THE SPARE MODEL
(SPARES PRIORITIZATION AND AVAILABILITY TO RESOURCE EVALUATION)
USERS GUIDE

Report NS901R1

March 1990

Robert C. Kline
Craig C. Sherbrooke
T. J. O'Malley

Prepared pursuant to Department of Defense Contract MDA903-85-C-0139.

LOGISTICS MANAGEMENT INSTITUTE
6400 Goldsboro Road
Bethesda, Maryland 20817-5886

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION / AVAILABILITY OF REPORT "A" Approved for public release; distribution unlimited.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) LMI-NS901R1	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Logistics Management Institute 6b. OFFICE SYMBOL *(If applicable)*	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS *(City, State, and ZIP Code)* 6400 Goldsboro Road Bethesda, Maryland 20817-5886	7b. ADDRESS *(City, State, and ZIP Code)*
8a. NAME OF FUNDING / SPONSORING ORGANIZATION NASA 8b. OFFICE SYMBOL *(if applicable)* SSFPO	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER MDA903-85-C-0139
8c. ADDRESS *(City, State, and ZIP Code)* 10701 Parkridge Boulevard Reston, VA 22091	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
The SPARE Model (Spares Prioritization and Availability to Resource Evaluation) Users Guide

12. PERSONAL AUTHOR(S)
Robert C. Kline, Craig C. Sherbrooke, T. J. O'Malley

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Final	FROM ______ TO ______	1990, March	35

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUB-GROUP	inventory management, space station, optimization, availability

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

At the request of the National Aeronautics and Space Administration (NASA), the Logistics Management Institute developed a methodology that estimates the optimal orbital replaceable unit (ORU) spares inventory for the Space Station Freedom. That methodology selects spares for the inventory to maximize station availability – the probability that no ORU had more demands during a resupply cycle than it had spares to satisfy those demands. It uses a marginal analysis approach. Spares are ranked in order of decreasing benefit per cost (essentially the improvement provided to station availability per dollar), and added, in that order, to the inventory until a target resource expenditure or station availability is reached. To demonstrate that methodology, we developed the prototype Spares Prioritization and Availability to Resource Evaluation (SPARE) model. This users guide describes how we implemented the methodology and how a user can operate the prototype model on a personal computer.

The SPARE model prototype is restricted to estimating the optimal mix of life-critical ORU spares stored on board the station. That optimal mix ensures the maximum station availability for the limiting resource expenditure. The limiting resource expenditure can be a single resource such as dollars, weight, or storage volume or it can be a combination of those resources. Unlike other models that present a single point solution, the SPARE model presents the maximum availability for an entire range of resource expenditures, a station resource-versus-availability curve. Thus, if the model user wishes to determine what is the best mix of spares for a specific budget, the model will determine not only the mix and the station availability associated with that mix but also how the availability and mix will change if the budget changes.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE *(Include Area Code)* / 22c. OFFICE SYMBOL

LMI

Executive Summary

THE SPARE MODEL USERS GUIDE

The National Aeronautics and Space Administration (NASA) is committed to a program of a permanently manned space station in earth orbit to serve as a stepping stone for human space exploration and to act as an orbital research laboratory. That program – Space Station Freedom – envisions a space station with a planned life cycle of 30 years. Because there will be long intervals between resupply flights from earth, the station must carry its own supply of spares to replace parts which fail during station operations.

At the request of NASA, the Logistics Management Institute developed a methodology that estimates the optimal orbital replaceable unit (ORU) spares inventory for the Space Station Freedom. That methodology selects spares for the inventory to maximize station availability – the probability that no ORU had more demands during a resupply cycle than it had spares to satisfy those demands. It uses a marginal analysis approach. Spares are ranked in order of decreasing benefit per cost (essentially the improvement provided to station availability per dollar), and added, in that order, to the inventory until a target resource expenditure or station availability is reached. To demonstrate that methodology, we developed the prototype Spares Prioritization and Availability to Resource Evaluation (SPARE) model. This users guide describes how we implemented the methodology and how a user can operate the prototype model on a personal computer.

The SPARE model prototype is restricted to estimating the optimal mix of life-critical ORU spares stored on board the station. That optimal mix ensures the maximum station availability for the limiting resource expenditure. The limiting resource expenditure can be a single resource such as dollars, weight, or storage volume or it can be a combination of those resources. Unlike other models that present a single point solution, the SPARE model presents the maximum availability for an entire range of resource expenditures, a station resource-versus-availability curve. Thus, if the model user wishes to determine what is the best mix of spares for a specific budget, the model will determine not only the mix and the station availability associated with that mix but also how the availability and mix will change if the budget changes.

In this users guide, we describe the SPARE model methodology, why that methodology is used, and the algorithms within the computer code. We offer further guidance to the user for installing and operating the model, we describe the model's data inputs and how to modify them, and we present the data outputs and what they mean. In addition, this guide describes how to modify model assumptions to perform various types of analyses. We also describe potential enhancements to the model to consider ground sparing, noncritical ORUs, and redundancies of subsystems and ORUs.

