 0.999942E+05
  1       2      12  1 12 15  1 23  .5745076E+11 000010     186087.     605000.  0.492416E+08 0.888408E+07 -.338961E+06 0.999942E+05
  1       2      12  1 10 15  1 23  .3524401E+07 000000     158087.     605000.  0.494587E+08 0.888408E+07 0.214830E+06 0.999942E+05
  1       2      12  1 11 16  1 23  .3542996E+07 000000     155099.     540000.  0.493837E+08 0.998038E+07 0.353311E+05 0.999999E+05
  1       2      12  1 11 15  2 23  .3712539E+07 000000     173360.     475000.  0.493837E+08 0.890115E+07 0.353311E+05 0.999996E+05
  1       2      12  1 11 15  1 24  .3526672E+07 000000     184773.     605000.  0.493837E+08 0.878104E+07 0.353311E+05 0.999911E+05
  1       2      12  1 11 15  1 22  .3525960E+07 000000     160909.     605000.  0.493837E+08 0.888408E+07 0.353311E+05 0.999942E+05
FROM INNER: RESOLUTION VALLEY OR LOCAL MINIMUM ENCOUNTERED IN GRADIENT APPOXIMATION SECTION - RETURN TO DUNLOP.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 11 15  1 23  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 11 15  1 23   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 10 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 10 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5 10 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5  9 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  6  9 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  7  9 15  1 22   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  7  8 15  1 22   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
  1       2      12  1 11 15  1 23        3521980.          173360.     605000.  0.493837E+08 0.888408E+07 0.353311E+05 0.999942E+05


                                  DESIGN COMBINATION:   EQUIPMENT 1:  1      EQUIPMENT 2:  2

FROM INIT1: INITIAL BASE POINT:                14    1    8   17    1   16
FROM INIT1: INITIAL PENALTY PARAMETERS:       1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT1: INITIAL LAGRANGE MULTIPLIERS:  0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC        APRC       AOPC        ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
  1       2      14  1  8 17  1 16  .4167212E+10 100000      -91250.     275000.  0.499845E+08 0.999980E+07 0.980548E+06 0.100000E+06
  1       2      15  1  8 17  1 16  .5917319E+10 100000     -108750.     175000.  0.499967E+08 0.999980E+07 0.995882E+06 0.100000E+06
  1       2      13  1  8 17  1 16  .2723342E+10 100000      -73750.     375000.  0.499218E+08 0.999980E+07 0.901587E+06 0.100000E+06
  1       2      14  2  8 17  1 16  .4167338E+10 100000      -91250.     185000.  0.499991E+08 0.999980E+07 0.999706E+06 0.100000E+06
  1       2      14  1  9 17  1 16  .2053702E+10 100000      -64028.     275000.  0.499808E+08 0.999980E+07 0.973837E+06 0.100000E+06
  1       2      14  1  8 18  1 16  .6907154E+10 100000     -117500.     210000.  0.499845E+08 0.100000E+08 0.980548E+06 0.100000E+06
  1       2      14  1  8 16  1 16  .2116318E+10 100000      -65000.     340000.  0.499845E+08 0.998690E+07 0.980548E+06 0.999999E+05
  1       2      14  1  8 17  2 16  .4167404E+10 100000      -91250.     145000.  0.499845E+08 0.999995E+07 0.980548E+06 0.100000E+06
  1       2      14  1  8 17  1 17  .2116425E+10 100000      -65000.     275000.  0.499845E+08 0.999980E+07 0.980548E+06 0.100000E+06
  1       2      13  1  9 16  1 17  .3689993E+07 000000        2484.     440000.  0.499068E+08 0.998690E+07 0.873112E+06 0.999999E+05
  1       2      13  1 10 15  1 18  .3641650E+07 000000       68000.     505000.  0.498928E+08 0.908937E+07 0.844437E+06 0.999979E+05
```

```
   1       2      12  1 11 15  1 19  .4030810E+07 000000    115694.     605000.  0.493837E+08 0.898686E+07 0.353311E+05 0.999964E+05
   1       2      14  1 11 16  1 19  .3789483E+07 000000     68134.     340000.  0.499717E+08 0.998381E+07 0.955745E+06 0.999999E+05
   1       2      13  1  9 15  1 17  .4150833E+07 000000     27190.     505000.  0.499068E+08 0.908937E+07 0.873112E+06 0.999979E+05
   1       2      13  2 10 15  1 18  .3766831E+07 000000     68000.     415000.  0.499882E+08 0.908937E+07 0.995318E+06 0.999979E+05
   1       2      13  1 10 15  1 18  .4141650E+07 000000     68000.     505000.  0.498928E+08 0.908937E+07 0.844437E+06 0.999979E+05
   1       2      12  1 10 15  1 18  .3529714E+07 000000     82000.     605000.  0.494587E+08 0.908937E+07 0.214830E+06 0.999979E+05
   1       2      11  1 10 15  1 18  .5417998E+13 000010     96000.     705000.  0.470415E+08 0.908937E+07 -.329181E+07 0.999979E+05
   1       2      12  1  9 15  1 18  .3532667E+07 000000     63333.     605000.  0.495136E+08 0.908937E+07 0.337480E+06 0.999979E+05
   1       2      12  1 11 15  1 18  .3527293E+07 000000     97273.     605000.  0.493837E+08 0.908937E+07 0.353311E+05 0.999979E+05
   1       2      12  1 12 15  1 17  .5745126E+11 000010     89412.     605000.  0.492416E+08 0.908937E+07 -.338961E+06 0.999979E+05
   1       2      12  1 11 15  2 18  .3718316E+07 000000     97273.     475000.  0.493837E+08 0.910078E+07 0.353311E+05 0.999998E+05
   1       2      12  1 11 15  1 18  .4027293E+07 000000     97273.     605000.  0.493837E+08 0.908937E+07 0.353311E+05 0.999979E+05
   1       2      12  1 11 15  1 19  .4030810E+07 000000    115694.     605000.  0.493837E+08 0.898686E+07 0.353311E+05 0.999964E+05
   1       2      12  1 11 16  1 17  .3571609E+07 000000     51979.     540000.  0.493837E+08 0.998690E+07 0.353311E+05 0.999999E+05
   1       2      13  1 12 16  1 19  .3668541E+07 000000     94649.     440000.  0.498328E+08 0.998381E+07 0.704910E+06 0.999999E+05
   1       2      12  1 10 15  1 17  .4034870E+07 000000     61412.     605000.  0.494587E+08 0.908937E+07 0.214830E+06 0.999979E+05
   1       2      12  2 11 15  1 18  .3645417E+07 000000     97273.     515000.  0.498658E+08 0.908937E+07 0.942187E+06 0.999979E+05
   1       2      12  1 11 15  1 18  .4027293E+07 000000     97273.     605000.  0.493837E+08 0.908937E+07 0.353311E+05 0.999979E+05
   1       2      11  1 10 16  1 19  .5417998E+13 000010     92316.     640000.  0.470415E+08 0.998381E+07 -.329181E+07 0.999999E+05
   1       2      13  1 12 15  1 17  .4139855E+07 000000     77745.     505000.  0.498328E+08 0.908937E+07 0.704910E+06 0.999979E+05
   1       2      13  1 10 15  1 18  .3641650E+07 000000     68000.     505000.  0.498928E+08 0.908937E+07 0.844437E+06 0.999979E+05
   1       2      11  1 12 15  1 18  .1556254E+14 000010    121667.     705000.  0.462734E+08 0.908937E+07 -.557899E+07 0.999979E+05
   1       2      12  1 11 15  2 18  .3718316E+07 000000     97273.     475000.  0.493837E+08 0.910078E+07 0.353311E+05 0.999998E+05
   1       2      12  1 11 15  1 18  .4027293E+07 000000     97273.     605000.  0.493837E+08 0.908937E+07 0.353311E+05 0.999979E+05
   1       2      12  1 11 15  1 19  .4030810E+07 000000    115694.     605000.  0.493837E+08 0.898686E+07 0.353311E+05 0.999964E+05
   1       2      12  1 11 16  1 17  .3571609E+07 000000     51979.     540000.  0.493837E+08 0.998690E+07 0.353311E+05 0.999999E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                  12  1 11 15  1 18       3527293.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11   1 11 15  1 18  FLAGC =000010
FROM LOCMIN: THERE ARE   1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE   1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 11 15  1 18   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 10 15  1 18   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 10 15  1 18   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 15  1 18   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5 10 15  1 18   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5  9 15  1 18   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  6  9 15  1 19   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  7  9 15  1 19   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  7  8 15  1 19   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   1       2      12  1 11 15  1 18       3527293.             97273.     605000.  0.493837E+08 0.908937E+07 0.353311E+05 0.999979E+05


                              DESIGN COMBINATION:   EQUIPMENT 1:  1      EQUIPMENT 2:  2

FROM INIT2: INITIAL BASE POINT:              11    2   14   15    2   30
FROM INIT2: INITIAL PENALTY PARAMETERS:    1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT2: INITIAL LAGRANGE MULTIPLIERS:  0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2    TOTAL COST  FLAGC         APRC        AOPC        ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   1       2      11  2 14 15  2 30  .9956858E+11 000010    280000.     485000.  0.478098E+08 0.820297E+07 -.446240E+06 0.999958E+05
   1       2      12  2 14 15  2 30  .3857462E+07 000000    270000.     385000.  0.497387E+08 0.820297E+07 0.827465E+06 0.999958E+05
   1       2      11  3 14 15  2 30  .3898462E+07 000000    280000.     395000.  0.479443E+08 0.820297E+07 0.372508E+06 0.999958E+05
   1       2      11  2 15 15  2 30  .4745071E+12 000010    287333.     485000.  0.474311E+08 0.820297E+07 -.974170E+06 0.999958E+05
   1       2      11  2 13 15  2 30  .2988401E+10 000010    271539.     485000.  0.481161E+08 0.820297E+07 -.772610E+05 0.999958E+05
   1       2      11  2 14 16  2 30  .9956852E+11 000010    266000.     420000.  0.478098E+08 0.997417E+07 -.446240E+06 0.999999E+05
   1       2      11  2 14 15  3 30  .9956878E+11 000010    280000.     355000.  0.478098E+08 0.820324E+07 -.446240E+06 0.999987E+05
   1       2      11  2 14 15  1 30  .9956839E+11 000010    280000.     615000.  0.478098E+08 0.815705E+07 -.446240E+06 0.999325E+05
   1       2      11  2 14 15  2 29  .9956858E+11 000010    272759.     485000.  0.478098E+08 0.830266E+07 -.446240E+06 0.999968E+05
   1       2      12  3 13 15  2 30  .3985335E+07 000000    260769.     295000.  0.498688E+08 0.820297E+07 0.965440E+06 0.999958E+05
   1       2      13  2 14 15  2 30  .3971980E+07 000000    260000.     285000.  0.499662E+08 0.820297E+07 0.977703E+06 0.999958E+05
   1       2      11  2 14 15  2 30  .9956858E+11 000010    280000.     485000.  0.478098E+08 0.820297E+07 -.446240E+06 0.999958E+05
   1       2      12  3 14 15  2 30  .3982602E+07 000000    270000.     295000.  0.498365E+08 0.820297E+07 0.950083E+06 0.999958E+05
   1       2      12  1 14 15  2 30  .1795590E+13 000010    270000.     475000.  0.487604E+08 0.820297E+07 -.189504E+07 0.999958E+05
   1       2      12  2 15 15  2 30  .3855573E+07 000000    278000.     385000.  0.496683E+08 0.820297E+07 0.745090E+06 0.999958E+05
   1       2      12  2 15 15  2 30  .4355573E+07 000000    278000.     385000.  0.496683E+08 0.820297E+07 0.745090E+06 0.999958E+05
   1       2      12  2 15 16  2 30  .3819667E+07 000000    264000.     320000.  0.496683E+08 0.997417E+07 0.745090E+06 0.999999E+05
   1       2      12  2 15 17  2 30  .3927131E+07 000000    250000.     255000.  0.496683E+08 0.999961E+07 0.745090E+06 0.100000E+06
   1       2      12  2 15 16  3 30  .4011609E+07 000000    264000.     190000.  0.496683E+08 0.997439E+07 0.745090E+06 0.999999E+05
```

