BRL-TR-2914

DTIC FILE COPY

(4)

AD-A196 266

TECHNICAL REPORT BRL-TR-2914

BRL

1938 – Serving the Army for Fifty Years – 1988

INPUT MANUAL FOR THE ARMY RESILIENCY ANALYSIS (AURA) METHODOLOGY: 1988 UPDATE

J. TERRENCE KLOPCIC

MAY 1988

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

U.S. ARMY LABORATORY COMMAND

BALLISTIC RESEARCH LABORATORY

ABERDEEN PROVING GROUND, MARYLAND

88

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A196 266

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): TECHNICAL REPORT BRL-TR-2914	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: U.S. Army Ballistic Research Laboratory — 6b. OFFICE SYMBOL (If applicable): SLCBR-VL-I	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (City, State, and ZIP Code): Aberdeen Proving Ground, MD 21005-5066	7b. ADDRESS (City, State, and ZIP Code)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION — 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
RDT&E	1L162618	AH80	

11. TITLE (Include Security Classification)

INPUT MANUAL FOR THE ARMY UNIT RESILIENCY ANALYSIS (AURA) METHODOLOGY: 1988 UPDATE

12. PERSONAL AUTHOR(S)

J. Terrence Klopcic

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
FINAL	FROM 1984 TO Mar 88		

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

AURA, Resiliency, Combat models; Military unit Reconstitution; Optimization; Chemical models; Nuclear models

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

The purpose of this report is quite limited, viz. to update the compendium of input commands and formats for the Army Unit Resiliency Analysis (AURA) main program. Only a brief description is given with each of the 60 commands listed. Commands are grouped by function: Execution, Names, Asset descriptors, Deployment Weapons and Weapon effects, Individual tasks, Unit functions and Program controls. In addition, sections are included on general information and appendices containing lethality file information and on AURA overview. (edc)

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: J. Terrence Klopcic	22b. TELEPHONE (Include Area Code): 301-278-6322 — 22c. OFFICE SYMBOL: SLCBR-VL-I

DD Form 1473, JUN 86 — Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

Since its inception in 1978, the Army Unit Resiliency Analysis (AURA) methodology has been applied to an ever broader spectrum of unit-level survivability/sustainability problems by an increasing number of analysts in the United States and abroad. Meanwhile, the methodology has continued to grow to respond to special needs that have arisen in the many applications. This situation resulted in a strong need for the publication of an AURA user's manual to update and standardize runstreams between users. The first such a manual was published in 1985.[1]

Since the publication of the first manual, both AURA and the AURA users community have continued to grow. Several options have been added, including a major extension to the functional structure to include the CREW option. Thus, the need for an updated manual has become apparent. This report is designed to fill that need.

As with the first user's manual, the purpose of this report is quite limited, viz., to provide a concise compendium of input commands and formats for the Army Unit Resiliency Analysis (AURA) computer code. It is assumed that the reader is reasonably familiar with the AURA family of methodologies in general, and with the purpose and function of the many options of the AURA code itself. Therefore, only a brief description of command functions will be given with each entry. For more information, the user is referred to Appendix B or the Introduction to the Use of AURA report.[2]

The command descriptions in this report are grouped by function: those that describe assets are presented together, followed by those that describe threat weapons, etc. The groupings and corresponding sections are listed in Table 1. Within each grouping, the entries are arranged alphabetically.

---

1. J. Terrence Klopcic, *Input Manual for the Army Unit Resiliency Analysis (AURA) Methodology*, BRL-TR-2670, (SEP 85). AD A159327.

2. Reference: J.T. Klopcic and L.K. Roach, *An Introduction to the Use of the Army Unit Resiliency Analysis (AURA) Methodology: Volume I,* BRL-MR-3384. (SEP 84).

TABLE 1. GROUPINGS OF COMMANDS FOR THIS REPORT.

CHAPTER	SECTION	GROUPING
II	A	EXECUTION
II	B	GENERAL RUNSTREAM INFORMATION
III	A	NAMES
III	B	ASSET INPUTS
III	C	DEPLOYMENT INPUTS
III	D	WEAPON INPUTS
III	E	WEAPON EFFECTS INPUTS
III	F	UNIT TASK INPUTS
III	G	UNIT FUNCTION INPUTS
III	H	PROGRAM CONTROLS

II. EXECUTION and GENERAL INFORMATION

A. EXECUTION

It is generally convenient to assemble the commands and data for an AURA run in a file (hereafter called the RUNSTREAM). It is therefore necessary to redirect the computer input from its normal input channel to the RUNSTREAM. For this purpose, an AURA execution input stream requires one card containing the name of the runstream file. (In this report, the term "card" will refer to a single input line.)