CONTENTS

CONTENTS (Continued)

TABLES

FIGURES

CHAPTER 1

INTRODUCTION

The prototype Spares Prioritization and Availability to Resource Evaluation (SPARE) model performs several functions. First, it specifies the entire range, a curve, of how station availability changes as limiting resource expenditures change. Every point on that curve corresponds to an optimal spares mix that maximizes station availability (see Figure 1). Second, the model develops a list of initial spares based upon a user-specified station availability target or limiting resource target. Station availability is defined as the probability that no critical orbital replaceable unit (ORU) is missing a spare at the end of a logistics cycle (e.g., 90 days). The limiting resource is user specified and can be a single resource such as dollars, weight, or storage volume or a combination of resources. Currently, the model determines the on-orbit spares mix for critical ORUs [maintenance criticality codes (MCCs) of 1 or 1R] and does not consider the levels of redundancy of the space station.

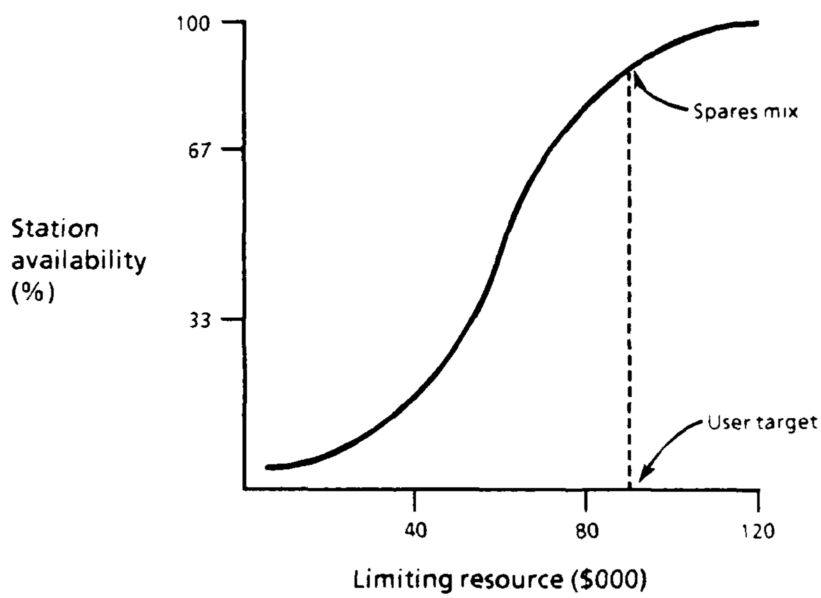

FIG. 1. STATION RESOURCE-VERSUS-AVAILABILITY CURVE
(Critical on-orbit ORUs)

We begin this guide with a discussion of how to install the model (Chapter 2) on your personal computer (PC) (the model requires an IBM-compatible computer with at least 400 kilobytes memory and a 1 megabyte hard disk), and then we guide you through a test drive. That test drive, which is explained in the next chapter, will take about 5 minutes. Once you have finished the test, we step back and discuss the details of the model: the reason we use this approach for sparing (Chapter 3), the detailed description of the approach with algorithms and examples (Chapter 4), the input data required for the model and how to modify them (Chapter 5), the model's various options and how to modify them (Chapter 6), the steps required to run the model (Chapter 7), the model outputs and how to interpret them (Chapter 8), and the possible model enhancements and uses (Chapter 9).

CHAPTER 2

INSTALLATION

To install the SPARE model, log on to your PC and then insert our floppy disk into your A drive. The next commands will make a new subdirectory called SPARE on your C drive and will copy the required files from our floppy disk to the new directory. [*Note:* All characters enclosed in double quotations are the commands you enter from your PC keyboard followed by a return.]

- Enter "A:INSTALL" to install the model.

QUICK TEST DRIVE

Now that the model is installed on your hard disk, we will show you how easy it is to run and some of what it can do. The test drive of our model uses a sample data base and requires the following inputs to the PC:

- Enter "CD\SPARE" to move to the C:\SPARE subdirectory.
- Enter "RUNSPARE" to run the SPARE model.

The model then asks you for three basic inputs:

- Enter "0" to select price as the resource type.
- Enter "0" to select resources as the target type.
- Enter "82000" for the target value. In this case, you have set the model so that it will produce the maximum station availability for a budget (the resource) of $82,000.

After about 15 seconds (45 seconds for 286 PCs), a plot should appear on the screen (a return ends the model run). The plot presents the resource (dollars)-versus-availability curve, and the target point you selected ($82,000) (shaded area) defines the initial spares list. That list and the other model data are in a file that you can browse with your editor, word processor, or LOTUS 1-2-3. In fact, all data files are text files that you can browse. We will discuss those files in turn, but first let us start from the beginning and explain what you have just done.

INSTALLED FILES DESCRIPTION

The installation program copied the following files to your PC from the floppy disk:

- *C:\SPARE\RUNSPARE.EXE:* This file is the SPARE model in an executable form.
- *C:\SPARE\SPAREIN.RPT:* This is a text file that contains sample input data for 154 ORUs from the Model for Estimating Space Station Operations Costs (MESSOC) data base.
- *C:\SPARE\OPTIONS.RPT:* This file is the options text file that allows you to change detailed model options without recompiling a new executable file.
- *C:\SPARE*.WK1:* These files are similar to the previous two files but are compatible with LOTUS 1-2-3.
- *C:\SPARE\OUT.RPT:* This file is the text output file in which all model results are stored.
- *C:\SPARE*.BGI and C:\SPARE*.CHR:* These files are software drivers so that the resource-versus-availability curve can be plotted on different PC monitors.
- *C:\SPARE*.PAS:* These files contain the source code (Turbo Pascal 5.0) for the model.