```
  1    2    12  2 15 16  1 30  .3629936E+07 000000    264000.    450000.  0.496683E+08 0.994702E+07 0.745090E+06 0.999981E+05
  1    2    12  2 15 16  1 30  .4129936E+07 000000    264000.    450000.  0.496683E+08 0.994702E+07 0.745090E+06 0.999981E+05
  1    2    12  2 15 16  1 30  .4129936E+07 000000    264000.    450000.  0.496683E+08 0.994702E+07 0.745090E+06 0.999981E+05
  1    2    12  2 15 16  1 29  .3632317E+07 000000    256276.    450000.  0.496683E+08 0.995280E+07 0.745090E+06 0.999986E+05
  1    2    13  2 15 16  1 30  .3743136E+07 000000    254667.    350000.  0.499536E+08 0.994702E+07 0.964344E+06 0.999981E+05
  1    2    11  2 15 16  1 30  .4745069E+12 000010    273333.    550000.  0.474311E+08 0.994702E+07 -.974170E+06 0.999981E+05
  1    2    12  3 15 16  1 30  .3754575E+07 000000    264000.    360000.  0.497935E+08 0.994702E+07 0.926395E+06 0.999981E+05
  1    2    12  1 15 16  1 30  .5393732E+13 000010    264000.    540000.  0.484056E+08 0.994702E+07 -.328443E+07 0.999981E+05
  1    2    12  2 15 17  2 30  .4427131E+07 000000    250000.    255000.  0.496683E+08 0.999961E+07 0.745090E+06 0.100000E+06
  1    2    12  2 14 15  1 30  .4169273E+07 000000    270000.    515000.  0.497387E+08 0.815705E+07 0.827465E+06 0.999325E+05
  1    2    12  2 14 16  1 30  .3631825E+07 000000    256000.    450000.  0.497387E+08 0.994702E+07 0.827465E+06 0.999981E+05
  1    2    12  2 15 16  1 30  .4129936E+07 000000    264000.    450000.  0.496683E+08 0.994702E+07 0.745090E+06 0.999981E+05
  1    2    12  2 15 15  2 30  .3855573E+07 000000    278000.    385000.  0.496683E+08 0.820297E+07 0.745090E+06 0.999958E+05
  1    2    12  2 15 17  1 30  .4235271E+07 000000    250000.    385000.  0.496683E+08 0.999838E+07 0.745090E+06 0.999999E+05
  1    2    12  2 15 16  1 30  .4129936E+07 000000    264000.    450000.  0.496683E+08 0.994702E+07 0.745090E+06 0.999981E+05
  1    2    12  2 15 16  1 29  .3632317E+07 000000    256276.    450000.  0.496683E+08 0.995280E+07 0.745090E+06 0.999986E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
            12  2 15 16  1 30       3629936.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 15 15  1 30  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  2 15 15  1 30   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 14 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 14 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 13 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 13 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 12 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 12 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 11 15  1 30   FLAG1= 0
FROM LOCMIN: RETURN 8
  1    2    12  1 11 15  1 30  .3559262E+07 000000    237273.    605000.  0.493837E+08 0.815705E+07 0.353311E+05 0.999325E+05
  1    2    13  1 11 15  1 30  .3669875E+07 000000    224546.    505000.  0.498728E+08 0.815705E+07 0.800858E+06 0.999325E+05
  1    2    11  1 11 15  1 30  .8244067E+13 000010    250000.    705000.  0.467668E+08 0.815705E+07 -.406056E+07 0.999325E+05
  1    2    12  2 11 15  1 30  .3677386E+07 000000    237273.    515000.  0.498658E+08 0.815705E+07 0.942187E+06 0.999325E+05
  1    2    12  1 12 15  1 30  .5745080E+11 000010    250000.    605000.  0.492416E+08 0.815705E+07 -.338961E+06 0.999325E+05
  1    2    12  1 10 15  1 30  .3561683E+07 000000    222000.    605000.  0.494587E+08 0.815705E+07 0.214830E+06 0.999325E+05
  1    2    12  1 11 16  1 30  .3521814E+07 000000    223273.    540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
  1    2    12  1 11 15  2 30  .3747451E+07 000000    237273.    475000.  0.493837E+08 0.820297E+07 0.353311E+05 0.999958E+05
  1    2    12  1 11 15  1 29  .3553342E+07 000000    230031.    605000.  0.493837E+08 0.826173E+07 0.353311E+05 0.999492E+05
  1    2    12  1 11 16  1 30  .3521814E+07 000000    223273.    540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
  1    2    12  1 11 17  1 30  .3627149E+07 000000    209273.    475000.  0.493837E+08 0.999838E+07 0.353311E+05 0.999999E+05
  1    2    13  1 11 16  1 30  .3632427E+07 000000    210546.    440000.  0.498728E+08 0.994702E+07 0.800858E+06 0.999981E+05
  1    2    11  1 11 16  1 30  .8244067E+13 000010    236000.    640000.  0.467668E+08 0.994702E+07 -.406056E+07 0.999981E+05
  1    2    12  2 11 16  1 30  .3639938E+07 000000    223273.    450000.  0.498658E+08 0.994702E+07 0.942187E+06 0.999981E+05
  1    2    12  1 11 16  1 30  .4021814E+07 000000    223273.    540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
  1    2    12  1 12 16  1 30  .5745076E+11 000010    236000.    540000.  0.492416E+08 0.994702E+07 -.338961E+06 0.999981E+05
  1    2    12  1 10 16  1 30  .3524235E+07 000000    208000.    540000.  0.494587E+08 0.994702E+07 0.214830E+06 0.999981E+05
  1    2    12  1 11 17  1 30  .3627149E+07 000000    209273.    475000.  0.493837E+08 0.999838E+07 0.353311E+05 0.999999E+05
  1    2    12  1 11 15  1 30  .3559262E+07 000000    237273.    605000.  0.493837E+08 0.815705E+07 0.353311E+05 0.999325E+05
  1    2    12  1 11 16  2 30  .3711545E+07 000000    223273.    410000.  0.493837E+08 0.997417E+07 0.353311E+05 0.999999E+05
  1    2    12  1 11 16  1 30  .4021814E+07 000000    223273.    540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
  1    2    12  1 11 16  1 30  .4021814E+07 000000    223273.    540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
  1    2    12  1 11 16  1 29  .3524195E+07 000000    215549.    540000.  0.493837E+08 0.995280E+07 0.353311E+05 0.999986E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
            12  1 11 16  1 30       3521814.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 11 15  1 30  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  1 11 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 11 15  1 30   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5  9 15  1 30   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
  1    2    12  1 11 16  1 30       3521814.          223273.    540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
```

```
                         DESIGN COMBINATION:   EQUIPMENT 1:  1      EQUIPMENT 2:  2

FROM INIT1: INITIAL BASE POINT:              14    1    8   18    1   21
FROM INIT1: INITIAL PENALTY PARAMETERS:     1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT1: INITIAL LAGRANGE MULTIPLIERS:   0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC      APRC       AOPC         ASUNA(1)      ASUNA(2)      ACUNA(1)      ACUNA(2)
    1       2    14  1  8 18  1 21  .1651525E+08 100000     -5000.    210000.   0.499845E+08 0.999999E+07 0.980548E+06 0.100000E+06
    1       2    15  1  8 18  1 21  .2572605E+09 100000    -22500.    110000.   0.499967E+08 0.999999E+07 0.995882E+06 0.100000E+06
    1       2    13  1  8 18  1 21  .3895869E+07 000000     12500.    310000.   0.499218E+08 0.999999E+07 0.901587E+06 0.100000E+06
    1       2    14  2  8 18  1 21  .1664191E+08 100000     -5000.    120000.   0.499991E+08 0.999999E+07 0.999706E+06 0.100000E+06
    1       2    14  1  9 18  1 21  .4010516E+07 000000     22222.    210000.   0.499808E+08 0.999999E+07 0.973837E+06 0.100000E+06
    1       2    14  1  8 19  1 21  .3166262E+09 100000    -25000.    145000.   0.499845E+08 0.100000E+08 0.980548E+06 0.100000E+06
    1       2    14  1  8 17  1 21  .3903690E+07 000000     15000.    275000.   0.499845E+08 0.999963E+07 0.980548E+06 0.100000E+06
    1       2    14  1  8 18  2 21  .1670721E+08 100000     -5000.     80000.   0.499845E+08 0.100000E+08 0.980548E+06 0.100000E+06
    1       2    14  1  8 18  1 22  .4010215E+07 000000     11364.    210000.   0.499845E+08 0.999999E+07 0.980548E+06 0.100000E+06
    1       2    13  1  9 17  1 22  .3775357E+07 000000     73232.    375000.   0.499068E+08 0.999963E+07 0.873112E+06 0.100000E+06
    1       2    12  1 10 16  1 23  .3545417E+07 000000    139826.    540000.   0.494587E+08 0.998038E+07 0.214830E+06 0.999999E+05
    1       2    11  1 11 15  1 24  .8244067E+13 000010    197500.    705000.   0.467668E+08 0.878104E+07 -.406056E+07 0.999911E+05
    1       2    13  1 11 17  1 24  .3758810E+07 000000    137046.    375000.   0.498728E+08 0.999952E+07 0.800858E+06 0.100000E+06
    1       2    11  1  9 15  1 22  .3771212E+13 000010    142525.    705000.   0.472498E+08 0.888408E+07 -.274635E+07 0.999942E+05
    1       2    12  2 10 16  1 23  .3664625E+07 000000    139826.    450000.   0.498815E+08 0.998038E+07 0.952892E+06 0.999999E+05
    1       2    12  1 10 16  1 23  .4045417E+07 000000    139826.    540000.   0.494587E+08 0.998038E+07 0.214830E+06 0.999999E+05
    1       2    11  1 10 16  1 23  .5417998E+13 000010    153826.    640000.   0.470415E+08 0.998038E+07 -.329181E+07 0.999999E+05
    1       2    13  1 10 16  1 23  .3657353E+07 000000    125826.    440000.   0.498928E+08 0.998038E+07 0.844437E+06 0.999999E+05
    1       2    12  1  9 17  1 24  .3653571E+07 000000    115833.    475000.   0.495136E+08 0.999952E+07 0.337480E+06 0.100000E+06
    1       2    12  1 11 16  1 23  .3542996E+07 000000    155099.    540000.   0.493837E+08 0.998038E+07 0.353311E+05 0.999999E+05
    1       2    12  1 12 15  1 22  .5745076E+11 000010    173636.    605000.   0.492416E+08 0.888408E+07 -.338961E+06 0.999942E+05
    1       2    12  1 11 16  2 23  .3734138E+07 000000    155099.    410000.   0.493837E+08 0.999034E+07 0.353311E+05 0.100000E+06
    1       2    12  1 11 16  1 23  .4042996E+07 000000    155099.    540000.   0.493837E+08 0.998038E+07 0.353311E+05 0.999999E+05
    1       2    12  1 11 15  1 24  .3526672E+07 000000    184773.    605000.   0.493837E+08 0.878104E+07 0.353311E+05 0.999911E+05
    1       2    12  1 11 15  1 25  .4023102E+07 000000    195273.    605000.   0.493837E+08 0.878104E+07 0.353311E+05 0.999911E+05
    1       2    13  1 12 16  1 25  .3642879E+07 000000    179533.    440000.   0.498328E+08 0.997662E+07 0.704910E+06 0.999998E+05
    1       2    11  1 10 15  1 23  .5417998E+13 000010    172087.    705000.   0.470415E+08 0.888408E+07 -.329181E+07 0.999942E+05
    1       2    12  2 11 15  1 24  .3644796E+07 000000    184773.    515000.   0.498658E+08 0.878104E+07 0.942187E+06 0.999911E+05
    1       2    12  1 11 15  1 24  .4026672E+07 000000    184773.    605000.   0.493837E+08 0.878104E+07 0.353311E+05 0.999911E+05
    1       2    11  1 11 15  1 24  .8244067E+13 000010    197500.    705000.   0.467668E+08 0.878104E+07 -.406056E+07 0.999911E+05
    1       2    13  1 11 15  1 24  .3637285E+07 000000    172046.    505000.   0.498728E+08 0.878104E+07 0.800858E+06 0.999911E+05
    1       2    12  1 10 15  1 25  .3525523E+07 000000    180000.    605000.   0.494587E+08 0.878104E+07 0.214830E+06 0.999911E+05
    1       2    12  1 10 15  1 26  .4039141E+07 000000    189692.    605000.   0.494587E+08 0.857413E+07 0.214830E+06 0.999807E+05
    1       2    12  1 10 15  2 25  .3715814E+07 000000    180000.    475000.   0.494587E+08 0.880136E+07 0.214830E+06 0.999994E+05
    1       2    12  1 10 15  1 25  .4025523E+07 000000    180000.    605000.   0.494587E+08 0.878104E+07 0.214830E+06 0.999911E+05
    1       2    12  1 11 15  1 25  .4023102E+07 000000    195273.    605000.   0.493837E+08 0.878104E+07 0.353311E+05 0.999911E+05
    1       2    12  1  9 16  1 25  .3540847E+07 000000    144533.    540000.   0.495136E+08 0.997662E+07 0.337480E+06 0.999998E+05
    1       2    13  1 11 16  1 26  .3643187E+07 000000    176084.    440000.   0.498728E+08 0.996811E+07 0.800858E+06 0.999996E+05
    1       2    11  1  9 15  1 24  .3771212E+13 000010    166389.    705000.   0.472498E+08 0.878104E+07 -.274635E+07 0.999911E+05
    1       2    12  2 10 15  1 25  .3644731E+07 000000    180000.    515000.   0.498815E+08 0.878104E+07 0.952892E+06 0.999911E+05
    1       2    12  1 10 15  1 25  .4025523E+07 000000    180000.    605000.   0.494587E+08 0.878104E+07 0.214830E+06 0.999911E+05
    1       2    11  1 10 15  1 25  .5417998E+13 000010    194000.    705000.   0.470415E+08 0.878104E+07 -.329181E+07 0.999911E+05
    1       2    13  1 10 15  1 25  .3637459E+07 000000    166000.    505000.   0.498928E+08 0.878104E+07 0.844437E+06 0.999911E+05
    1       2    12  1  9 15  1 26  .3542094E+07 000000    171026.    605000.   0.495136E+08 0.857413E+07 0.337480E+06 0.999807E+05
    1       2    12  1 11 15  1 24  .3526672E+07 000000    184773.    605000.   0.493837E+08 0.878104E+07 0.353311E+05 0.999911E+05
    1       2    12  1 10 15  2 25  .3715814E+07 000000    180000.    475000.   0.494587E+08 0.880136E+07 0.214830E+06 0.999994E+05
    1       2    12  1 10 15  1 25  .4025523E+07 000000    180000.    605000.   0.494587E+08 0.878104E+07 0.214830E+06 0.999911E+05
    1       2    12  1 11 15  1 25  .4023102E+07 000000    195273.    605000.   0.493837E+08 0.878104E+07 0.353311E+05 0.999911E+05
    1       2    12  1  9 16  1 25  .3540847E+07 000000    144533.    540000.   0.495136E+08 0.997662E+07 0.337480E+06 0.999998E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                 12  1 10 15  1 25     3525523.
FROM LOCMIN: RETURN 2
    1       2    12  1 11 15  1 25  .3523102E+07 000000    195273.    605000.   0.493837E+08 0.878104E+07 0.353311E+05 0.999911E+05
    1       2    13  1 11 15  1 25  .3633715E+07 000000    182546.    505000.   0.498728E+08 0.878104E+07 0.800858E+06 0.999911E+05
    1       2    11  1 11 15  1 25  .8244067E+13 000010    208000.    705000.   0.467668E+08 0.878104E+07 -.406056E+07 0.999911E+05
    1       2    12  2 11 15  1 25  .3641226E+07 000000    195273.    515000.   0.498658E+08 0.878104E+07 0.942187E+06 0.999911E+05
    1       2    12  1 12 15  1 25  .5745076E+11 000010    208000.    605000.   0.492416E+08 0.878104E+07 -.338961E+06 0.999911E+05
    1       2    12  1 10 15  1 25  .3525523E+07 000000    180000.    605000.   0.494587E+08 0.878104E+07 0.214830E+06 0.999911E+05
    1       2    12  1 11 16  1 25  .3535473E+07 000000    178473.    540000.   0.493837E+08 0.997662E+07 0.353311E+05 0.999998E+05
    1       2    12  1 11 15  2 25  .3713393E+07 000000    195273.    475000.   0.493837E+08 0.880136E+07 0.353311E+05 0.999994E+05
    1       2    12  1 11 15  1 26  .3536720E+07 000000    204965.    605000.   0.493837E+08 0.857413E+07 0.353311E+05 0.999807E+05
    1       2    12  1 11 15  1 24  .3526672E+07 000000    184773.    605000.   0.493837E+08 0.878104E+07 0.353311E+05 0.999911E+05
FROM INNER: RESOLUTION VALLEY OR LOCAL MINIMUM ENCOUNTERED IN GRADIENT APPOXIMATION SECTION - RETURN TO DUNLOP.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 11 15  1 25  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
```

```
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 11 15  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 10 15  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 10 15  1 24   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 15  1 24   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5 10 15  1 24   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5  9 15  1 24   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  6  9 15  1 24   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  7  9 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  7  8 15  1 23   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   1       2       12  1 11 15  1 25     3523102.          195273.     605000.  0.493837E+08 0.878104E+07 0.353311E+05 0.999911E+05