Optionally, two more cards may be entered. If a second card is read from the normal input channel, it is interpreted as a heading to be printed at the beginning of the AURA output and again at the beginning of the results summary. If a third card is read, it is interpreted as a deployment offset. (Both headings and offsets, which can also be input via the RUNSTREAM, are described in the following sections. See HEADING and OFFSET in sections IV.G and IV.B respectively.)

Format:

```
name of runstream file
heading information ( 80 character maximum )
x-offset, y-offset (REAL)
```

B. GENERAL INFORMATION

1. **Formats.** All AURA data sets are preceded by a mnemonic control card and followed by an END card. For example, the mnemonic control to input deployment data is "DEPLOYMENT". Depending on the particular mnemonic, several cards of data may then follow: the most recent mnemonic "remains in effect" until ENDed or superseded. Thus, if several pieces of data are appropriate for a particular mnemonic (e.g. several items are to be deployed), the DEPLOYMENT mnemonic need only appear once, followed by several cards of deployment data.

Each data set should end with an END card. However, if an END card is missing, the input routine will detect the following mnemonic control card and begin a new data set.

Mnemonic control cards may be abbreviated to the first six letters.

2. **Order.** With few exceptions, the various data sets may be input in any order. Furthermore, although not advised, data of a particular type can be input in several sets (each preceded by the particular mnemonic). The major exceptions to the any-order rule are:

1. The first data set must be the NAMES data set

2. Functional forms (LINKs, CREWs, SUBCHAINs, ORLINKs, COMPOUND LINKs) must be input before any higher level functional forms (CREWs, SUBCHAINs, ORLINKs, COMPOUND LINKs, CHAINs) which use them.

3. **Data Forms.** All AURA data cards are in one of three forms:

TYPE	DESCRIPTION
HR	All Hollerith (text) words or names, with each set of words or names separated by a comma. (NOTES: Embedded commas are not allowed. Leading blanks are ignored.)
R0	All numbers separated by blanks or commas.
R1	One set of words or name followed by numbers. The words/name must be separated from the numbers by a comma; the numbers may be separated from each other by blanks or commas.

4. **Number Types.** Numbers in AURA are either INTEGERS or REALS. (REALS are floating point numbers or exponential numbers with decimal points). AURA checks to assure that the anticipated combination of INTEGERS and REALS is input and will terminate a run on a FATAL FORMAT ERROR if an errant combination is detected. The following sections specify the types of numbers anticipated.

5. **Names, General.** Names for assets, weapons and functional structure parts (LINKs, CREWs, SUBCHAINs, ORLINKs, COMPOUND LINKs) may not contain more than 18 characters, including imbedded blanks. Use of special characters within names (see following section) may cause fatal errors or unanticipated behavior and should be avoided.

6. **Special Characters.**

(a) **Continuation Cards.** Data may be continued on a subsequent card. A continuation card is denoted by a dollar sign ($) as the first character on the card, followed by data in format HR, R0, or R1 as appropriate. A comma may, but need not, follow the dollar sign.

Continued data falls into two types: supplementary data (data that is different from that of the preceding card) and overflow data (of the same type which did not fit on the preceding card). Several data sets allow options which must be entered on supplementary continuation cards. For example, to specify an alternate posture for a deployed asset, the deployment data card is followed by a supplementary continuation card containing the alternate posture code and mean-time-to-change.

(b) **Comments.** The pound sign (#) immediately stops the scan of a card by the AURA read routines. Thus the user can insert comment cards (for his own use, not to be read as data) into an AURA runstream by beginning the card with a pound sign. Similarly, any data card can have a comment appended, as shown below.

```
#THIS IS AN EXAMPLE OF COMMENTS IN A RUNSTREAM
DEPLOYMENT        # BASED ON FM - XYZ
MECHANIC, 100., 352., 1.0, 1,1,1,1,1,0  # MOTOR POOL
```

(c) **Functional Structure Names.** As described in the following sections, names for the CREW, SUBCHAIN, ORLINK and CPLINK functional structure begin with =, *, + and ! respectively.