We describe the pertinent user files in more detail in the subsequent chapters of this guide.

CHAPTER 3

WHY THIS APPROACH

Now that the model is installed, you may ask what this method does for you that other approaches do not. The best way to answer that question is with the simple example given in Table 1. The table compares two ways of selecting spares for a hypothetical station that contains only two ORUs. The first way treats all ORUs the same. It selects sufficient spares so each ORU has a 98 percent chance of having at least as many spares as demands (see top of Table 1). That is, the probability of sufficiency (POS) is 0.98 for all ORUs. The space station availability is the product of the two POS values, or 0.96. The station limiting resource expenditure is $13: $3 for 3 spares of ORU A at $1 each and $10 for 2 spares of ORU B at $5 each.

TABLE 1

A BETTER WAY TO DETERMINE SPARES

	ORU A Cost = $1	ORU B Cost = $5	Space station availability
Individual POS Targets			
POS	0.98	0.98	0.96
Number of spares	3	2	----
Cost (dollars)	3	10	13
A Better Way			
POS	0.99	0.97	0.96
Number of spares	4	1	----
Cost (dollars)	4	5	9

However, there is a better way to determine a spares mix. The approach focuses on how well the station – as a system of ORUs – is performing as opposed to how well each ORU is performing. Thus, the approach does not treat all ORUs the same but considers that their costs (limiting resource) are different. Table 1 (bottom section) demonstrates that by slightly changing the spares mix for ORUs A and B to 4 and 1, respectively, you can obtain the same station availability as before (0.96) but decrease the total cost from $13 to $9. That is similar to how the SPARE model selects spares. It prioritizes spares and selects the spare that will improve the station availability the most per dollar.

To prove our point, we evaluated the MESSOC data base using the first approach and selected spares so that each ORU's POS is 0.98 or more. The result is that the total spares budget is $82,000 and the station availability is approximately 30 percent. As you demonstrated in the test drive, for the same $82,000, the model selected spares that produced a station availability of 89 percent. In other words, at the end of the logistics cycle, the model spares would be three times more likely to have sufficient spares than the individual approach.

CHAPTER 4

MODEL DESCRIPTION

The crux of the model is the process that sets priorities for spares. That process establishes spares priorities for a "shopping list" based upon each spare's benefit-to-cost ratio (see Table 2). At the top of the shopping list is the spare that has the greatest marginal benefit to station availability per resource expenditure – that is, the spare with the biggest "bang for buck." Selecting from the list in the order indicated yields the maximum availability rate for the resource expended. The optimization process assures that no other combination of spares will give a higher availability for the same resource expenditure or the same availability for less resource expenditure. Each spare the model selects from the list causes the station availability and resources expended to increase. The entire selection process creates the resource-versus-availability curve that was generated in our test drive, and each spare selected creates a point on that curve.

TABLE 2

EXAMPLE OF SPACE STATION AVAILABILITY COST CURVE

Curve		ORU spares prioritized (shopping list)	*Bang for buck:* Marginal benefit / Cost	Unit resource (dollars)
Total resource (dollars)	Space station availability (percent)			
0	22.8	----	----	----
1	39.6	Filter	0.554	1
2	45.8	Filter	0.146	1
7	79.8	Valve	0.111	5
10	82.4	Sensor	0.033	3
⋮				

ALGORITHM DEFINITION

The station availability is the probability that no critical ORU is missing a spare at the end of a logistics cycle. We assume the station starts a cycle after a resupply from the shuttle with a full complement of spares. These spares are intended to satisfy demands over the logistics cycle until the next shuttle provides resupply. (Current planning envisions a cycle of 90 days, but cycle time may be varied by the model user.) The station availability is an indication of how well the entire space station might perform its mission given a certain spares mix. By tying spares level directly to station availability, the user gets a more meaningful measure upon which to base sparing decisions. This is not to say the model produces an all-inclusive view of station availability. It does not consider other important factors – crew availability, for example. However, it does capture the supply aspect of station availability due to on-orbit spares of critical ORUs.

Availability is calculated as the product of the individual ORU POS. The POS of a single ORU is the probability that the ORU will have at least as many spares (S_i) as demands for spares at the end of the logistics cycle (i.e., the time between shuttle flights, currently set at 90 days). The ORU demands are approximated by a Poisson distribution. The equation that expresses those relations is as follows:

$$\text{Station Availability} = \prod_{ORU_i} POS_i(S_i) \qquad \text{[Eq. 1]}$$

where:

$$POS_i(S_i) = \sum_{0}^{S_i} \frac{(DDR_i \times LC)^{S_i} \times exp - (DDR_i \times LC)}{S_i!},$$

and

S_i = number of spares for ORU_i,

DDR_i = daily demand rates for ORU_i, and

LC = logistics cycle.