                        DESIGN COMBINATION:   EQUIPMENT 1:  1      EQUIPMENT 2:  2

FROM INIT2: INITIAL BASE POINT:              11    2   14   15    2   30
FROM INIT2: INITIAL PENALTY PARAMETERS:     1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT2: INITIAL LAGRANGE MULTIPLIERS:   0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC      APRC        AOPC        ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   1       2     11  2 14 15  2 30  .9956858E+11 000010    280000.     485000.  0.478098E+08 0.820297E+07 -.446240E+06 0.999958E+05
   1       2     12  2 14 15  2 30  .3857462E+07 000000    270000.     385000.  0.497387E+08 0.820297E+07 0.827465E+06 0.999958E+05
   1       2     11  3 14 15  2 30  .3898462E+07 000000    280000.     395000.  0.479443E+08 0.820297E+07 0.372508E+06 0.999958E+05
   1       2     11  2 15 15  2 30  .4745071E+12 000010    287333.     485000.  0.474311E+08 0.820297E+07 -.974170E+06 0.999958E+05
   1       2     11  2 13 15  2 30  .2988401E+10 000010    271539.     485000.  0.481161E+08 0.820297E+07 -.772610E+05 0.999958E+05
   1       2     11  2 14 16  2 30  .9956852E+11 000010    266000.     420000.  0.478098E+08 0.997417E+07 -.446240E+06 0.999999E+05
   1       2     11  2 14 15  3 30  .9956878E+11 000010    280000.     355000.  0.478098E+08 0.820324E+07 -.446240E+06 0.999987E+05
   1       2     11  2 14 15  1 30  .9956839E+11 000010    280000.     615000.  0.478098E+08 0.815705E+07 -.446240E+06 0.999325E+05
   1       2     11  2 14 15  2 29  .9956858E+11 000010    272759.     485000.  0.478098E+08 0.830266E+07 -.446240E+06 0.999968E+05
   1       2     12  3 13 15  2 30  .3985335E+07 000000    260769.     295000.  0.498688E+08 0.820297E+07 0.965440E+06 0.999958E+05
   1       2     13  2 14 15  2 30  .3971980E+07 000000    260000.     285000.  0.499662E+08 0.820297E+07 0.977703E+06 0.999958E+05
   1       2     11  2 14 15  2 30  .9956858E+11 000010    280000.     485000.  0.478098E+08 0.820297E+07 -.446240E+06 0.999958E+05
   1       2     12  3 14 15  2 30  .3982602E+07 000000    270000.     295000.  0.498365E+08 0.820297E+07 0.950083E+06 0.999958E+05
   1       2     12  1 14 15  2 30  .1795590E+13 000010    270000.     475000.  0.487604E+08 0.820297E+07 -.189504E+07 0.999958E+05
   1       2     12  2 15 15  2 30  .3855573E+07 000000    278000.     385000.  0.496683E+08 0.820297E+07 0.745090E+06 0.999958E+05
   1       2     12  2 15 15  2 30  .4355573E+07 000000    278000.     385000.  0.496683E+08 0.820297E+07 0.745090E+06 0.999958E+05
   1       2     12  2 15 16  2 30  .3819667E+07 000000    264000.     320000.  0.496683E+08 0.997417E+07 0.745090E+06 0.999999E+05
   1       2     12  2 15 17  2 30  .3927131E+07 000000    250000.     255000.  0.496683E+08 0.999961E+07 0.745090E+06 0.100000E+06
   1       2     12  2 15 16  3 30  .4011609E+07 000000    264000.     190000.  0.496683E+08 0.997439E+07 0.745090E+06 0.999999E+05
   1       2     12  2 15 16  1 30  .3629936E+07 000000    264000.     450000.  0.496683E+08 0.994702E+07 0.745090E+06 0.999981E+05
   1       2     12  2 15 16  1 30  .4129936E+07 000000    264000.     450000.  0.496683E+08 0.994702E+07 0.745090E+06 0.999981E+05
   1       2     12  2 15 16  1 30  .4129936E+07 000000    264000.     450000.  0.496683E+08 0.994702E+07 0.745090E+06 0.999981E+05
   1       2     12  2 15 16  1 29  .3632317E+07 000000    256276.     450000.  0.496683E+08 0.995280E+07 0.745090E+06 0.999986E+05
   1       2     13  2 15 16  1 30  .3743136E+07 000000    254667.     350000.  0.499536E+08 0.994702E+07 0.964344E+06 0.999981E+05
   1       2     11  2 15 16  1 30  .4745069E+12 000010    273333.     550000.  0.474311E+08 0.994702E+07 -.974170E+06 0.999981E+05
   1       2     12  3 15 16  1 30  .3754575E+07 000000    264000.     360000.  0.497935E+08 0.994702E+07 0.926395E+06 0.999981E+05
   1       2     12  1 15 16  1 30  .5393732E+13 000010    264000.     540000.  0.484056E+08 0.994702E+07 -.328443E+07 0.999981E+05
   1       2     12  2 15 17  2 30  .4427131E+07 000000    250000.     255000.  0.496683E+08 0.999961E+07 0.745090E+06 0.100000E+06
   1       2     12  2 14 15  1 30  .4169273E+07 000000    270000.     515000.  0.497387E+08 0.815705E+07 0.827465E+06 0.999325E+05
   1       2     12  2 14 16  1 30  .3631825E+07 000000    256000.     450000.  0.497387E+08 0.994702E+07 0.827465E+06 0.999981E+05
   1       2     12  2 15 16  1 30  .4129936E+07 000000    264000.     450000.  0.496683E+08 0.994702E+07 0.745090E+06 0.999981E+05
   1       2     12  2 15 15  2 30  .3855573E+07 000000    278000.     385000.  0.496683E+08 0.820297E+07 0.745090E+06 0.999958E+05
   1       2     12  2 15 17  1 30  .4235271E+07 000000    250000.     385000.  0.496683E+08 0.999838E+07 0.745090E+06 0.999999E+05
   1       2     12  2 15 16  1 30  .4129936E+07 000000    264000.     450000.  0.496683E+08 0.994702E+07 0.745090E+06 0.999981E+05
   1       2     12  2 15 16  1 29  .3632317E+07 000000    256276.     450000.  0.496683E+08 0.995280E+07 0.745090E+06 0.999986E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                 12  2 15 16  1 30     3629936.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 15 15  1 30  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  2 15 15  1 30   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 14 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 14 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 13 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 13 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 12 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 12 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 11 15  1 30   FLAG1= 0
FROM LOCMIN: RETURN 8
```

```
  1       2      12  1 11 15  1 30  .3559262E+07 000000    237273.     605000.  0.493837E+08 0.815705E+07 0.353311E+05 0.999325E+05
  1       2      13  1 11 15  1 30  .3669875E+07 000000    224546.     505000.  0.498728E+08 0.815705E+07 0.800858E+06 0.999325E+05
  1       2      11  1 11 15  1 30  .8244067E+13 000010    250000.     705000.  0.467668E+08 0.815705E+07 -.406056E+07 0.999325E+05
  1       2      12  2 11 15  1 30  .3677386E+07 000000    237273.     515000.  0.498658E+08 0.815705E+07 0.942187E+06 0.999325E+05
  1       2      12  1 12 15  1 30  .5745080E+11 000010    250000.     605000.  0.492416E+08 0.815705E+07 -.338961E+06 0.999325E+05
  1       2      12  1 10 15  1 30  .3561683E+07 000000    222000.     605000.  0.494587E+08 0.815705E+07 0.214830E+06 0.999325E+05
  1       2      12  1 11 16  1 30  .3521814E+07 000000    223273.     540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
  1       2      12  1 11 15  2 30  .3747451E+07 000000    237273.     475000.  0.493837E+08 0.820297E+07 0.353311E+05 0.999958E+05
  1       2      12  1 11 15  1 29  .3553342E+07 000000    230031.     605000.  0.493837E+08 0.826173E+07 0.353311E+05 0.999492E+05
  1       2      12  1 11 16  1 30  .3521814E+07 000000    223273.     540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
  1       2      12  1 11 17  1 30  .3627149E+07 000000    209273.     475000.  0.493837E+08 0.999838E+07 0.353311E+05 0.999999E+05
  1       2      13  1 11 16  1 30  .3632427E+07 000000    210546.     440000.  0.498728E+08 0.994702E+07 0.800858E+06 0.999981E+05
  1       2      11  1 11 16  1 30  .8244067E+13 000010    236000.     640000.  0.467668E+08 0.994702E+07 -.406056E+07 0.999981E+05
  1       2      12  2 11 16  1 30  .3639938E+07 000000    223273.     450000.  0.498658E+08 0.994702E+07 0.942187E+06 0.999981E+05
  1       2      12  1 11 16  1 30  .4021814E+07 000000    223273.     540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
  1       2      12  1 12 16  1 30  .5745076E+11 000010    236000.     540000.  0.492416E+08 0.994702E+07 -.338961E+06 0.999981E+05
  1       2      12  1 10 16  1 30  .3524235E+07 000000    208000.     540000.  0.494587E+08 0.994702E+07 0.214830E+06 0.999981E+05
  1       2      12  1 11 17  1 30  .3627149E+07 000000    209273.     475000.  0.493837E+08 0.999838E+07 0.353311E+05 0.999999E+05
  1       2      12  1 11 15  1 30  .3559262E+07 000000    237273.     605000.  0.493837E+08 0.815705E+07 0.353311E+05 0.999325E+05
  1       2      12  1 11 16  2 30  .3711545E+07 000000    223273.     410000.  0.493837E+08 0.997417E+07 0.353311E+05 0.999999E+05
  1       2      12  1 11 16  1 30  .4021814E+07 000000    223273.     540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
  1       2      12  1 11 16  1 30  .4021814E+07 000000    223273.     540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
  1       2      12  1 11 16  1 29  .3524195E+07 000000    215549.     540000.  0.493837E+08 0.995280E+07 0.353311E+05 0.999986E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                 12  1 11 16  1 30       3521814.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 11 15  1 30  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  1 11 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 11 15  1 30   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5 10 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5  9 15  1 30   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
  1       2      12  1 11 16  1 30       3521814.          223273.     540000.  0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05


                            DESIGN COMBINATION:   EQUIPMENT 1:  2      EQUIPMENT 2:  1

FROM INIT: INITIAL BASE POINT:                 11    2   15   16    2   19
FROM INIT: INITIAL PENALTY PARAMETERS:        1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT: INITIAL LAGRANGE MULTIPLIERS:      0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2    TOTAL COST  FLAGC       APRC         AOPC        ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
  2       1      11  2 15 16  2 19  .3635397E+07 000000    187930.     450000.  0.486301E+08 0.997696E+07 0.423291E+06 0.999999E+05
  2       1      12  2 15 16  2 19  .3722535E+07 000000    177263.     360000.  0.498706E+08 0.997696E+07 0.945514E+06 0.999999E+05
  2       1      10  2 15 16  2 19  .1601193E+14 000010    198597.     540000.  0.341826E+08 0.997696E+07 -.565896E+07 0.999999E+05
  2       1      11  3 15 16  2 19  .3761111E+07 000000    187930.     360000.  0.486964E+08 0.997696E+07 0.748505E+06 0.999999E+05
  2       1      11  1 15 16  2 19  .7314985E+13 000010    187930.     540000.  0.468489E+08 0.997696E+07 -.382491E+07 0.999999E+05
  2       1      11  2 16 16  2 19  .3636529E+07 000000    195263.     450000.  0.484377E+08 0.997696E+07 0.253543E+06 0.999999E+05
  2       1      11  2 14 16  2 19  .3635186E+07 000000    179549.     450000.  0.487783E+08 0.997696E+07 0.539511E+06 0.999999E+05
  2       1      11  2 15 17  2 19  .3743731E+07 000000    169509.     380000.  0.486301E+08 0.999967E+07 0.423291E+06 0.100000E+06
  2       1      11  2 15 15  2 19  .3662698E+07 000000    206351.     520000.  0.486301E+08 0.830266E+07 0.423291E+06 0.999968E+05
  2       1      11  2 15 16  3 19  .3827342E+07 000000    187930.     320000.  0.486301E+08 0.997714E+07 0.423291E+06 0.999999E+05
  2       1      11  2 15 16  1 19  .3445421E+07 000000    187930.     580000.  0.486301E+08 0.995280E+07 0.423291E+06 0.999986E+05
  2       1      11  2 15 16  2 20  .3630654E+07 000000    202667.     450000.  0.486301E+08 0.997417E+07 0.423291E+06 0.999999E+05
  2       1      11  2 15 16  1 19  .3445421E+07 000000    187930.     580000.  0.486301E+08 0.995280E+07 0.423291E+06 0.999986E+05
  2       1      11  2 15 16  1 19  .3945421E+07 000000    187930.     580000.  0.486301E+08 0.995280E+07 0.423291E+06 0.999986E+05
  2       1      12  2 15 16  1 19  .3532559E+07 000000    177263.     490000.  0.498706E+08 0.995280E+07 0.945514E+06 0.999986E+05
  2       1      10  2 15 16  1 19  .1601193E+14 000010    198597.     670000.  0.341826E+08 0.995280E+07 -.565896E+07 0.999986E+05
  2       1      11  3 15 16  1 19  .3571135E+07 000000    187930.     490000.  0.486964E+08 0.995280E+07 0.748505E+06 0.999986E+05
  2       1      11  1 15 16  1 19  .7314985E+13 000010    187930.     670000.  0.468489E+08 0.995280E+07 -.382491E+07 0.999986E+05
  2       1      11  2 16 16  1 19  .3446553E+07 000000    195263.     580000.  0.484377E+08 0.995280E+07 0.253543E+06 0.999986E+05
  2       1      11  2 14 16  1 19  .3445210E+07 000000    179549.     580000.  0.487783E+08 0.995280E+07 0.539511E+06 0.999986E+05
  2       1      11  2 13 16  1 19  .3445547E+07 000000    169879.     580000.  0.489030E+08 0.995280E+07 0.627456E+06 0.999986E+05
  2       1      11  2 14 17  1 19  .3551644E+07 000000    161128.     510000.  0.487783E+08 0.999864E+07 0.539511E+06 0.999999E+05
```