(d) **Embedded Procedure Commands.** Anticipated extensions of the AURA input routines include the development of embedded procedures, which will be signified by a backslash (\) in column 1. An example of such a procedure is the embedded offset (under DEPLOYMENT) which translates blocks of data.

7. **Terms Used in the Following Sections.** **ASSETS:** Personnel and items of equipment. The term "asset" is generally used when referring to the physical attributes of an item in a unit.

WEAPONS: Incoming weapons. The term "weapon" includes warheads and delivery systems.

MISSION: Performance rate or capability upon which the study unit is being evaluated. For example, a supply unit might be evaluated on its ability to issue 2000 tons of supplies per day.

FUNCTIONAL STRUCTURE: The organization of activities and tasks that result in the accomplishment of a unit's mission(s). Note that the functional structure is made up of JOBS that are done, not the ASSETS that do them. (The connection between job positions and assets that can fill the positions is made through the LINK input.)

It is convenient to consider the functional structure for a mission as being constructed through three levels of aggregation. The lowest delineation of subtasks or job positions is called a LINK. For example, one might define a radio operator's job as a LINK if one were not concerned with subtasks within the job.

Jobs combine together in different ways to perform the various activities that go on within a unit. For example, an artillery unit might have several activities going on at one time during a fire mission: firing, loading, fire direction calculating and new position reconnaissance. The AURA name for such an activity is called a SEGMENT. SEGMENTS are made up of LINKs; however, the various LINKs within a SEGMENT may be combined to show different relationships. Thus, a SEGMENT may be a single job (LINK), a group of jobs that must be done together (CREW and/or SUBCHAIN), a choice between groups of jobs (ORLINK) or several groups of jobs such that the total activity is a summation of the independent contributions of the different groups (COMPOUND LINK).

The essential feature of SEGMENTS is that they all contribute to the mission in such a way that mission accomplishment is limited by the weakest SEGMENT. The collection of SEGMENTS, which span the activities of the unit toward mission accomplishment is called a CHAIN, the final level of aggregation in the AURA functional structure model.

A more complete description of the AURA functional structure is given in Appendix B.

HOMELINK: A LINK which has the same name as an asset is referred to as that asset's "HOMELINK." An asset is immediately available for its HOMELINK task and contributes to its accomplishment at 100 percent effectiveness. An asset has priority for assignment to his own HOMELINK. However, this priority may be violated if the asset is needed elsewhere, or is too ill to perform at an acceptable level. (See Allocation Algorithm Decision Rules, next section.) Any asset which does not have a LINK of the same name is assigned the HOMELINK NOLNK.

DUMMYLINK: A LINK which has no asset of the same name is called a "DUMMYLINK." DUMMYLINKs are jobs which do not have a particular asset assigned to them, but are filled -- when needed -- by substitutes. DUMMYLINKs are defined with the same commands as HOMELINKs. (See section IV.E.2.)

8. **Allocation Algorithm Decision Rules.** The decision rules followed by the asset allocation algorithm (the "commander") in assigning assets to LINKs are as follows:

HOMELINK. A LINK is filled by its HOMELINK asset (an asset having the same name as the LINK) if one is available. If no HOMELINK asset is available, the commander will attempt to fill the LINK with a substitute. Also, if the available HOMELINK asset is degraded (e.g. because of sickness or fatigue) below a user-settable level (sicklv) and if there is a substitute available at a performance level more than (1/sicklv) greater than the best HOMELINK asset, then a substitute will be selected.

SUBSTITUTES. A potential substitute does not become available until the elapsed time (time since the need for a substitute developed) exceeds the substitute's (user-specified) substitution time. (See LINKS, section IV.E.2.) If *more than one* substitute is available in the elapsed time involved, a particular substitute is chosen by the following criteria:

1. **Effectiveness**
   To be considered as a substitute into a LINK, an asset must be relatively effective compared to the other potential substitutes. (Effectiveness is affected by cross-training, sub-lethal doses of radiation or chemicals, fatigue, etc.) AURA provides a user setable level (signif). Any potential substitute which is less effective than the best substitute by more than signif is automatically dropped from consideration.
2. **Versatility**
   The commander will assign a less versatile asset in preference to assigning a more versatile asset. (Versatility, an integer number, is defined as the number of LINKs to which an asset can be assigned. AURA internally predetermines the versatility of each asset by analysis of the substitution matrix.)
3. **User-Input-Order**
   The allocation algorithm numbers the substitutes for a particular LINK in the order in which they were named. (See LINKS, section IV.E.2.) The commander will assign a lower-numbered substitute in preference to a higher-numbered one. (Note, however, that several assets may

have equal order numbers, since several substitutes may be specified by the same common name. See NAMES, chapter III.)