HOW IT WORKS

The first step required for developing our shopping list is to determine the bang-for-buck measure (i.e., the marginal benefit to station availability divided by unit resource expenditure) for each spare. The SPARE model starts the process with the station availability equation, Equation 1. The model takes the natural logarithm (ln) of each side of that equation. Because of the multiplicative nature of the availability equation, that step ensures the mathematical accuracy of the optimization. [See B. L. Fox, "Discrete Optimization via Marginal Analysis," *Management Science*, Series A13 (1966), pp. 210 – 216.] Also, the model sets ORU spares levels to zero to obtain a starting availability position with no resource expenditures:

$$ln\,(Station\ Availability) = \sum_{ORU_i} ln[POS_i(0)] \qquad \text{[Eq. 2]}$$

Next, the model estimates the marginal benefit for the next spare of each ORU. That benefit is the difference between the new benefit of the ORU and the old benefit of the ORU:

$$Marginal\ Benefit = ln[POS_i(1)] - ln[POS_i(0)] \qquad \text{[Eq. 3]}$$

By dividing that benefit by the ORU's unit resource expenditure, the model obtains the bang-for-buck measure required to set priorities for the ORU's spare. That measure is generalized in the following algorithm:

$$\frac{Marginal\ Benefit_i}{Unit\ Resource_i} = \frac{ln[POS_i(S+1)] - ln[POS_i(S)]}{Unit\ Resource_i} \qquad \text{[Eq. 4]}$$

EXAMPLE OF THE PRIORITIZATION PROCESS

To explain the prioritization process, we use a simple example for a handful of ORUs and assume price (dollars) is the resource type. We start the process by solving Equation 2 for the station availability with no resource expenditures. In the example given in Table 1, that value is equal to 22.8 percent.

Next, the model checks all ORUs and chooses the one with the largest marginal benefit-to-cost ratio (i.e., the filter with a ratio of 0.554). It then increases the station availability by the marginal benefit to 39.6 percent and increases the total resource expenditure by the $1 unit cost of the filter. The selection of the filter creates the second point on our curve (first two columns of Table 1). Next, the model calculates a new benefit-to-cost ratio for the filter using Equation 4; now, the marginal benefit is the difference between obtaining the second and the first spare (i.e., 0.146). At that point the process is repeated: the model selects the spare with the next biggest bang for buck (again, the filter); increments the resource expenditure and station availability to $2 and 45.8 percent, respectively; and recalculates a new bang-for-buck ratio for the third filter. The model continues to repeat that process until the space station availability is close to 100 percent.

A final feature of the model is its ability to generate the initial spares mix for a user-specified resource or availability target. To do that, the model checks each point as it processes the curve. When it reaches the point at which the curve value first becomes greater than the target value, it stops and stores the spares level for each ORU. In our example, if we want the spares mix for a resource target of $9, the model solution is to select two filters, one valve, and one sensor. Those spares levels would then be stored in the model output file (described subsequently). [*Note:* The model solution ($10) is slightly larger than the user target ($9) because the model must select whole spares.]

MULTIPLE RESOURCE OPTIMIZATION

So far in our examples, the limiting unit resource used is the unit price of the ORU. However, the model can handle unit weight, unit volume, or a linear combination of the individual resources as the limiting unit resource. Through the use of linear combination of resources, the model performs a multiple criteria optimization and can balance possible conflicting resource utilization. Currently, the value for the unit resource in Equation 4 is estimated with the linear combination shown in Equation 5:

$$Unit\,Resource\; i = (coefficient \times price) + [(1 - coefficient) \times weight] \qquad \text{[Eq. 5]}$$

If the user wishes to produce a spares mix for the minimal total cost (dollars), the case for the test drive, the model sets the coefficient to 1 and ignores the weight of the ORU. Conversely, if the user wishes to produce a spares mix for a minimal total weight, the model sets the coefficient to 0 and ignores the cost of the ORU. If you wish to use a combination of the two, enter a coefficient between 0 and 1. As the coefficient value changes over this range, the relative importance of price to weight changes proportionately. Through the use of the coefficient, the user can perform multicriteria optimization for two resources.

To demonstrate the multicriteria optimization, we run the model five times with the coefficient set to 1, 0.5, 0.3, 0.1, and 0.0, respectively. For each run, the user-specified target is set to a 90 percent availability. Figure 2 displays the plot of total cost versus weight for the five runs.

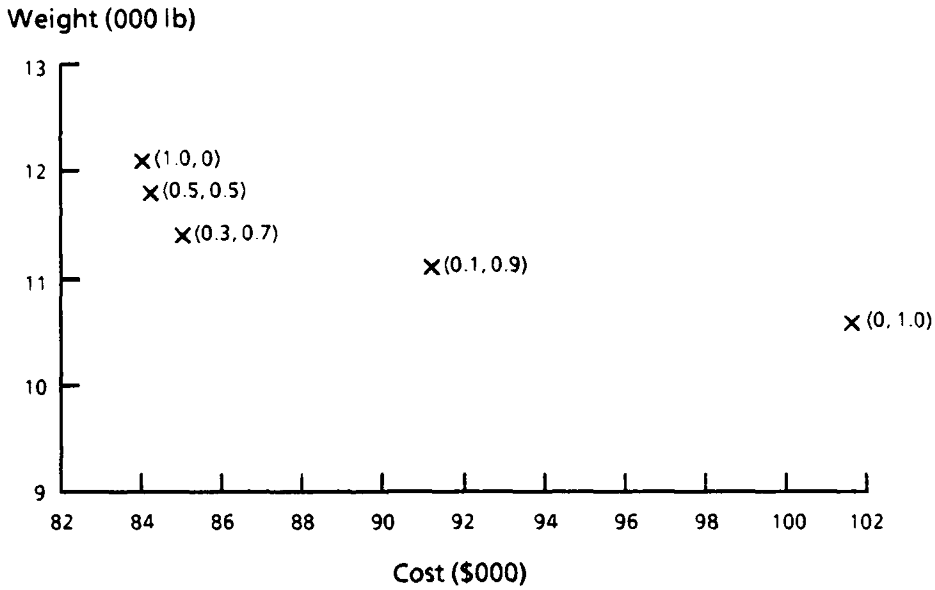

FIG. 2. COST VERSUS WEIGHT AT 90 PERCENT STATION AVAILABILITY
(Coefficient shown in parentheses)