```
   2       1      11  2 14 15  1 19  .3473889E+07 000000      197970.     650000.  0.487783E+08 0.826173E+07 0.539511E+06 0.999492E+05
   2       1      11  2 14 16  2 19  .3635186E+07 000000      179549.     450000.  0.487783E+08 0.997696E+07 0.539511E+06 0.999999E+05
   2       1      11  2 14 16  1 19  .3945210E+07 000000      179549.     580000.  0.487783E+08 0.995280E+07 0.539511E+06 0.999986E+05
   2       1      11  2 14 16  1 20  .3440712E+07 000000      194286.     580000.  0.487783E+08 0.994702E+07 0.539511E+06 0.999981E+05
   2       1      11  2 14 16  1 21  .3437580E+07 000000      207619.     580000.  0.487783E+08 0.992763E+07 0.539511E+06 0.999951E+05
   2       1      11  2 14 16  1 22  .3434305E+07 000000      219740.     580000.  0.487783E+08 0.991297E+07 0.539511E+06 0.999916E+05
   2       1      11  2 14 16  1 24  .3429547E+07 000000      240952.     580000.  0.487783E+08 0.987023E+07 0.539511E+06 0.999727E+05
   2       1      11  2 14 16  1 25  .5427346E+07 000000      250286.     580000.  0.487783E+08 0.985067E+07 0.539511E+06 0.999587E+05
   2       1      11  2 14 16  1 25  .3427346E+07 000000      250286.     580000.  0.487783E+08 0.985067E+07 0.539511E+06 0.999587E+05
   2       1      11  2 14 16  1 25  .3927346E+07 000000      250286.     580000.  0.487783E+08 0.985067E+07 0.539511E+06 0.999587E+05
   2       1      12  2 14 16  1 25  .3517043E+07 000000      238857.     490000.  0.498910E+08 0.985067E+07 0.958904E+06 0.999587E+05
   2       1      10  2 14 16  1 25  .1071316E+14 000010      261714.     670000.  0.350661E+08 0.985067E+07 -.462886E+07 0.999587E+05
   2       1      11  3 14 16  1 25  .3553251E+07 000000      250286.     490000.  0.488341E+08 0.985067E+07 0.798972E+06 0.999587E+05
   2       1      11  1 14 16  1 25  .4047831E+13 000010      250286.     670000.  0.472111E+08 0.985067E+07 -.284529E+07 0.999587E+05
   2       1      11  2 14 16  1 25  .3927346E+07 000000      250286.     580000.  0.487783E+08 0.985067E+07 0.539511E+06 0.999587E+05
   2       1      11  2 14 16  1 24  .3429547E+07 000000      240952.     580000.  0.487783E+08 0.987023E+07 0.539511E+06 0.999727E+05
   2       1      11  2 14 17  1 25  .3524263E+07 000000      236286.     510000.  0.487783E+08 0.999241E+07 0.539511E+06 0.999979E+05
   2       1      11  2 14 15  1 25  .3562827E+07 000000      264286.     650000.  0.487783E+08 0.687910E+07 0.539511E+06 0.991378E+05
   2       1      11  2 14 16  2 25  .3612922E+07 000000      250286.     450000.  0.487783E+08 0.992840E+07 0.539511E+06 0.999987E+05
   2       1      11  2 14 16  1 25  .3927346E+07 000000      250286.     580000.  0.487783E+08 0.985067E+07 0.539511E+06 0.999587E+05
   2       1      11  2 13 16  1 25  .3427683E+07 000000      240615.     580000.  0.489030E+08 0.985067E+07 0.627456E+06 0.999587E+05
   2       1      11  2 15 16  1 25  .3427557E+07 000000      258667.     580000.  0.486301E+08 0.985067E+07 0.423291E+06 0.999587E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                  11  2 14 16  1 25       3427346.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 12  1 14 17  1 25  FLAGC =000000
FROM LOCMIN: NEXT GRAD SEARCH POINT: 13  1 14 18  1 25  FLAGC =000000
FROM LOCMIN: THERE ARE  0 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  2 14 17  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  13  2 14 18  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  14  2 14 19  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  15  2 14 20  1 25   FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  16  2 14 21  1 25   FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  17  2 14 22  1 25   FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  18  2 14 23  1 25   FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  19  2 14 24  1 25   FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  20  2 14 25  1 25   FLAG1= 2
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   2       1      11  2 14 16  1 25       3427346.            250286.     580000.  0.487783E+08 0.985067E+07 0.539511E+06 0.999587E+05


                              DESIGN COMBINATION:   EQUIPMENT 1:  2      EQUIPMENT 2:  1

FROM INIT1: INITIAL BASE POINT:                  13    1   10   18    1   17
FROM INIT1: INITIAL PENALTY PARAMETERS:         1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT1: INITIAL LAGRANGE MULTIPLIERS:       0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2    TOTAL COST  FLAGC          APRC         AOPC          ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   2       1      13  1 10 18  1 17  .3758874E+07 000000       21412.     350000.  0.499411E+08 0.999998E+07 0.934700E+06 0.100000E+06
   2       1      14  1 10 18  1 17  .3869016E+07 000000        5412.     260000.  0.499891E+08 0.999998E+07 0.987944E+06 0.100000E+06
   2       1      12  1 10 18  1 17  .3653049E+07 000000       37412.     440000.  0.496564E+08 0.999998E+07 0.619191E+06 0.100000E+06
   2       1      13  2 10 18  1 17  .3884843E+07 000000       21412.     260000.  0.499934E+08 0.999998E+07 0.998029E+06 0.100000E+06
   2       1      13  1 11 18  1 17  .3755418E+07 000000       40321.     350000.  0.499374E+08 0.999998E+07 0.928739E+06 0.100000E+06
   2       1      13  1 10 19  1 17  .3869743E+07 000000         824.     280000.  0.499411E+08 0.100000E+08 0.934700E+06 0.100000E+06
   2       1      13  1 10 17  1 17  .3648065E+07 000000       42000.     420000.  0.499411E+08 0.999924E+07 0.934700E+06 0.999999E+05
   2       1      13  1 10 18  2 17  .3950833E+07 000000       21412.     220000.  0.499411E+08 0.100000E+08 0.934700E+06 0.100000E+06
   2       1      13  1 10 18  1 18  .3752688E+07 000000       42000.     350000.  0.499411E+08 0.999997E+07 0.934700E+06 0.100000E+06
   2       1      12  1 10 17  1 17  .3542240E+07 000000       58000.     510000.  0.496564E+08 0.999924E+07 0.619191E+06 0.999999E+05
   2       1      11  1 10 16  1 17  .9873596E+12 000010       94588.     670000.  0.478297E+08 0.996811E+07 -.140525E+07 0.999996E+05
   2       1      13  1 10 18  1 17  .3758874E+07 000000       21412.     350000.  0.499411E+08 0.999998E+07 0.934700E+06 0.100000E+06
   2       1      11  1 10 16  1 17  .9873596E+12 000010       94588.     670000.  0.478297E+08 0.996811E+07 -.140525E+07 0.999996E+05
   2       1      12  2 10 17  1 17  .3664290E+07 000000       58000.     420000.  0.499235E+08 0.999924E+07 0.977063E+06 0.999999E+05
   2       1      12  1 10 17  1 17  .4042240E+07 000000       58000.     510000.  0.496564E+08 0.999924E+07 0.619191E+06 0.999999E+05
   2       1      12  1 11 17  1 17  .3539283E+07 000000       75455.     510000.  0.496404E+08 0.999924E+07 0.590429E+06 0.999999E+05
   2       1      12  1 12 17  1 17  .3536781E+07 000000       90000.     510000.  0.496068E+08 0.999924E+07 0.527749E+06 0.999999E+05
   2       1      12  1 13 17  1 17  .3534288E+07 000000      102308.     510000.  0.495803E+08 0.999924E+07 0.475940E+06 0.999999E+05
   2       1      12  1 15 17  1 17  .3531890E+07 000000      122000.     510000.  0.494064E+08 0.999924E+07 0.911298E+05 0.999999E+05
   2       1      12  1 18 17  1 17  .1557736E+13 000010      143333.     510000.  0.487964E+08 0.999924E+07 -.176507E+07 0.999999E+05
   2       1      12  1 16 17  1 17  .3330126E+11 000010      130000.     510000.  0.492709E+08 0.999924E+07 -.258061E+06 0.999999E+05
   2       1      12  1 14 17  1 17  .3532837E+07 000000      112857.     510000.  0.495036E+08 0.999924E+07 0.315544E+06 0.999999E+05
```

```
   2    1     11  1 15 18  1 17  .7314985E+13 000010     112079.     530000.  0.468489E+08 0.999998E+07 -.382491E+07 0.100000E+06
   2    1     13  1 15 16  1 17  .3524731E+07 000000     131922.     490000.  0.498789E+08 0.996811E+07 0.814594E+06 0.999996E+05
   2    1     14  1 15 15  1 17  .3636394E+07 000000     141843.     470000.  0.499734E+08 0.857413E+07 0.959284E+06 0.999807E+05
   2    1     13  1 15 16  2 17  .3715356E+07 000000     131922.     360000.  0.498789E+08 0.998436E+07 0.814594E+06 0.999999E+05
   2    1     13  1 15 16  1 17  .4024731E+07 000000     131922.     490000.  0.498789E+08 0.996811E+07 0.814594E+06 0.999996E+05
   2    1     13  1 15 16  1 18  .3520042E+07 000000     150222.     490000.  0.498789E+08 0.995824E+07 0.814594E+06 0.999990E+05
   2    1     13  1 15 16  1 19  .3515260E+07 000000     166597.     490000.  0.498789E+08 0.995280E+07 0.814594E+06 0.999986E+05
   2    1     13  1 15 16  1 20  .3510762E+07 000000     181333.     490000.  0.498789E+08 0.994702E+07 0.814594E+06 0.999981E+05
   2    1     13  1 15 16  1 22  .3504355E+07 000000     206788.     490000.  0.498789E+08 0.991297E+07 0.814594E+06 0.999916E+05
   2    1     13  1 15 16  1 25  .3497396E+07 000000     237333.     490000.  0.498789E+08 0.985067E+07 0.814594E+06 0.999587E+05
   2    1     13  1 15 16  1 25  .1599740E+08 000000     237333.     490000.  0.498789E+08 0.985067E+07 0.814594E+06 0.999587E+05
   2    1     13  1 15 16  1 25  .5497396E+07 000000     237333.     490000.  0.498789E+08 0.985067E+07 0.814594E+06 0.999587E+05
   2    1     13  1 15 16  1 24  .3499597E+07 000000     228000.     490000.  0.498789E+08 0.987023E+07 0.814594E+06 0.999727E+05
   2    1     13  1 15 16  1 25  .3997396E+07 000000     237333.     490000.  0.498789E+08 0.985067E+07 0.814594E+06 0.999587E+05
   2    1     14  1 15 17  1 25  .3702366E+07 000000     212667.     330000.  0.499734E+08 0.999241E+07 0.959284E+06 0.999979E+05
   2    1     12  1 15 15  1 25  .3531723E+07 000000     262000.     650000.  0.494064E+08 0.687910E+07 0.911298E+05 0.991378E+05
   2    1     13  2 15 16  1 25  .3622352E+07 000000     237333.     400000.  0.499868E+08 0.985067E+07 0.994423E+06 0.999587E+05
   2    1     13  1 15 16  1 25  .3997396E+07 000000     237333.     490000.  0.498789E+08 0.985067E+07 0.814594E+06 0.999587E+05
   2    1     13  1 16 16  1 25  .3995271E+07 000000     246000.     490000.  0.498413E+08 0.985067E+07 0.726165E+06 0.999587E+05
   2    1     13  1 14 16  1 24  .3502080E+07 000000     218095.     490000.  0.499043E+08 0.987023E+07 0.868056E+06 0.999727E+05
   2    1     13  1 14 16  1 25  .3999879E+07 000000     227429.     490000.  0.499043E+08 0.985067E+07 0.868056E+06 0.999587E+05
   2    1     13  1 16 16  1 24  .3497472E+07 000000     236667.     490000.  0.498413E+08 0.987023E+07 0.726165E+06 0.999727E+05
   2    1     13  1 15 16  2 25  .3682972E+07 000000     237333.     360000.  0.498789E+08 0.992840E+07 0.814594E+06 0.999987E+05
   2    1     13  1 15 16  1 25  .3997396E+07 000000     237333.     490000.  0.498789E+08 0.985067E+07 0.814594E+06 0.999587E+05
   2    1     14  1 15 15  1 25  .3740930E+07 000000     240667.     470000.  0.499734E+08 0.687910E+07 0.959284E+06 0.991378E+05
   2    1     12  1 15 17  1 25  .3493159E+07 000000     234000.     510000.  0.494064E+08 0.999241E+07 0.911298E+05 0.999979E+05
   2    1     11  1 15 18  1 25  .7314985E+13 000010     230667.     530000.  0.468489E+08 0.999959E+07 -.382491E+07 0.999999E+05
   2    1     13  1 15 18  1 25  .3702282E+07 000000     209333.     350000.  0.498789E+08 0.999959E+07 0.814594E+06 0.999999E+05
   2    1     11  1 15 16  1 25  .7314985E+13 000010     258667.     670000.  0.468489E+08 0.985067E+07 -.382491E+07 0.999587E+05
   2    1     12  2 15 17  1 25  .3611612E+07 000000     234000.     420000.  0.498706E+08 0.999241E+07 0.945514E+06 0.999979E+05
   2    1     12  1 15 17  1 25  .3993159E+07 000000     234000.     510000.  0.494064E+08 0.999241E+07 0.911298E+05 0.999979E+05
   2    1     12  1 16 17  1 25  .3330172E+11 000010     242000.     510000.  0.492709E+08 0.999241E+07 -.258061E+06 0.999979E+05
   2    1     12  1 14 17  1 24  .3498034E+07 000000     214941.     510000.  0.495036E+08 0.999384E+07 0.315544E+06 0.999987E+05
   2    1     12  1 14 17  1 25  .3994106E+07 000000     224857.     510000.  0.495036E+08 0.999241E+07 0.315544E+06 0.999979E+05
   2    1     12  1 16 17  1 24  .3330122E+11 000010     232083.     510000.  0.492709E+08 0.999384E+07 -.258061E+06 0.999987E+05
   2    1     12  1 15 17  2 25  .3684645E+07 000000     234000.     380000.  0.494064E+08 0.999818E+07 0.911298E+05 0.999999E+05
   2    1     12  1 15 17  1 25  .3993159E+07 000000     234000.     510000.  0.494064E+08 0.999241E+07 0.911298E+05 0.999979E+05
   2    1     13  1 15 16  1 25  .3497396E+07 000000     237333.     490000.  0.498789E+08 0.985067E+07 0.814594E+06 0.999587E+05
   2    1     11  1 15 18  1 25  .7314985E+13 000010     230667.     530000.  0.468489E+08 0.999959E+07 -.382491E+07 0.999999E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
              12  1 15 17  1 25       3493159.
FROM LOCMIN: RETURN 3
   2    1     12  1 15 16  1 25  .3396242E+07 000000     248000.     580000.  0.494064E+08 0.985067E+07 0.911298E+05 0.999587E+05
   2    1     13  1 15 16  1 25  .3497396E+07 000000     237333.     490000.  0.498789E+08 0.985067E+07 0.814594E+06 0.999587E+05
   2    1     11  1 15 16  1 25  .7314985E+13 000010     258667.     670000.  0.468489E+08 0.985067E+07 -.382491E+07 0.999587E+05
   2    1     12  2 15 16  1 25  .3514695E+07 000000     248000.     490000.  0.498706E+08 0.985067E+07 0.945514E+06 0.999587E+05
   2    1     12  1 16 16  1 25  .3330112E+11 000010     256000.     580000.  0.492709E+08 0.985067E+07 -.258061E+06 0.999587E+05
   2    1     12  1 14 16  1 25  .3397189E+07 000000     238857.     580000.  0.495036E+08 0.985067E+07 0.315544E+06 0.999587E+05
   2    1     12  1 15 17  1 25  .3493159E+07 000000     234000.     510000.  0.494064E+08 0.999241E+07 0.911298E+05 0.999979E+05
   2    1     12  1 15 15  1 25  .3531723E+07 000000     262000.     650000.  0.494064E+08 0.687910E+07 0.911298E+05 0.991378E+05
   2    1     12  1 15 16  2 25  .3581818E+07 000000     248000.     450000.  0.494064E+08 0.992840E+07 0.911298E+05 0.999987E+05
   2    1     12  1 15 16  1 24  .3398443E+07 000000     238667.     580000.  0.494064E+08 0.987023E+07 0.911298E+05 0.999727E+05
FROM INNER: RESOLUTION VALLEY OR LOCAL MINIMUM ENCOUNTERED IN GRADIENT APPOXIMATION SECTION - RETURN TO DUNLOP.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 15 17  1 25  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 15 17  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 15 18  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 15 19  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 14 20  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 14 21  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 14 22  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 14 23  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 14 24  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 14 25  1 25   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   2    1     12  1 15 16  1 25       3396242.           248000.     580000.  0.494064E+08 0.985067E+07 0.911298E+05 0.999587E+05