The normal operation of the allocation algorithm is to take the decision criteria in the order presented above: a decision passes to the next criterion only if there is a "tie" in all preceding criteria.

The decision rules and values can be modified by the user. As stated above, the user can set the values of signif and sicklv. Furthermore, the order of consideration of criteria 2 and 3 (VERSATILITY and USER-INPUT-ORDER) can be reversed. See DECISION RULES, section II.B.8.

Finally, note that any decision made by the above rules can be over-ruled by a correction made through the look-back capability of the algorithm. For a more detailed discussion of the AURA Asset Allocation Algorithm, see Appendix B.

9. **Coordinate Systems.** Three natural coordinate systems are used in AURA to express data that refers to geographical locations or extents.

(a) **The UNIT Coordinate System.** Any right-handed coordinate system may be used to lay out the study unit. Any particular point in the unit, such as the geographical center or "lower left" corner, may be designated as the origin of the UNIT Coordinate system, and all other unit elements are deployed relative to that point.

(b) **The INCOMING FIRE (RANGE-DEFLECTION) System.** Threat weapon parameters are specified in RANGE and DEFLECTION, where RANGE is in the direction of the incoming fire, and DEFLECTION is normal to RANGE such that RANGE-DEFLECTION define a right-handed, horizontal coordinate system. The user can specify the orientation of the RANGE direction relative to the UNIT Coordinate system. (See INCOMING FIRE, section IV.C.6.)

(c) **The WIND DIRECTION (DOWNWIND-CROSSWIND) System.** Toxic cloud dispersion is specified in the DOWNWIND and CROSSWIND directions, which are defined to form a right-handed, horizontal coordinate system. (The NUSSE3 standard toxic dispersion code uses this coordinate system: Thus, no adjustments are necessary to use NUSSE3 data in AURA.) The user can specify the orientation of the DOWNWIND direction relative to the UNIT Coordinate system. (See WIND DIRECTION, section IV.C.12.)

10. **Units, Times and Time Intervals.** Much of the data which is input into AURA has associated units, most commonly length or time. It is essential, therefore, to establish a consistent system of units, since parameter values can be markedly different in different systems. (For example, an event time could be input as

3600., 60., or 1. depending on whether seconds, minutes or hours are being used.)

With the exception of certain nuclear algorithms which contain built-in basic data, AURA can be used with any consistent set of units. However, achieving such consistency may require a great deal of thoroughness. For example, both length and time are buried in the threshold value for chemical alarm functioning. We therefore recommend the following set of units, consistent with the nuclear algorithms:

Time	minutes
Distance	meters
Toxic deposition	mgm/m**2
Toxic dose	mgm-min/m**3

The following units are essential:

Angles	degrees
Nuclear dose	rads (cGy)
Nuclear yield	kt

Some of the AURA options require the input of a time duration. For example, to allow individuals under attack to change posture requires specifying the mean time for the change. On the other hand, those options which result in the scheduling of an event, such as the options which specify that an attack will occur, require input of an absolute (clock) time into the simulation. In this report, all time inputs will be identified as (intrvl) or (clock) to indicate whether the value is a duration interval or an absolute event time.

11. **Alphabetical Listing.** The following table is an alphabetical listing of all current mnemonics and the corresponding section of this report which describes them.

SECTION	MNEMONIC
C.1	ACQUISITION PROBABILITY
C.2	AGENT
A.1	ALARM
C.3	CEP ERROR
C.4	CEP TLE
F.1	CHAINS
F.2	CREW
F.3	COMPOUND LINK
D.1	CONVENTIONAL LETHALITY
A.2	CONTAMINATED USAGE
G.1	DECISION RULES
B.1	DEGRADATION
C.5	DELIVERY ERROR
B.2	DEPLOYMENT
D.2	DOSE PARAMETERS
A.3	EXPENDABLE
A.4	FAILURE RATE
E.1	FATIGUE
G.2	GO
A.5	GRANULARITY
G.3	HEADING
C.6	INCOMING FIRE DIRECTION
G.4	INTERNAL RECONSTITUTION TIMES
E.2	LINKS
A.6	LOSSES
C.7	MISS DISTANCE
G.5	MODE
B.3	MOPP
III.	NAMES
D.3	NUCLEAR VULNERABILITY
B.4	OFFSET
F.4	ORLINKS
G.6	OUTPUT OPTIONS
D.4	PERSISTENCE
G.7	RECONSTITUTION EVENTS
A.7	REINFORCEMENTS
C.8	RELIABILITY OF WEAPONS
A.8	REPAIR
G.8	REPLICATIONS
B.5	REST