Similar to the resource-versus-availability curve, the plot in Figure 2 describes the range of tradeoffs between cost and weight. The user then can determine which tradeoff is most appropriate. For instance, we can attain a fairly low cost solution without totally sacrificing station weight. The point associated with a coefficient of 0.3 offers that balance. By increasing the minimal total cost of the spares mix by 0.9 percent (from $83,716 with a coefficient of 1.0 to $84,495 with a coefficient of 0.3) the total weight improves 6 percent (from 12,119 lb down to 11,395 lb).

Each point in Figure 2 is an undominated solution. That is, for a point's total cost and weight, no other spares mix will produce a higher station availability. Any mix with higher station availability will have higher cost or higher weight (or both). In the model enhancements chapter, we discuss an automated way to obtain Figure 2 and have an additional user target or constraint.

CHAPTER 5

MODEL INPUT DATA

Initially, the most difficult part of the model processing is the creation of the input file (SPAREIN.RPT). That file defines all the ORUs and their characteristics (daily demand, cost, weight, etc.) that drive the SPARE model. The sample data used in the test drive are from the MESSOC data base (Version 2.1). That data base contains data for most of the distributed systems. If you wish the model to optimize only for your particular distributed system, rather than for the entire space station, your data base should contain the ORUs of your system only.

INPUT FILE DESCRIPTION

The file is in a text format (American Standard Code for Information Interchange – ASCII) so you can browse or edit it with standard PC editors. [*Note:* Take care to save the file in an ASCII or print format.] Examples of the data file headings and a few ORUs from the MESSOC data base are presented in Table 3. The eight fields of the file are defined as follows:

- *ORU NAME:* a 25-character ORU name.
- *INDEX Q:* any integer identification number you like. MESSOC uses a different Q value for every distinct ORU, but the model uses this value only for reporting purposes.
- *DAILY DEMAND:* expected mean demands per day. This value is the driver of the SPARE model. It is derived by taking the inverse of the mean time between failures in days. That demand value is the sum of all common ORU demands across the station. Thus, each distinct ORU has only one row in the data base.
- *COL A PRICE:* the unit price of the ORU. It can also contain any other limiting resource such as volume.
- *COL B WEIGHT:* the unit weight in pounds of the ORU. It can also contain any limiting resource. The user input section of the model run will ask you whether you want to use COL A, COL B, or some linear combination of the two to act as the limiting resource.

- *MIN. SPARE:* a value that allows the user some flexibility in determining how the model selects spares. Values greater than zero may force the model to buy at least the number of spares specified. (See the options discussion for information on alternative ways of selecting spares.)

- *SYS.:* the distributed system number. That value allows the model to further break down station outputs to a distributed system level. If the data base contains ORUs for a specific distributed system only, this field can be used to report results at a subsystem level.

- *MUL:* a demand multiplier. Certain ORUs have more demands than actual failures. Environmental problems, false removals, and scheduled maintenance are examples of additional demands that require spares. The ORU MUL value is multiplied by the DAILY DEMAND to obtain the total demands used in the model. The model requires that the MUL value be greater than zero.

TABLE 3

MODEL INPUT FILE

(SPAREIN.RPT)

ORU NAME	INDEX Q	DAILY DEMAND	COL A PRICE	COL B WEIGHT	MIN. SPARE	SYS.	MUL
PUMP	1018	002192	16 40	3 4	0	4	1
ACCUMULATOR	1019	000548	18.90	3.9	0	4	1
PLT SENSORS	1023	004384	50	.1	0	4	1
THRUSTER ASSEMBLY	1025	008219	46.50	16.6	0	2	1

You can also create your own file from your ORU data base with spreadsheets or data base software packages. If you do so, you should make sure the package output file is in an ASCII format, sometimes referred to as a "report," a "text," or a "print" file. The SPARE model directly reads the SPAREIN.RPT file and thus requires the eight data fields discussed in Table 3 in relatively loose data format. The order of the ORUs is not important; only the relative position of the data fields is significant. Adherence to the following five format rules is necessary when creating your own ORU input file:

- The ORU data starts on the fifth file line (i.e., after four lines of titles). In Table 2, that position is where the first ORU named "PUMP" is located.

- The end of the file is at the last ORU in the file; the file contains no extraneous lines below the input data.
- The only field that is required in particular columns of the file is the ORU NAME field. The model assumes the first 25 columns of the file is the ORU name.
- No other fields have to be in specific columns, but the fields must be separated by one or more blanks.
- All eight fields must be included even though the model, when optimizing, only uses the demand and resource (COL A and COL B) fields in its algorithms. If you do not wish to produce the other information, just put "0"s in the MIN. SPARE field and "1"s in the others.