                          DESIGN COMBINATION:   EQUIPMENT 1:  2      EQUIPMENT 2:  1
```

```
FROM INIT2: INITIAL BASE POINT:              10    3   18   15    2   25
FROM INIT2: INITIAL PENALTY PARAMETERS:     1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT2: INITIAL LAGRANGE MULTIPLIERS:   0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC       APRC        AOPC        ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   2       1     10  3 18 15  2 25 .2720466E+14 000010     301111.     520000.  0.302921E+08 0.700776E+07 -.737627E+07 0.999469E+05
   2       1     11  3 18 15  2 25 .3876390E+07 000000     292222.     430000.  0.479811E+08 0.700776E+07 0.394837E+06 0.999469E+05
   2       1     10  4 18 15  2 25 .2646884E+14 000010     301111.     430000.  0.302960E+08 0.700776E+07 -.727583E+07 0.999469E+05
   2       1     10  3 19 15  2 25 .4563153E+14 000010     305790.     520000.  0.286597E+08 0.700776E+07 -.955317E+07 0.999469E+05
   2       1     10  3 17 15  2 25 .1461379E+14 000010     295882.     520000.  0.320261E+08 0.700776E+07 -.540625E+07 0.999469E+05
   2       1     10  3 18 16  2 25 .2720466E+14 000010     287111.     450000.  0.302921E+08 0.992840E+07 -.737627E+07 0.999987E+05
   2       1     10  3 18 15  3 25 .2720466E+14 000010     301111.     390000.  0.302921E+08 0.700896E+07 -.737627E+07 0.999843E+05
   2       1     10  3 18 15  1 25 .2720466E+14 000010     301111.     650000.  0.302921E+08 0.687910E+07 -.737627E+07 0.991378E+05
   2       1     10  3 18 15  2 24 .2720466E+14 000010     292361.     520000.  0.302921E+08 0.720683E+07 -.737627E+07 0.999622E+05
   2       1     11  3 18 15  2 25 .3876390E+07 000000     292222.     430000.  0.479811E+08 0.700776E+07 0.394837E+06 0.999469E+05
   2       1     12  3 17 15  2 25 .3953274E+07 000000     277059.     340000.  0.498787E+08 0.700776E+07 0.969655E+06 0.999469E+05
   2       1     12  3 18 15  2 25 .3951607E+07 000000     283333.     340000.  0.498409E+08 0.700776E+07 0.952304E+06 0.999469E+05
   2       1     10  3 18 15  2 25 .2720466E+14 000010     301111.     520000.  0.302921E+08 0.700776E+07 -.737627E+07 0.999469E+05
   2       1     11  4 18 15  2 25 .4003228E+07 000000     292222.     340000.  0.479859E+08 0.700776E+07 0.551567E+06 0.999469E+05
   2       1     11  2 18 15  2 25 .7780395E+11 000010     292222.     520000.  0.478503E+08 0.700776E+07 -.394463E+06 0.999469E+05
   2       1     11  3 19 15  2 25 .3880500E+07 000000     297368.     430000.  0.476433E+08 0.700776E+07 0.174453E+06 0.999469E+05
   2       1     11  3 17 15  2 25 .3872499E+07 000000     286471.     430000.  0.483127E+08 0.700776E+07 0.577868E+06 0.999469E+05
   2       1     11  3 16 15  2 25 .3871011E+07 000000     280000.     430000.  0.485186E+08 0.700776E+07 0.674905E+06 0.999469E+05
   2       1     11  3 15 15  2 25 .3870145E+07 000000     272667.     430000.  0.486964E+08 0.700776E+07 0.748505E+06 0.999469E+05
   2       1     11  3 13 15  2 25 .3870615E+07 000000     254615.     430000.  0.489504E+08 0.700776E+07 0.837210E+06 0.999469E+05
   2       1     11  3 14 15  2 25 .3870125E+07 000000     264286.     430000.  0.488341E+08 0.700776E+07 0.798972E+06 0.999469E+05
   2       1     11  3 14 16  2 25 .3738827E+07 000000     250286.     360000.  0.488341E+08 0.992840E+07 0.798972E+06 0.999987E+05
   2       1     11  3 14 17  2 25 .3841654E+07 000000     236286.     290000.  0.488341E+08 0.999818E+07 0.798972E+06 0.999999E+05
   2       1     11  3 14 16  3 25 .3930704E+07 000000     250286.     230000.  0.488341E+08 0.992941E+07 0.798972E+06 0.999997E+05
   2       1     11  3 14 16  1 25 .3553251E+07 000000     250286.     490000.  0.488341E+08 0.985067E+07 0.798972E+06 0.999587E+05
   2       1     11  3 14 16  1 25 .4053251E+07 000000     250286.     490000.  0.488341E+08 0.985067E+07 0.798972E+06 0.999587E+05
   2       1     11  3 14 16  1 25 .4053251E+07 000000     250286.     490000.  0.488341E+08 0.985067E+07 0.798972E+06 0.999587E+05
   2       1     11  3 14 16  1 24 .3555452E+07 000000     240952.     490000.  0.488341E+08 0.987023E+07 0.798972E+06 0.999727E+05
   2       1     12  3 14 16  1 25 .3643244E+07 000000     238857.     400000.  0.499306E+08 0.985067E+07 0.988039E+06 0.999587E+05
   2       1     10  3 14 16  1 25 .3606451E+13 000010     261714.     580000.  0.351233E+08 0.985067E+07 -.268568E+07 0.999587E+05
   2       1     11  4 14 16  1 25 .3680147E+07 000000     250286.     400000.  0.488357E+08 0.985067E+07 0.850580E+06 0.999587E+05
   2       1     11  2 14 16  1 25 .3427346E+07 000000     250286.     580000.  0.487783E+08 0.985067E+07 0.539511E+06 0.999587E+05
   2       1     11  1 14 16  1 25 .4047831E+13 000010     250286.     670000.  0.472111E+08 0.985067E+07 -.284529E+07 0.999587E+05
   2       1     11  2 15 16  1 25 .3427557E+07 000000     258667.     580000.  0.486301E+08 0.985067E+07 0.423291E+06 0.999587E+05
   2       1     11  2 13 16  1 25 .3427683E+07 000000     240615.     580000.  0.489030E+08 0.985067E+07 0.627456E+06 0.999587E+05
   2       1     11  2 14 17  2 25 .3715749E+07 000000     236286.     380000.  0.487783E+08 0.999818E+07 0.539511E+06 0.999999E+05
   2       1     11  2 15 15  1 25 .4063038E+07 000000     272667.     650000.  0.486301E+08 0.687910E+07 0.423291E+06 0.991378E+05
   2       1     11  2 14 15  2 25 .3744220E+07 000000     264286.     520000.  0.487783E+08 0.700776E+07 0.539511E+06 0.999469E+05
   2       1     11  2 14 17  1 25 .4024263E+07 000000     236286.     510000.  0.487783E+08 0.999241E+07 0.539511E+06 0.999979E+05
   2       1     11  2 14 16  1 25 .3927346E+07 000000     250286.     580000.  0.487783E+08 0.985067E+07 0.539511E+06 0.999587E+05
   2       1     11  2 14 16  1 24 .3429547E+07 000000     240952.     580000.  0.487783E+08 0.987023E+07 0.539511E+06 0.999727E+05
   2       1     12  2 14 16  1 25 .3517043E+07 000000     238857.     490000.  0.498910E+08 0.985067E+07 0.958904E+06 0.999587E+05
   2       1     10  2 14 16  1 25 .1071316E+14 000010     261714.     670000.  0.350661E+08 0.985067E+07 -.462886E+07 0.999587E+05
   2       1     11  3 14 16  1 25 .3553251E+07 000000     250286.     490000.  0.488341E+08 0.985067E+07 0.798972E+06 0.999587E+05
   2       1     11  1 14 16  1 25 .4047831E+13 000010     250286.     670000.  0.472111E+08 0.985067E+07 -.284529E+07 0.999587E+05
   2       1     11  2 15 16  1 25 .3427557E+07 000000     258667.     580000.  0.486301E+08 0.985067E+07 0.423291E+06 0.999587E+05
   2       1     11  2 13 16  1 25 .3427683E+07 000000     240615.     580000.  0.489030E+08 0.985067E+07 0.627456E+06 0.999587E+05
   2       1     11  2 14 17  2 25 .3715749E+07 000000     236286.     380000.  0.487783E+08 0.999818E+07 0.539511E+06 0.999999E+05
   2       1     11  2 14 15  1 25 .4062827E+07 000000     264286.     650000.  0.487783E+08 0.687910E+07 0.539511E+06 0.991378E+05
   2       1     11  2 14 15  2 25 .3744220E+07 000000     264286.     520000.  0.487783E+08 0.700776E+07 0.539511E+06 0.999469E+05
   2       1     11  2 14 17  1 25 .4024263E+07 000000     236286.     510000.  0.487783E+08 0.999241E+07 0.539511E+06 0.999979E+05
   2       1     11  2 14 16  1 25 .3927346E+07 000000     250286.     580000.  0.487783E+08 0.985067E+07 0.539511E+06 0.999587E+05
   2       1     11  2 14 16  1 24 .3429547E+07 000000     240952.     580000.  0.487783E+08 0.987023E+07 0.539511E+06 0.999727E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                 11  2 14 16  1 25      3427346.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 12  1 14 17  1 25  FLAGC =000000
FROM LOCMIN: NEXT GRAD SEARCH POINT: 13  1 14 18  1 25  FLAGC =000000
FROM LOCMIN: THERE ARE  0 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  2 14 17  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  13  2 14 18  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  14  2 14 19  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  15  2 14 20  1 25   FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  16  2 14 21  1 25   FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  17  2 14 22  1 25   FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  18  2 14 23  1 25   FLAG1= 2
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  19  2 14 24  1 25   FLAG1= 2
```

```
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  20  2 14 25  1 25   FLAG1= 2
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   2       1      11  2 14 16  1 25      3427346.           250286.    580000.  0.487783E+08 0.985067E+07 0.539511E+06 0.999587E+05


                                        DESIGN COMBINATION:   EQUIPMENT 1:  2      EQUIPMENT 2:  1

FROM INIT1: INITIAL BASE POINT:                  14    1   10   18    1   17
FROM INIT1: INITIAL PENALTY PARAMETERS:         1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT1: INITIAL LAGRANGE MULTIPLIERS:       0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1  M1 R1 N2 M2 R2   TOTAL COST  FLAGC       APRC        AOPC        ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   2       1     14  1 10 18  1 17  .3869016E+07 000000       5412.     260000.  0.499891E+08 0.999998E+07 0.987944E+06 0.100000E+06
   2       1     15  1 10 18  1 17  .6003467E+08 100000     -10588.     170000.  0.499978E+08 0.999998E+07 0.997616E+06 0.100000E+06
   2       1     13  1 10 18  1 17  .3758874E+07 000000      21412.     350000.  0.499411E+08 0.999998E+07 0.934700E+06 0.100000E+06
   2       1     14  2 10 18  1 17  .3995753E+07 000000       5412.     170000.  0.499994E+08 0.999998E+07 0.999818E+06 0.100000E+06
   2       1     14  1 11 18  1 17  .3865239E+07 000000      25776.     260000.  0.499883E+08 0.999998E+07 0.986651E+06 0.100000E+06
   2       1     14  1 10 19  1 17  .1191420E+09 100000     -15176.     190000.  0.499891E+08 0.100000E+08 0.987944E+06 0.100000E+06
   2       1     14  1 10 17  1 17  .3758207E+07 000000      26000.     330000.  0.499891E+08 0.999924E+07 0.987944E+06 0.999999E+05
   2       1     14  1 10 18  2 17  .4060975E+07 000000       5412.     130000.  0.499891E+08 0.100000E+08 0.987944E+06 0.100000E+06
   2       1     14  1 10 18  1 18  .3862830E+07 000000      26000.     260000.  0.499891E+08 0.999997E+07 0.987944E+06 0.100000E+06
   2       1     13  1 10 17  1 17  .3648065E+07 000000      42000.     420000.  0.499411E+08 0.999924E+07 0.934700E+06 0.999999E+05
   2       1     12  1 10 16  1 17  .3433927E+07 000000      78588.     580000.  0.496564E+08 0.996811E+07 0.619191E+06 0.999996E+05
   2       1     11  1 10 15  1 17  .9873595E+12 000010     115177.     740000.  0.478297E+08 0.857413E+07 -.140525E+07 0.999807E+05
   2       1     13  1 10 17  1 17  .3648065E+07 000000      42000.     420000.  0.499411E+08 0.999924E+07 0.934700E+06 0.999999E+05
   2       1     11  1 10 15  1 17  .9873595E+12 000010     115177.     740000.  0.478297E+08 0.857413E+07 -.140525E+07 0.999807E+05
   2       1     12  2 10 16  1 17  .3555977E+07 000000      78588.     490000.  0.499235E+08 0.996811E+07 0.977063E+06 0.999996E+05
   2       1     12  1 10 16  1 17  .3933927E+07 000000      78588.     580000.  0.496564E+08 0.996811E+07 0.619191E+06 0.999996E+05
   2       1     12  1 11 16  1 17  .3430970E+07 000000      96043.     580000.  0.496404E+08 0.996811E+07 0.590429E+06 0.999996E+05
   2       1     12  1 12 16  1 17  .3428468E+07 000000     110588.     580000.  0.496068E+08 0.996811E+07 0.527749E+06 0.999996E+05
   2       1     12  1 13 16  1 17  .3425975E+07 000000     122896.     580000.  0.495803E+08 0.996811E+07 0.475940E+06 0.999996E+05
   2       1     12  1 15 16  1 17  .3423577E+07 000000     142588.     580000.  0.494064E+08 0.996811E+07 0.911298E+05 0.999996E+05
   2       1     12  1 18 16  1 17  .1557736E+13 000010     163922.     580000.  0.487964E+08 0.996811E+07 -.176507E+07 0.999996E+05
   2       1     12  1 16 16  1 17  .3330115E+11 000010     150588.     580000.  0.492709E+08 0.996811E+07 -.258061E+06 0.999996E+05
   2       1     12  1 14 16  1 17  .3424524E+07 000000     133445.     580000.  0.495036E+08 0.996811E+07 0.315544E+06 0.999996E+05
   2       1     11  1 15 17  1 17  .7314985E+13 000010     132667.     600000.  0.468489E+08 0.999924E+07 -.382491E+07 0.999999E+05
   2       1     13  1 15 15  1 17  .3528341E+07 000000     152510.     560000.  0.498789E+08 0.857413E+07 0.814594E+06 0.999807E+05
   2       1     12  1 15 16  2 17  .3614202E+07 000000     142588.     450000.  0.494064E+08 0.998436E+07 0.911298E+05 0.999999E+05
   2       1     12  1 15 16  1 17  .3923577E+07 000000     142588.     580000.  0.494064E+08 0.996811E+07 0.911298E+05 0.999996E+05
   2       1     12  1 15 16  1 18  .3418888E+07 000000     160889.     580000.  0.494064E+08 0.995824E+07 0.911298E+05 0.999990E+05
   2       1     12  1 15 16  1 19  .3414106E+07 000000     177263.     580000.  0.494064E+08 0.995280E+07 0.911298E+05 0.999986E+05
   2       1     12  1 15 16  1 20  .3409608E+07 000000     192000.     580000.  0.494064E+08 0.994702E+07 0.911298E+05 0.999981E+05
   2       1     12  1 15 16  1 22  .3403201E+07 000000     217455.     580000.  0.494064E+08 0.991297E+07 0.911298E+05 0.999916E+05
   2       1     12  1 15 16  1 25  .3396242E+07 000000     248000.     580000.  0.494064E+08 0.985067E+07 0.911298E+05 0.999587E+05
   2       1     12  1 15 16  1 25  .1589624E+08 000000     248000.     580000.  0.494064E+08 0.985067E+07 0.911298E+05 0.999587E+05
   2       1     12  1 15 16  1 25  .5396242E+07 000000     248000.     580000.  0.494064E+08 0.985067E+07 0.911298E+05 0.999587E+05
   2       1     12  1 15 16  1 24  .3398443E+07 000000     238667.     580000.  0.494064E+08 0.987023E+07 0.911298E+05 0.999727E+05
   2       1     12  1 15 16  1 25  .3896242E+07 000000     248000.     580000.  0.494064E+08 0.985067E+07 0.911298E+05 0.999587E+05
   2       1     13  1 15 17  1 25  .3594313E+07 000000     223333.     420000.  0.498789E+08 0.999241E+07 0.814594E+06 0.999979E+05
   2       1     11  1 15 15  1 25  .7314985E+13 000010     272667.     740000.  0.468489E+08 0.687910E+07 -.382491E+07 0.991378E+05
   2       1     12  2 15 16  1 25  .3514695E+07 000000     248000.     490000.  0.498706E+08 0.985067E+07 0.945514E+06 0.999587E+05
   2       1     12  1 15 16  1 25  .3896242E+07 000000     248000.     580000.  0.494064E+08 0.985067E+07 0.911298E+05 0.999587E+05
   2       1     12  1 16 16  1 25  .3330162E+11 000010     256000.     580000.  0.492709E+08 0.985067E+07 -.258061E+06 0.999587E+05
   2       1     12  1 14 16  1 24  .3399390E+07 000000     229524.     580000.  0.495036E+08 0.987023E+07 0.315544E+06 0.999727E+05
   2       1     11  1 15 17  1 25  .7314985E+13 000010     244667.     600000.  0.468489E+08 0.999241E+07 -.382491E+07 0.999979E+05
   2       1     13  1 15 15  1 25  .3632877E+07 000000     251333.     560000.  0.498789E+08 0.687910E+07 0.814594E+06 0.991378E+05
   2       1     12  1 15 16  2 25  .3581818E+07 000000     248000.     450000.  0.494064E+08 0.992840E+07 0.911298E+05 0.999987E+05
   2       1     12  1 15 16  1 25  .3896242E+07 000000     248000.     580000.  0.494064E+08 0.985067E+07 0.911298E+05 0.999587E+05
   2       1     12  1 14 16  1 25  .3897189E+07 000000     238857.     580000.  0.495036E+08 0.985067E+07 0.315544E+06 0.999587E+05
   2       1     12  1 16 16  1 24  .3330112E+11 000010     246667.     580000.  0.492709E+08 0.987023E+07 -.258061E+06 0.999727E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                 12  1 15 16  1 25       3396242.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 15 17  1 25  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 15 17  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 15 18  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 15 19  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 14 20  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 14 21  1 25   FLAG1= 5
```

```
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 14 22  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 14 23  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 14 24  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 14 25  1 25   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   2       1      12  1 15 16  1 25     3396242.          248000.     580000.  0.494064E+08 0.985067E+07 0.911298E+05 0.999587E+05

                                   DESIGN COMBINATION:   EQUIPMENT 1:  2      EQUIPMENT 2:  1

FROM INIT2: INITIAL BASE POINT:                11    2   20   15    2   25
FROM INIT2: INITIAL PENALTY PARAMETERS:      1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT2: INITIAL LAGRANGE MULTIPLIERS:  0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC       APRC        AOPC        ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   2       1      11  2 20 15  2 25  .1393297E+13 000010     302000.     520000.  0.470002E+08 0.700776E+07 -.166931E+07 0.999469E+05
   2       1      12  2 20 15  2 25  .3825041E+07 000000     294000.     430000.  0.495817E+08 0.700776E+07 0.627813E+06 0.999469E+05
   2       1      11  3 20 15  2 25  .1177088E+11 000010     302000.     430000.  0.472162E+08 0.700776E+07 -.153408E+06 0.999469E+05
   2       1      11  2 19 15  2 25  .4105021E+12 000010     297368.     520000.  0.474769E+08 0.700776E+07 -.906089E+06 0.999469E+05
   2       1      11  2 20 16  2 25  .1393297E+13 000010     288000.     450000.  0.470002E+08 0.992840E+07 -.166931E+07 0.999987E+05
   2       1      11  2 20 15  3 25  .1393297E+13 000010     302000.     390000.  0.470002E+08 0.700896E+07 -.166931E+07 0.999843E+05
   2       1      11  2 20 15  1 25  .1393297E+13 000010     302000.     650000.  0.470002E+08 0.687910E+07 -.166931E+07 0.991378E+05
   2       1      11  2 20 15  2 24  .1393297E+13 000010     293250.     520000.  0.470002E+08 0.720683E+07 -.166931E+07 0.999622E+05
   2       1      12  3 19 15  2 25  .3950133E+07 000000     288947.     340000.  0.497989E+08 0.700776E+07 0.929549E+06 0.999469E+05
   2       1      13  2 20 15  2 25  .3927354E+07 000000     286000.     340000.  0.499368E+08 0.700776E+07 0.943785E+06 0.999469E+05
   2       1      11  2 20 15  2 25  .1393297E+13 000010     302000.     520000.  0.470002E+08 0.700776E+07 -.166931E+07 0.999469E+05
   2       1      12  3 20 15  2 25  .3949056E+07 000000     294000.     340000.  0.497411E+08 0.700776E+07 0.892750E+06 0.999469E+05
   2       1      12  1 20 15  2 25  .1385429E+14 000010     294000.     520000.  0.479622E+08 0.700776E+07 -.526389E+07 0.999469E+05
   2       1      12  2 20 15  2 25  .4325041E+07 000000     294000.     430000.  0.495817E+08 0.700776E+07 0.627813E+06 0.999469E+05
   2       1      12  2 19 15  2 25  .3825431E+07 000000     288947.     430000.  0.496771E+08 0.700776E+07 0.756068E+06 0.999469E+05
   2       1      12  2 20 16  2 25  .3693743E+07 000000     280000.     360000.  0.495817E+08 0.992840E+07 0.627813E+06 0.999987E+05
   2       1      12  2 20 17  2 25  .3796570E+07 000000     266000.     290000.  0.495817E+08 0.999818E+07 0.627813E+06 0.999999E+05
   2       1      12  2 20 16  3 25  .3885620E+07 000000     280000.     230000.  0.495817E+08 0.992941E+07 0.627813E+06 0.999997E+05
   2       1      12  2 20 16  1 25  .3508167E+07 000000     280000.     490000.  0.495817E+08 0.985067E+07 0.627813E+06 0.999587E+05
   2       1      12  2 20 16  1 25  .4008167E+07 000000     280000.     490000.  0.495817E+08 0.985067E+07 0.627813E+06 0.999587E+05
   2       1      12  2 20 16  1 25  .4008167E+07 000000     280000.     490000.  0.495817E+08 0.985067E+07 0.627813E+06 0.999587E+05
   2       1      12  2 20 16  1 24  .3510368E+07 000000     270667.     490000.  0.495817E+08 0.987023E+07 0.627813E+06 0.999727E+05
   2       1      13  2 20 16  1 25  .3610480E+07 000000     272000.     400000.  0.499368E+08 0.985067E+07 0.943785E+06 0.999587E+05
   2       1      11  2 20 16  1 25  .1393297E+13 000010     288000.     580000.  0.470002E+08 0.985067E+07 -.166931E+07 0.999587E+05
   2       1      12  3 20 16  1 25  .3632182E+07 000000     280000.     400000.  0.497411E+08 0.985067E+07 0.892750E+06 0.999587E+05
   2       1      12  1 20 16  1 25  .1385429E+14 000010     280000.     580000.  0.479622E+08 0.985067E+07 -.526389E+07 0.999587E+05
   2       1      12  2 20 16  1 25  .4008167E+07 000000     280000.     490000.  0.495817E+08 0.985067E+07 0.627813E+06 0.999587E+05
   2       1      12  2 19 16  1 25  .3508557E+07 000000     274947.     490000.  0.496771E+08 0.985067E+07 0.756068E+06 0.999587E+05
   2       1      12  2 20 17  2 25  .3796570E+07 000000     266000.     290000.  0.495817E+08 0.999818E+07 0.627813E+06 0.999999E+05
   2       1      12  2 20 15  1 25  .4143648E+07 000000     294000.     560000.  0.495817E+08 0.687910E+07 0.627813E+06 0.991378E+05
   2       1      12  2 20 15  2 25  .3825041E+07 000000     294000.     430000.  0.495817E+08 0.700776E+07 0.627813E+06 0.999469E+05
   2       1      12  2 20 17  1 25  .4105084E+07 000000     266000.     420000.  0.495817E+08 0.999241E+07 0.627813E+06 0.999979E+05
   2       1      12  2 20 16  1 25  .4008167E+07 000000     280000.     490000.  0.495817E+08 0.985067E+07 0.627813E+06 0.999587E+05
   2       1      12  2 20 16  1 24  .3510368E+07 000000     270667.     490000.  0.495817E+08 0.987023E+07 0.627813E+06 0.999727E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                  12  2 20 16  1 25       3508167.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 20 17  1 25  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  2 20 17  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  2 19 18  1 25   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 19 19  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 19 20  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 18 21  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 18 22  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 18 23  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 17 24  1 25   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 17 25  1 25   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   2       1      12  2 20 16  1 25     3508167.          280000.     490000.  0.495817E+08 0.985067E+07 0.627813E+06 0.999587E+05