EDITING INPUT FILES WITH LOTUS 1-2-3

SPAREIN.RPT can be edited with LOTUS 1-2-3, a standard software package used throughout NASA. For that operation, we have created a special file (SPAREIN.WK1) that contains the model ORU data in the LOTUS format. That file can then be edited with LOTUS and the results directly transferred to the SPAREIN.RPT file.

The string of LOTUS menu commands (in quotes) required to edit and transfer the file are the following:

- Enter the LOTUS worksheet software
- Retrieve the ORU data in the following LOTUS format:

  "/ FILE RETRIEVE C:\SPARE\SPAREIN"
- Make your edits to the LOTUS worksheet
- Print the file directly to the following model input file:

  "/ PRINT FILE C:\SPARE\SPAREIN.RPT REPLACE GO".

If you add ORUs to the file, increase the "RANGE" before selecting the "GO" command. The LOTUS file has already been adapted so that the format of the print file is identical to the SPAREIN.RPT file.

INPUT FILE SUMMARY

In conclusion, you have several options for obtaining an input file. You may use the MESSOC file we supplied and edit the ORU data to more closely reflect your specific work package information. Second, you may create your own input file with another software package, following the format rules presented above. Third, you may combine the two alternatives. For instance, you may want to produce your own ORU data for a specific distributed system, delete the MESSOC ORU data of that distributed system from the sample MESSOC file, and then add your data to the end of the sample file.

CHAPTER 6

MODEL OPTIONS

The model has several optional ways of producing results that are all triggered from the OPTIONS.RPT file (see Table 4). The user can change any of the option values in the file's left-hand column with a PC editor or LOTUS 1-2-3. The replaced values do not have to be in exactly the same columns as the original but merely in the general vicinity. If you use an editor, make sure the file remains a text file (ASCII) and not the specialized format of the software.

If you wish to use LOTUS 1-2-3 to edit the options file, follow the steps discussed in the section in Chapter 5 entitled "Editing Input Files with LOTUS 1-2-3" but change the file names. The LOTUS 1-2-3 file retrieval command is changed to "/ FILE RETRIEVE C:\SPARE\OPTIONS" and the print command is changed to "/ PRINT FILE C:\SPARE\OPTIONS.RPT REPLACE GO".

The definition and range of global values for each of the options are as follows:

- *LOGISTICS CYCLE:* the time in days between shuttle flights.
- *MIN. SPARES:* this option has three alternatives. For the first two, the model determines the spares mixes from different starting points, and for the third alternative, the model evaluates the spares mix the user specifies. If the global value is 0, the process proceeds as discussed earlier in the "How It Works" section of this document. That is, the model sets all ORU spares levels to zero and then selects spares until the user target is reached. If the global value is 1, the model sets all ORU spares levels at the specific ORU MIN. SPARE value from the input file (SPAREIN.RPT), as opposed to zero, and then selects spares until the user target is reached. If the global value is 2, the model sets all ORU spares levels to the MIN. SPARE value from the input file, but now the availability and resource expenditure of the MIN. SPARE levels replace the user targets. Thus, the MIN. SPARE levels become the initial spares solution and the model evaluates the station availability and the given resource expenditures.
- *GLOBAL MULTIPLIER:* the value that is multiplied by the ORU demand to obtain a new total demand. This value is similar in function to the ORU multiplier in the input file but is used to uniformly change all ORU demands.

- *MAXIMUM:* the maximum availability for the entire curve (expressed as a percentage). Depending on the degree of accuracy, you can set the maximum to 99 percent or 99.999 percent.

- *PLOT:* a value that allows the user to turn the plot of the resource-versus-availability curve on (global value = 1) or off (global value = 0).

TABLE 4

MODEL OPTIONS FILE

(OPTIONS.RPT)

Global value	Description
90	LOGISTICS CYCLE in days: time between shuttle flights
0	MIN. SPARES: 0 – minimum spares level set to 0. 1 – minimum spares level is starting point for optimization 2 – minimum spares level is evaluated as user target on curve
1.0	GLOBAL MULTIPLIER for demand: daily demand x multiplier
99.0	MAXIMUM percent availability for curve
1	PLOT: If "1" will plot the resource-versus-availability curve If "0" will not

CHAPTER 7

MODEL OPERATION

Once you have edited or created the ORU data base (SPAREIN.RPT) and possibly modified the model options (OPTIONS.RPT), you are ready to run the SPARE model (type "RUNSPARE" from the C:\SPARE directory). As demonstrated in the "Quick Test Drive" section, the model requires a few simple inputs from you before it is operated. Figure 3 displays the user input choices that appear on the PC monitor during the interactive session of the model run.

```
 ENTER THE RESOURCE TYPE:
    0  FOR PRICE  AS THE RESOURCE (COLUMN A IN DATA FILE)
    1  FOR WEIGHT AS THE RESOURCE (COLUMN B IN DATA FILE)
    2  FOR LINEAR COMBINATION OF PRICE AND WEIGHT:
       (PRICE * COEFFICIENT)  +   (WEIGHT * (1-COEFFICIENT))
    ENTER PRICE COEFFICIENT FROM 0 TO 1
     THEN WEIGHT COEFFICIENT =

ENTER THE TARGET TYPE:
(The target specifies the point on the Resource Vs Availability
Curve that in turn determines the initial spares list.
   0  FOR A RESOURCE TARGET
   1  FOR AN AVAILABILITY TARGET
ENTER THE RESOURCE TARGET
     or
ENTER THE AVAILABILITY TARGET PERCENTAGE FROM 0 TO 100
```

FIG. 3. USER INPUT CHOICES

The first model choice in Figure 3 is type of limiting resource. If you enter "0" or "1", the model uses the price or weight as the resource, respectively, from the SPAREIN.RPT input file. As discussed, you might also want to use other resources such as unit volume. In that case, when developing the input file, insert volume into one of the resource columns (A or B) in the file. Then enter "0" for Column A (the old price column) or "1" for Column B (the old resource column) at this point in the model run.

If you enter "2" for the first input, the model uses a linear combination of both resources. The model then asks you to supply the price coefficient from 0 to 1. That coefficient determines the relative importance of the two resources. A price coefficient of 1 (i.e., a weight coefficient of 0) means the model will produce solutions that give the lowest total dollars for spares but ignore the weight of the ORU. As the price coefficient becomes smaller (values less than 1), more importance is given to reducing the total weight of the ORU by sacrificing the low total dollar price.

For the next input, the model requires you to select the target type. If you enter a "0", the model assumes the target (next input) will be a resource target. If you enter a "1", the model assumes the target is an availability target.

Finally, the model requires the actual target value. Availability target values must be greater than 0 and less than 100. A meaningful resource target depends upon the resource used. The target specifies a point on the curve and that point, in turn, specifies your initial spares. In our test drive example, you used a resource target of $82,000. That means you wanted the optimal spares mix given that you have approximately $82,000 to spend. Alternatively, you might choose to determine the optimal spares mix for a availability target of, say, 90 percent. Whatever the case, the edge of the shaded portion under the curve shows the approximate location of your target on the curve. In the following chapter, we discuss where to find the initial spares list generated by your target point. To print a hard copy of the plot, you press the *print screen* key. [*Note:* If nothing prints, add the command "GRAPHICS" to your "AUTOEXEC.BAT" file.] If you enter a return, the model session will end.

CHAPTER 8

MODEL OUTPUTS

The model output file (C:\SPARE\OUT.RPT) contains all the input and output data we have just discussed: (1) the user inputs entered from the screen, (2) the model options selected from the OPTIONS.RPT file, (3) the ORU input information from the SPAREIN.RPT file, (4) the initial spares solution for the user target value, (5) the availability and resource output expenditures disaggregated to a distributed system level, and (6) the resource-versus-availability curve in tabular form. We briefly discuss each of those tables in the following sections. To better understand that discussion, you may want to browse the OUT.RPT file using your editor as we go. If you use LOTUS 1-2-3, you can import the OUT.RPT file into a blank worksheet to browse by using the following LOTUS 1-2-3 commands:

"/ FILE IMPORT TEXT C:\SPARE\OUT.RPT"

Each of the output tables is discussed in a separate section of our document and is identified by the table title.

USER INPUTS AND OPTIONS TABLE

This first table contains the user inputs from the interactive session and the model options from the OPTIONS.RPT file. The user inputs reflect the inputs from the current run: the resource (price, weight, or combinations), the coefficient for price and weight, and the user resource or availability target. The last line in the table contains the global values for all the model options.

ORU INPUT DATA TABLE

The next table is the ORU input data read from the SPAREIN.RPT file. The column headings of the table are similar to the input file headings defined previously. If you make changes to either of the input files, you should verify that the model incorporates them correctly.

SPARES MIX FOR SOLUTION POINT TABLE

The key output of the model is the optimal mix of spares that maximizes availability for a user-specified point. The point, availability and resource value, on the curve is given first. For our test drive example, the point's availability is 89.22 percent and the resource is 82,023. Next, the table lists all the ORUs (index, name, etc.) and the corresponding number of spares chosen (column labeled "SPARES") for the optimal solution. If you multiply the number of spares and the unit resource for each ORU and then sum up all ORUs, you will produce the value of 82,023. If you multiply the individual probability of sufficiency of the ORUs (column labeled "POS") together, you will produce the 89.22 percent availability.

DISTRIBUTED SYSTEMS RESULTS TABLE

The distributed systems results table is a more-detailed breakdown of the station solution point just discussed but is derived in a similar manner. Table 5 is an example from our test drive of those results. The bottom line of that table gives the station solution point on the curve (89.22 vs. 82,023) as well as the total weight of the initial spares of 11,429 pounds. The table disaggregates the space station results to distributed systems by number (1 to 15). The distributed systems are ORU subsets of the total space station and are defined by the SYS. field of the input data. The model calculates the system results for each ORU subset in a similar manner as the station results. For instance, a distributed system availability is the product of its ORUs POS. Currently, each ORU can only appear in a single distributed system, i.e., an item cannot have more than one application.

RESOURCE-VERSUS-AVAILABILITY CURVE TABLE

The resource-versus-availability curve table shows all the points that make up the curve produced by the model. Point by point, the table lists the station availability, accumulated resource, and the ORU selected. The format of this table is similar to that of Table 2. The table starts with all spare levels at zero (or no resources expended) and ends when the station availability is 99 percent (the maximum availability set in the OPTIONS.RPT file). In our test drive, the model selected 839 spares and the total resources expenditure was $123,740.