                                   DESIGN COMBINATION:   EQUIPMENT 1:  2      EQUIPMENT 2:  2
```

```
FROM INIT: INITIAL BASE POINT:              11    2   15   16    2   23
FROM INIT: INITIAL PENALTY PARAMETERS:     1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT: INITIAL LAGRANGE MULTIPLIERS:   0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC      APRC       AOPC        ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
    2       2     11  2 15 16  2 23  .3647877E+07 000000    190493.    530000.  0.486301E+08 0.999034E+07 0.423291E+06 0.100000E+06
    2       2     12  2 15 16  2 23  .3735015E+07 000000    179826.    440000.  0.498706E+08 0.999034E+07 0.945514E+06 0.100000E+06
    2       2     10  2 15 16  2 23  .1601193E+14 000010    201160.    620000.  0.341826E+08 0.999034E+07 -.565896E+07 0.100000E+06
    2       2     11  3 15 16  2 23  .3773591E+07 000000    190493.    440000.  0.486964E+08 0.999034E+07 0.748505E+06 0.100000E+06
    2       2     11  1 15 16  2 23  .7314985E+13 000010    190493.    620000.  0.468489E+08 0.999034E+07 -.382491E+07 0.100000E+06
    2       2     11  2 16 16  2 23  .3649009E+07 000000    197826.    530000.  0.484377E+08 0.999034E+07 0.253543E+06 0.100000E+06
    2       2     11  2 14 16  2 23  .3647666E+07 000000    182112.    530000.  0.487783E+08 0.999034E+07 0.539511E+06 0.100000E+06
    2       2     11  2 15 17  2 23  .3758139E+07 000000    172232.    465000.  0.486301E+08 0.999991E+07 0.423291E+06 0.100000E+06
    2       2     11  2 15 15  2 23  .3626278E+07 000000    208754.    595000.  0.486301E+08 0.890115E+07 0.423291E+06 0.999996E+05
    2       2     11  2 15 16  3 23  .3839833E+07 000000    190493.    400000.  0.486301E+08 0.999039E+07 0.423291E+06 0.100000E+06
    2       2     11  2 15 16  1 23  .3456735E+07 000000    190493.    660000.  0.486301E+08 0.998038E+07 0.423291E+06 0.999999E+05
    2       2     11  2 15 16  2 24  .3644004E+07 000000    202667.    530000.  0.486301E+08 0.998851E+07 0.423291E+06 0.999999E+05
    2       2     11  2 15 16  1 23  .3456735E+07 000000    190493.    660000.  0.486301E+08 0.998038E+07 0.423291E+06 0.999999E+05
    2       2     11  2 15 16  1 23  .3956735E+07 000000    190493.    660000.  0.486301E+08 0.998038E+07 0.423291E+06 0.999999E+05
    2       2     12  2 15 16  1 23  .3543873E+07 000000    179826.    570000.  0.498706E+08 0.998038E+07 0.945514E+06 0.999999E+05
    2       2     10  2 15 16  1 23  .1601193E+14 000010    201160.    750000.  0.341826E+08 0.998038E+07 -.565896E+07 0.999999E+05
    2       2     11  3 15 16  1 23  .3582449E+07 000000    190493.    570000.  0.486964E+08 0.998038E+07 0.748505E+06 0.999999E+05
    2       2     11  1 15 16  1 23  .7314985E+13 000010    190493.    750000.  0.468489E+08 0.998038E+07 -.382491E+07 0.999999E+05
    2       2     11  2 16 16  1 23  .3457867E+07 000000    197826.    660000.  0.484377E+08 0.998038E+07 0.253543E+06 0.999999E+05
    2       2     11  2 14 16  1 23  .3456524E+07 000000    182112.    660000.  0.487783E+08 0.998038E+07 0.539511E+06 0.999999E+05
    2       2     11  2 13 16  1 23  .3456861E+07 000000    172442.    660000.  0.489030E+08 0.998038E+07 0.627456E+06 0.999999E+05
    2       2     11  2 14 17  1 23  .3565991E+07 000000    163851.    595000.  0.487783E+08 0.999963E+07 0.539511E+06 0.100000E+06
    2       2     11  2 14 15  1 23  .3435508E+07 000000    200373.    725000.  0.487783E+08 0.888408E+07 0.539511E+06 0.999942E+05
    2       2     11  2 14 15  1 23  .3935508E+07 000000    200373.    725000.  0.487783E+08 0.888408E+07 0.539511E+06 0.999942E+05
    2       2     11  2 14 15  2 23  .3626067E+07 000000    200373.    595000.  0.487783E+08 0.890115E+07 0.539511E+06 0.999996E+05
    2       2     11  2 14 15  1 23  .3935508E+07 000000    200373.    725000.  0.487783E+08 0.888408E+07 0.539511E+06 0.999942E+05
    2       2     11  2 14 15  1 24  .3440200E+07 000000    211786.    725000.  0.487783E+08 0.878104E+07 0.539511E+06 0.999911E+05
    2       2     11  2 14 15  1 22  .3439488E+07 000000    187922.    725000.  0.487783E+08 0.888408E+07 0.539511E+06 0.999942E+05
    2       2     12  2 14 15  1 23  .3525205E+07 000000    188944.    635000.  0.498910E+08 0.888408E+07 0.958904E+06 0.999942E+05
    2       2     10  2 14 15  1 23  .1071316E+14 000010    211801.    815000.  0.350661E+08 0.888408E+07 -.462886E+07 0.999942E+05
    2       2     11  3 14 15  1 23  .3561413E+07 000000    200373.    635000.  0.488341E+08 0.888408E+07 0.798972E+06 0.999942E+05
    2       2     11  1 14 15  1 23  .4047831E+13 000010    200373.    815000.  0.472111E+08 0.888408E+07 -.284529E+07 0.999942E+05
    2       2     11  2 15 16  1 23  .3456735E+07 000000    190493.    660000.  0.486301E+08 0.998038E+07 0.423291E+06 0.999999E+05
    2       2     11  2 13 15  1 23  .3935845E+07 000000    190702.    725000.  0.489030E+08 0.888408E+07 0.627456E+06 0.999942E+05
    2       2     11  2 13 16  1 23  .3456861E+07 000000    172442.    660000.  0.489030E+08 0.998038E+07 0.627456E+06 0.999999E+05
    2       2     11  2 15 15  1 23  .3935719E+07 000000    208754.    725000.  0.486301E+08 0.888408E+07 0.423291E+06 0.999942E+05
    2       2     11  2 14 15  2 23  .3626067E+07 000000    200373.    595000.  0.487783E+08 0.890115E+07 0.539511E+06 0.999996E+05
    2       2     11  2 14 15  1 23  .3935508E+07 000000    200373.    725000.  0.487783E+08 0.888408E+07 0.539511E+06 0.999942E+05
    2       2     11  2 14 15  1 24  .3440200E+07 000000    211786.    725000.  0.487783E+08 0.878104E+07 0.539511E+06 0.999911E+05
    2       2     11  2 14 15  1 22  .3439488E+07 000000    187922.    725000.  0.487783E+08 0.888408E+07 0.539511E+06 0.999942E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                  11  2 14 15  1 23      3435508.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 12   1 14 15   1 23   FLAGC =000000
FROM LOCMIN: RETURN 5
    2       2     12  1 14 15  1 23  .3405351E+07 000000    188944.    725000.  0.495036E+08 0.888408E+07 0.315544E+06 0.999942E+05
    2       2     13  1 14 15  1 23  .3508041E+07 000000    177516.    635000.  0.499043E+08 0.888408E+07 0.868056E+06 0.999942E+05
    2       2     11  1 14 15  1 23  .4047831E+13 000010    200373.    815000.  0.472111E+08 0.888408E+07 -.284529E+07 0.999942E+05
    2       2     12  2 14 15  1 23  .3525205E+07 000000    188944.    635000.  0.498910E+08 0.888408E+07 0.958904E+06 0.999942E+05
    2       2     12  1 15 15  1 23  .3404404E+07 000000    198087.    725000.  0.494064E+08 0.888408E+07 0.911298E+05 0.999942E+05
    2       2     12  1 14 16  1 23  .3426367E+07 000000    170683.    660000.  0.495036E+08 0.998038E+07 0.315544E+06 0.999999E+05
    2       2     12  1 14 15  2 23  .3595910E+07 000000    188944.    595000.  0.495036E+08 0.890115E+07 0.315544E+06 0.999996E+05
    2       2     12  1 14 15  1 24  .3410043E+07 000000    200357.    725000.  0.495036E+08 0.878104E+07 0.315544E+06 0.999911E+05
    2       2     12  1 14 15  1 22  .3409331E+07 000000    176494.    725000.  0.495036E+08 0.888408E+07 0.315544E+06 0.999942E+05
    2       2     12  1 15 15  1 23  .3404404E+07 000000    198087.    725000.  0.494064E+08 0.888408E+07 0.911298E+05 0.999942E+05
    2       2     12  1 16 15  1 23  .3330113E+11 000010    206087.    725000.  0.492709E+08 0.888408E+07 -.258061E+06 0.999942E+05
    2       2     13  1 15 15  1 23  .3505558E+07 000000    187420.    635000.  0.498789E+08 0.888408E+07 0.814594E+06 0.999942E+05
    2       2     11  1 15 15  1 23  .7314985E+13 000010    208754.    815000.  0.468489E+08 0.888408E+07 -.382491E+07 0.999942E+05
    2       2     12  2 15 15  1 23  .3522857E+07 000000    198087.    635000.  0.498706E+08 0.888408E+07 0.945514E+06 0.999942E+05
    2       2     12  1 15 15  1 23  .3904404E+07 000000    198087.    725000.  0.494064E+08 0.888408E+07 0.911298E+05 0.999942E+05
    2       2     12  1 16 15  1 23  .3330113E+11 000010    206087.    725000.  0.492709E+08 0.888408E+07 -.258061E+06 0.999942E+05
    2       2     12  1 14 15  1 23  .3405351E+07 000000    188944.    725000.  0.495036E+08 0.888408E+07 0.315544E+06 0.999942E+05
    2       2     12  1 15 16  1 23  .3425420E+07 000000    179826.    660000.  0.494064E+08 0.998038E+07 0.911298E+05 0.999999E+05
    2       2     12  1 15 15  1 23  .3904404E+07 000000    198087.    725000.  0.494064E+08 0.888408E+07 0.911298E+05 0.999942E+05
    2       2     12  1 15 15  2 23  .3594963E+07 000000    198087.    595000.  0.494064E+08 0.890115E+07 0.911298E+05 0.999996E+05
    2       2     12  1 15 15  1 23  .3904404E+07 000000    198087.    725000.  0.494064E+08 0.888408E+07 0.911298E+05 0.999942E+05
```

```
   2        2      12  1 15 15  1 24  .3409096E+07 000000     209500.     725000.  0.494064E+08 0.878104E+07 0.911298E+05 0.999911E+05
   2        2      12  1 15 15  1 22  .3408384E+07 000000     185636.     725000.  0.494064E+08 0.888408E+07 0.911298E+05 0.999942E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                   12  1 15 15  1 23        3404404.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 15 15  1 23  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11  2 15 15  1 23   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  2 14 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  3 14 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  4 14 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5 14 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  5 13 15  1 23   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  6 13 15  1 22   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  7 13 15  1 22   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10  7 12 15  1 22   FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   2        2      12  1 15 15  1 23        3404404.              198087.     725000.  0.494064E+08 0.888408E+07 0.911298E+05 0.999942E+05


                                     DESIGN COMBINATION:   EQUIPMENT 1:  2      EQUIPMENT 2:  2

FROM INIT1: INITIAL BASE POINT:                14     1    10    17     1    16
FROM INIT1: INITIAL PENALTY PARAMETERS:       1.0   1.0   1.0   1.0   1.0   1.0
FROM INIT1: INITIAL LAGRANGE MULTIPLIERS:     0.0   0.0   0.0   0.0   0.0   0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC       APRC        AOPC        ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   2        2      14  1 10 17  1 16  .2471328E+10 100000     -70250.     415000.  0.499891E+08 0.999980E+07 0.987944E+06 0.100000E+06
   2        2      15  1 10 17  1 16  .3723439E+10 100000     -86250.     325000.  0.499978E+08 0.999980E+07 0.997616E+06 0.100000E+06
   2        2      13  1 10 17  1 16  .1475218E+10 100000     -54250.     505000.  0.499411E+08 0.999980E+07 0.934700E+06 0.100000E+06
   2        2      14  2 10 17  1 16  .2471455E+10 100000     -70250.     325000.  0.499994E+08 0.999980E+07 0.999818E+06 0.100000E+06
   2        2      14  1 11 17  1 16  .1248116E+10 100000     -49886.     415000.  0.499883E+08 0.999980E+07 0.986651E+06 0.100000E+06
   2        2      14  1 10 18  1 16  .4660023E+10 100000     -96500.     350000.  0.499891E+08 0.100000E+08 0.987944E+06 0.100000E+06
   2        2      14  1 10 16  1 16  .9716851E+09 100000     -44000.     480000.  0.499891E+08 0.998690E+07 0.987944E+06 0.999999E+05
   2        2      14  1 10 17  2 16  .2471520E+10 100000     -70250.     285000.  0.499891E+08 0.999995E+07 0.987944E+06 0.100000E+06
   2        2      14  1 10 17  1 17  .9717914E+09 100000     -44000.     415000.  0.499891E+08 0.999980E+07 0.987944E+06 0.100000E+06
   2        2      13  1 11 16  1 17  .3566752E+07 000000      15615.     570000.  0.499374E+08 0.998690E+07 0.928739E+06 0.999999E+05
   2        2      13  1 12 15  1 18  .3519206E+07 000000      76667.     635000.  0.499296E+08 0.908937E+07 0.915464E+06 0.999979E+05
   2        2      12  1 12 15  1 19  .3918125E+07 000000     108421.     725000.  0.496068E+08 0.898686E+07 0.527749E+06 0.999964E+05
   2        2      14  1 13 16  1 19  .3659054E+07 000000      74008.     480000.  0.499849E+08 0.998381E+07 0.981153E+06 0.999999E+05
   2        2      13  1 11 15  1 17  .4027592E+07 000000      40321.     635000.  0.499374E+08 0.908937E+07 0.928739E+06 0.999979E+05
   2        2      13  2 12 15  1 18  .3644988E+07 000000      76667.     545000.  0.499922E+08 0.908937E+07 0.997450E+06 0.999979E+05
   2        2      13  1 12 15  1 18  .4019206E+07 000000      76667.     635000.  0.499296E+08 0.908937E+07 0.915464E+06 0.999979E+05
   2        2      12  1 12 15  1 18  .3414608E+07 000000      90000.     725000.  0.496068E+08 0.908937E+07 0.527749E+06 0.999979E+05
   2        2      11  1 12 15  1 18  .1724727E+13 000010     103333.     815000.  0.476210E+08 0.908937E+07 -.185727E+07 0.999979E+05
   2        2      12  1 11 15  1 18  .3417110E+07 000000      75455.     725000.  0.496404E+08 0.908937E+07 0.590429E+06 0.999979E+05
   2        2      12  1 13 15  1 18  .3412115E+07 000000     102308.     725000.  0.495803E+08 0.908937E+07 0.475940E+06 0.999979E+05
   2        2      12  1 14 15  1 17  .3915820E+07 000000      92269.     725000.  0.495036E+08 0.908937E+07 0.315544E+06 0.999979E+05
   2        2      12  1 13 15  2 18  .3603138E+07 000000     102308.     595000.  0.495803E+08 0.910078E+07 0.475940E+06 0.999998E+05
   2        2      12  1 13 15  1 18  .3912115E+07 000000     102308.     725000.  0.495803E+08 0.908937E+07 0.475940E+06 0.999979E+05
   2        2      12  1 13 15  1 19  .3915632E+07 000000     120729.     725000.  0.495803E+08 0.898686E+07 0.475940E+06 0.999964E+05
   2        2      12  1 13 16  1 17  .3456431E+07 000000      57014.     660000.  0.495803E+08 0.998690E+07 0.475940E+06 0.999999E+05
   2        2      13  1 14 16  1 19  .3547196E+07 000000      97744.     570000.  0.499043E+08 0.998381E+07 0.868056E+06 0.999999E+05
   2        2      12  1 12 15  1 17  .3919764E+07 000000      69412.     725000.  0.496068E+08 0.908937E+07 0.527749E+06 0.999979E+05
   2        2      12  2 13 15  1 18  .3533072E+07 000000     102308.     635000.  0.499072E+08 0.908937E+07 0.968489E+06 0.999979E+05
   2        2      12  1 13 15  1 18  .3912115E+07 000000     102308.     725000.  0.495803E+08 0.908937E+07 0.475940E+06 0.999979E+05
   2        2      11  1 12 16  1 19  .1724727E+13 000010      99649.     750000.  0.476210E+08 0.998381E+07 -.185727E+07 0.999999E+05
   2        2      13  1 14 15  1 17  .4018510E+07 000000      80840.     635000.  0.499043E+08 0.908937E+07 0.868056E+06 0.999979E+05
   2        2      13  1 12 15  1 18  .3519206E+07 000000      76667.     635000.  0.499296E+08 0.908937E+07 0.915464E+06 0.999979E+05
   2        2      11  1 14 15  1 18  .4047831E+13 000010     124286.     815000.  0.472111E+08 0.908937E+07 -.284529E+07 0.999979E+05
   2        2      12  1 13 15  2 18  .3603138E+07 000000     102308.     595000.  0.495803E+08 0.910078E+07 0.475940E+06 0.999998E+05
   2        2      12  1 13 15  1 18  .3912115E+07 000000     102308.     725000.  0.495803E+08 0.908937E+07 0.475940E+06 0.999979E+05
   2        2      12  1 13 15  1 19  .3915632E+07 000000     120729.     725000.  0.495803E+08 0.898686E+07 0.475940E+06 0.999964E+05
   2        2      12  1 13 16  1 17  .3456431E+07 000000      57014.     660000.  0.495803E+08 0.998690E+07 0.475940E+06 0.999999E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                   12  1 13 15  1 18        3412115.
FROM LOCMIN: RETURN 2
   2        2      12  1 14 15  1 18  .3410664E+07 000000     112857.     725000.  0.495036E+08 0.908937E+07 0.315544E+06 0.999979E+05
   2        2      13  1 14 15  1 18  .3513354E+07 000000     101429.     635000.  0.499043E+08 0.908937E+07 0.868056E+06 0.999979E+05
   2        2      11  1 14 15  1 18  .4047831E+13 000010     124286.     815000.  0.472111E+08 0.908937E+07 -.284529E+07 0.999979E+05
```