TABLE 5

DISTRIBUTED SYSTEMS RESULTS

System	Availability	Weight	Price	# ORUs
1	99.4799	2,723	3,327.10	28
2	99.9748	111	312.00	3
3	99.9972	145	105.90	10
4	99.7066	550	2,481.70	5
5	99.7859	403	1,855.50	17
6	97.9307	283	13,159.90	6
7	99.9712	346	502.60	15
9	98.8951	573	9,508.90	7
10	99.9713	50	176.00	2
11	95.1934	1,936	38,426.30	3
13	99.6088	3,496	2,541.00	16
14	99.9789	355	568.80	4
15	98.2642	455	9,057.60	38
Station	89.2195	11,429	82,023.30	154

CHAPTER 9

MODEL ENHANCEMENTS AND USES

As discussed, the SPARE model is a prototype that establishes a framework for determining the initial spares mix. However, there are many additional enhancements that will make the SPARE model more comprehensive and beneficial. Below, we list some possible enhancements to the SPARE model and give a description and advantage of each. SPARE's basic architecture has been designed to make such enhancements possible without necessitating complete restructuring of the model. At the end of this chapter, we discuss some general uses for the prototype model and an enhanced model.

MULTI-ECHELON STOCKAGE

With the multi-echelon stockage enhancement, the model will estimate the optimal levels of ground and on-orbit spares. The addition of ground spares estimates to the prototype model will give a complete *initial spares picture*. The model enhancements will consider the number of logistics cycles required to repair or replace ORUs on the ground and the tradeoffs between higher costs of on-orbit sparing and the less expensive and less responsive ground spares.

NONCRITICAL ORUs

Once multi-echelon enhancement is incorporated, the model will also estimate optimal spares levels for noncritical ORUs (MCC > 1). The model will only determine the optimal ground spares levels, excluding on-orbit spares, as a possibility.

DUAL CONSTRAINTS

With the dual constraint enhancement, the model will estimate an optimal spares mix under two constraints as opposed to the current single constraint optimization. For instance, the model will estimate an optimal spares mix for a budget constraint *and* an availability constraint.

ORU DEMAND UNCERTAINTY

The ORU demand uncertainty, defined as the ORU demand process variance-to-mean ratio (VMR), directly impacts the spares optimization. Larger VMRs translate into higher spares requirements. Currently, the model uses a VMR of 1 reflecting an assumption of a Poisson demand process for all ORUs. To better reflect ORU differences, the enhanced model might use a VMR less than 1 for ORUs that exhibit "wearout" characteristics. Conversely, the model might use a VMR greater than 1 for ORUs where the data quality is suspect.

REDUNDANCY

An enhanced model will estimate the benefit to station availability of an ORU based upon the number and levels of redundancy. For instance, a distributed system may contain identical subsystems with redundant functions. Further, those subsystems may contain ORUs that also have on-line redundancy. Subsystems with a higher degree of redundancy should require less spares than a subsystem with a lower degree of redundancy, all else being equal. In addition, the enhanced model will estimate station availability for different baselines. For example, if the station has four identical subsystems, the model will estimate station availability assuming that at least one out of four subsystems are operating, two out of four subsystems operating, etc.

REPAIR ISSUES

With additional enhancements, the model will assess the impacts to the spares mix when ORUs are repaired in orbit with lower indenture items that are cheaper, lighter, and smaller. The model will also optimize the on-orbit and the ground spares of those lower indenture items. Also, the model will compare the on-orbit spares levels with and without cannibalization in space.

Because current literature does not address the unique aspects of the space station inventory, LMI recently developed the methodology to incorporate the above enhancements into the SPARE model optimizing framework. However, detailed implementation of computer algorithms, data requirements, and other specifics cannot be accomplished in our task's timeframe although we feel they are important to eventually include in the model.

OTHER MODEL USES

The SPARE model prototype can estimate initial spares and analyze tradeoffs between availability and resources for a number of logistics issues. For instance, for critical pressurized ORUs, the model can estimate the optimal spares list for a fixed on-orbit storage volume target. For unpressurized critical ORUs, the model can estimate spares based upon shuttle upweight constraints. Also the prototype can analyze the impact on the space station if the resupply time is increased or decreased, if the minimum spare level is changed from 1 to 0, or if the on-orbit storage capacity of the station is increased. Finally, through multicriteria optimization, the prototype can balance several resources and quantify the tradeoffs of sacrificing one resource for an improvement in another.

With the enhancements we suggest, the SPARE model could perform additional analyses. For noncritical ORUs, the model could estimate spares for a specific budget or determine which ORUs are most beneficial to be spared if excess on-orbit storage becomes available. Also, the model could analyze how ground stockage is affected by the length of the repair cycle or examining the tradeoffs between ORU replacement versus shop replaceable unit repair, Kennedy Space Center repair versus vendor repair, and cannibalization versus no cannibalization. Finally, model results could defend the spares budget or estimate the benefits of common components.

The prototype, although limited in scope, is a useful tool and demonstrates the importance of the SPARE model methodology. An enhanced SPARE model will significantly increase the scope, the usefulness, and the accuracy of the prototype and can be an important decision tool at NASA Levels II, III, and IV. We realize that as work proceeds on the spares inventory additional issues will arise. We feel that because the SPARE model methodology is flexible enough to incorporate the current enhancements, it also will be flexible enough to address future enhancements.