```
   2        2      12  2 14 15  1 18  .3530518E+07 000000    112857.     635000.  0.498910E+08 0.908937E+07 0.958904E+06 0.999979E+05
   2        2      12  1 15 15  1 18  .3409717E+07 000000    122000.     725000.  0.494064E+08 0.908937E+07 0.911298E+05 0.999979E+05
   2        2      12  1 14 16  1 18  .3449481E+07 000000     89524.     660000.  0.495036E+08 0.998690E+07 0.315544E+06 0.999999E+05
   2        2      12  1 14 15  2 18  .3601687E+07 000000    112857.     595000.  0.495036E+08 0.910078E+07 0.315544E+06 0.999998E+05
   2        2      12  1 14 15  1 19  .3414181E+07 000000    131278.     725000.  0.495036E+08 0.898686E+07 0.315544E+06 0.999964E+05
   2        2      12  1 14 15  1 17  .3415820E+07 000000     92269.     725000.  0.495036E+08 0.908937E+07 0.315544E+06 0.999979E+05
   2        2      12  1 15 15  1 18  .3409717E+07 000000    122000.     725000.  0.494064E+08 0.908937E+07 0.911298E+05 0.999979E+05
   2        2      12  1 16 15  1 18  .3330114E+11 000010    130000.     725000.  0.492709E+08 0.908937E+07 -.258061E+06 0.999979E+05
   2        2      13  1 15 15  1 18  .3510871E+07 000000    111333.     635000.  0.498789E+08 0.908937E+07 0.814594E+06 0.999979E+05
   2        2      11  1 15 15  1 18  .7314985E+13 000010    132667.     815000.  0.468489E+08 0.908937E+07 -.382491E+07 0.999979E+05
   2        2      12  2 15 15  1 18  .3528170E+07 000000    122000.     635000.  0.498706E+08 0.908937E+07 0.945514E+06 0.999979E+05
   2        2      12  1 15 15  1 18  .3909717E+07 000000    122000.     725000.  0.494064E+08 0.908937E+07 0.911298E+05 0.999979E+05
   2        2      12  1 16 15  1 18  .3330114E+11 000010    130000.     725000.  0.492709E+08 0.908937E+07 -.258061E+06 0.999979E+05
   2        2      12  1 14 15  1 18  .3410664E+07 000000    112857.     725000.  0.495036E+08 0.908937E+07 0.315544E+06 0.999979E+05
   2        2      12  1 15 16  1 18  .3448534E+07 000000     98667.     660000.  0.494064E+08 0.998690E+07 0.911298E+05 0.999999E+05
   2        2      12  1 15 15  1 18  .3909717E+07 000000    122000.     725000.  0.494064E+08 0.908937E+07 0.911298E+05 0.999979E+05
   2        2      12  1 15 15  2 18  .3600740E+07 000000    122000.     595000.  0.494064E+08 0.910078E+07 0.911298E+05 0.999998E+05
   2        2      12  1 15 15  1 18  .3909717E+07 000000    122000.     725000.  0.494064E+08 0.908937E+07 0.911298E+05 0.999979E+05
   2        2      12  1 15 15  1 19  .3413234E+07 000000    140421.     725000.  0.494064E+08 0.898686E+07 0.911298E+05 0.999964E+05
   2        2      12  1 15 15  1 17  .3414873E+07 000000    101412.     725000.  0.494064E+08 0.908937E+07 0.911298E+05 0.999979E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                   12  1 15 15  1 18       3409717.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 15 15  1 18  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  11   2 15 15  1 18    FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10   2 14 15  1 18    FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10   3 14 15  1 18    FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10   4 14 15  1 18    FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10   5 14 15  1 18    FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10   5 13 15  1 18    FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10   6 13 15  1 19    FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10   7 13 15  1 19    FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  10   7 12 15  1 19    FLAG1= 5
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
   2        2      12  1 15 15  1 18       3409717.          122000.     725000.  0.494064E+08 0.908937E+07 0.911298E+05 0.999979E+05


                         DESIGN COMBINATION:   EQUIPMENT 1:  2      EQUIPMENT 2:  2

FROM INIT2: INITIAL BASE POINT:               11    2   20   15    2   30
FROM INIT2: INITIAL PENALTY PARAMETERS:      1.0  1.0  1.0  1.0  1.0  1.0
FROM INIT2: INITIAL LAGRANGE MULTIPLIERS:  0.0  0.0  0.0  0.0  0.0  0.0

 |<***** DECISION VARIABLES ****>|  |<*************** SYSTEM PERFORMANCE MEASURES AND CONSTRAINT FUNCTION VALUES ***************>|
 DSGN(1) DSGN(2) N1 M1 R1 N2 M2 R2   TOTAL COST  FLAGC        APRC        AOPC        ASUNA(1)     ASUNA(2)     ACUNA(1)     ACUNA(2)
   2        2      11  2 20 15  2 30  .1393297E+13 000010    302000.     595000.  0.470002E+08 0.820297E+07 -.166931E+07 0.999958E+05
   2        2      12  2 20 15  2 30  .3741800E+07 000000    294000.     505000.  0.495817E+08 0.820297E+07 0.627813E+06 0.999958E+05
   2        2      11  3 20 15  2 30  .1177080E+11 000010    302000.     505000.  0.472162E+08 0.820297E+07 -.153408E+06 0.999958E+05
   2        2      11  2 19 15  2 30  .4105021E+12 000010    297368.     595000.  0.474769E+08 0.820297E+07 -.906089E+06 0.999958E+05
   2        2      11  2 20 16  2 30  .1393297E+13 000010    288000.     530000.  0.470002E+08 0.997417E+07 -.166931E+07 0.999999E+05
   2        2      11  2 20 15  3 30  .1393297E+13 000010    302000.     465000.  0.470002E+08 0.820324E+07 -.166931E+07 0.999987E+05
   2        2      11  2 20 15  1 30  .1393297E+13 000010    302000.     725000.  0.470002E+08 0.815705E+07 -.166931E+07 0.999325E+05
   2        2      11  2 20 15  2 29  .1393297E+13 000010    294759.     595000.  0.470002E+08 0.830266E+07 -.166931E+07 0.999968E+05
   2        2      12  3 19 15  2 30  .3866892E+07 000000    288947.     415000.  0.497989E+08 0.820297E+07 0.929549E+06 0.999958E+05
   2        2      13  2 20 15  2 30  .3844113E+07 000000    286000.     415000.  0.499368E+08 0.820297E+07 0.943785E+06 0.999958E+05
   2        2      11  2 20 15  2 30  .1393297E+13 000010    302000.     595000.  0.470002E+08 0.820297E+07 -.166931E+07 0.999958E+05
   2        2      12  3 20 15  2 30  .3865815E+07 000000    294000.     415000.  0.497411E+08 0.820297E+07 0.892750E+06 0.999958E+05
   2        2      12  1 20 15  2 30  .1385429E+14 000010    294000.     595000.  0.479622E+08 0.820297E+07 -.526389E+07 0.999958E+05
   2        2      12  2 20 15  2 30  .4241800E+07 000000    294000.     505000.  0.495817E+08 0.820297E+07 0.627813E+06 0.999958E+05
   2        2      12  2 19 15  2 30  .3742190E+07 000000    288947.     505000.  0.496771E+08 0.820297E+07 0.756068E+06 0.999958E+05
   2        2      12  2 20 16  2 30  .3705894E+07 000000    280000.     440000.  0.495817E+08 0.997417E+07 0.627813E+06 0.999999E+05
   2        2      12  2 20 17  2 30  .3813358E+07 000000    266000.     375000.  0.495817E+08 0.999961E+07 0.627813E+06 0.100000E+06
   2        2      12  2 20 16  3 30  .3897836E+07 000000    280000.     310000.  0.495817E+08 0.997439E+07 0.627813E+06 0.999999E+05
   2        2      12  2 20 16  1 30  .3516163E+07 000000    280000.     570000.  0.495817E+08 0.994702E+07 0.627813E+06 0.999981E+05
   2        2      12  2 20 16  1 30  .4016163E+07 000000    280000.     570000.  0.495817E+08 0.994702E+07 0.627813E+06 0.999981E+05
   2        2      12  2 20 16  1 30  .4016163E+07 000000    280000.     570000.  0.495817E+08 0.994702E+07 0.627813E+06 0.999981E+05
   2        2      12  2 20 16  1 29  .3518544E+07 000000    272276.     570000.  0.495817E+08 0.995280E+07 0.627813E+06 0.999986E+05
   2        2      13  2 20 16  1 30  .3618476E+07 000000    272000.     480000.  0.499368E+08 0.994702E+07 0.943785E+06 0.999981E+05
   2        2      11  2 20 16  1 30  .1393297E+13 000010    288000.     660000.  0.470002E+08 0.994702E+07 -.166931E+07 0.999981E+05
   2        2      12  3 20 16  1 30  .3640178E+07 000000    280000.     480000.  0.497411E+08 0.994702E+07 0.892750E+06 0.999981E+05
```

```
  2       2      12  1 20 16  1 30  .1385429E+14 000010     280000.     660000.  0.479622E+08 0.994702E+07 -.526389E+07 0.999981E+05
  2       2      12  2 20 16  1 30  .4016163E+07 000000     280000.     570000.  0.495817E+08 0.994702E+07 0.627813E+06 0.999981E+05
  2       2      12  2 19 16  1 30  .3516553E+07 000000     274947.     570000.  0.496771E+08 0.994702E+07 0.756068E+06 0.999981E+05
  2       2      12  2 20 17  2 30  .3813358E+07 000000     266000.     375000.  0.495817E+08 0.999961E+07 0.627813E+06 0.100000E+06
  2       2      12  2 20 15  1 30  .4053611E+07 000000     294000.     635000.  0.495817E+08 0.815705E+07 0.627813E+06 0.999325E+05
  2       2      12  2 20 15  2 30  .3741800E+07 000000     294000.     505000.  0.495817E+08 0.820297E+07 0.627813E+06 0.999958E+05
  2       2      12  2 20 17  1 30  .4121498E+07 000000     266000.     505000.  0.495817E+08 0.999838E+07 0.627813E+06 0.999999E+05
  2       2      12  2 20 16  1 30  .4016163E+07 000000     280000.     570000.  0.495817E+08 0.994702E+07 0.627813E+06 0.999981E+05
  2       2      12  2 20 16  1 29  .3518544E+07 000000     272276.     570000.  0.495817E+08 0.995280E+07 0.627813E+06 0.999986E+05
FROM INNER: ROSENBROCKS SEARCH HAS NOT MOVED FROM THE PREVIOUS BASE POINT - DISPLAY CURRENT BEST SOLUTION AND GO TO LOCMIN.
                 12  2 20 16  1 30       3516163.
FROM LOCMIN: NEXT GRAD SEARCH POINT: 11  1 20 15  1 30  FLAGC =000010
FROM LOCMIN: THERE ARE  1 BINDING IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: THERE ARE  1 BINDING EXPLICIT AND IMPLICIT CONSTRAINTS IN M.
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  2 20 15  1 30   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 19 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 19 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 18 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 18 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 17 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  12  1 17 15  1 30   FLAG1= 5
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  13  1 16 15  1 30   FLAG1= 0
FROM LOCMIN: NEXT PROJECTED GRADIENT SEARCH POINT:  13  1 16 15  1 30   FLAG1= 0
FROM LOCMIN: RETURN 9

FROM RESULT: FINAL SOLUTION FOR THIS DESIGN COMBINATION IS:
  2       2      12  2 20 16  1 30       3516163.           280000.     570000.  0.495817E+08 0.994702E+07 0.627813E+06 0.999981E+05


                          SUMMARY OF RESULTS FROM THE MREAL1 OPTIMIZER

OPTIMAL SOLUTIONS FOR EACH DESIGN COMBINATION:

 DESIGN(1) DESIGN(2) N1  M1  R1  N2  M2  R2   TOTAL COST      APRC        AOPC       ASUNA(1)     ASUNA(2)     ACUNA(1)     ASUNA(2)
    1         1      12  1 11 16  1 25 .3513818E+07    223273.     460000. 0.493837E+08 0.985067E+07 0.353311E+05 0.999587E+05
    1         2      12  1 11 16  1 30 .3521814E+07    223273.     540000. 0.493837E+08 0.994702E+07 0.353311E+05 0.999981E+05
    2         1      12  1 15 16  1 25 .3396242E+07    248000.     580000. 0.494064E+08 0.985067E+07 0.911298E+05 0.999587E+05
    2         2      12  1 15 15  1 23 .3404404E+07    198087.     725000. 0.494064E+08 0.888408E+07 0.911298E+05 0.999942E+05


FROM MANOUT: THE OPTIMAL SOLUTION TO PROBLEM MREAL1 (TEST) IS:
 DESIGN(1) DESIGN(2) N1  M1  R1  N2  M2  R2   TOTAL COST      APRC        AOPC       ASUNA(1)     ASUNA(2)     ACUNA(1)     ASUNA(2)
    2         1      12  1 15 16  1 25 .3396242E+07    248000.     580000. 0.494064E+08 0.985067E+07 0.911298E+05 0.999587E+05

                          DETAILED SUMMARY OF THE OPTIMAL SOLUTION

                ANNUAL REPLACEMENT  ANNUAL OPERATING  EXPECTED NUMBER  PROBABILITY OF N
REAL SYSTEM        CAPITAL COST           COST         OF SHORTAGES    OR MORE SHORTAGES  N

BUS DESIGN 2        128000.00         1170000.00       0.005507         0.00008433        3
RAILCAR DESIGN 1    224000.00         1250000.00       0.002285         0.00000000        5
SYSTEMS TOTAL:      352000.00         2420000.00
BUDGET:             600000.00         3000000.00
```

The vita has been removed from the scanned document