C.9	ROUND
A.9	SECONDARY EXPLOSIVE
G.9	SEEDS (random number)
B.6	SHIELDING
G.10	STOP
F.5	SUBCHAINS
E.3	SUBLETHAL DOSE DEGRADATION
D.5	THERMAL
A.10	TIREDNESS
B.7	T.K.C. (toxic kill code)
C.10	TLE
D.6	TOXIC DISPERSION DATA
G.11	TRACE LINK USAGE
C.11	VOLLEY
C.12	WIND DIRECTION
C.13	YIELD

12. **Events.** The following table contains the mnemonics from the above table which can be used to insert various types of events into the EVENT TABLE for an AURA SIMULATION.

SECTION	MNEMONIC
C.1	ACQUISITION PROBABILITY
C.3	CEP ERROR
C.4	CEP TLE
C.5	DELIVERY ERROR
C.6	INCOMING FIRE
G.4	INTERNAL RECONS. TIMES
A.6	LOSSES
G.7	RECONSTITUTION EVENTS
A.7	REINFORCEMENTS
C.9	ROUND
C.10	TLE
C.11	VOLLEY
C.12	WIND DIRECTION

III. NAMES

The first data set in an AURA input stream is a list of names to be used for assets and weapons. This block may be headed by a NAMES or REPERTOIRE card and must be terminated (after all asset and weapon names have been listed) by one END card. A group of asset names must be headed by a ASSETS card; weapon names must be headed by a WEAPONS card. NOTE: AN END CARD MAY NOT BE PLACED BETWEEN THE ASSET AND WEAPON NAME LISTS.

Format:

```
(HR)
ASSETS
asset name1, alternate name1a, alternate name1b, ....
asset name2, alternate name2a, ....
....
WEAPONS
weapon name1, alternate name1i, alternate name1j, ....
weapon name2, alternate name2i, ....
....
END
```

where all names conform to restrictions listed in section II.B.5.

Each asset and weapon must have at least one unique name. Alternate names are used to associate common parameters to groups of assets or weapons. To invoke certain code defaults, the following alternate names SHOULD be used:

- PERSONNEL - should be attached to each personnel asset
- CONVENTIONAL - should be attached to each conventional or mixed conventional/toxic weapon
- NUCLEAR - should be attached to each nuclear weapon
- TOXIC - should be attached to each toxic or mixed conventional/toxic weapon

IV. AURA INPUT OPTIONS and COMMANDS

A. ASSET INPUTS

The following data sets input parameters that describe the assets (personnel and equipment) of a unit.

1. **ALARM.** This option identifies chemical alarms and indicates threshold for activation due to each (toxic) weapon. (Currently, the threshold is in terms of dosage.)

Format:
```
(R1)
ALARM
asset name of alarm (as defined in NAMES input)
weapon name1, threshold dosage for alarm to sound (REAL)
weapon name2, threshold dosage ....
....
```

NOTES: Toxic dissemination data (UNIT #4) must be read in first to allow the code to adjust for dose normalization. (See TOXIC DISPERSION DATA, section D.6.)
Alarms are deployed by asset name like any other equipment. (See DEPLOYMENT, section B.2.)
Alarms will have no effect (will be too late) if personnel begin to MOPP-up as soon as round arrives. (See 'ROUND' option under MOPP, section B.3.)

2. **CONTAMINATED USAGE.** This option allows designating those items of equipment which may be used when contaminated, along with the missions (CHAINs) in which this usage is allowed.

Format:
```
(R1)
CONTAMINATED USAGE
asset name, CHAIN numbers (INTEGERS)
....
```

NOTE: If CHAIN numbers are omitted, all CHAINs are assumed.

Option: If asset name is "ALL", all assets become contaminated-usable

Option: If MUST MOPP ON, contaminated items usable only if ALL personnel in study are in a safe posture.

Format:
```
(R1)
MUST MOPP ON
MUST MOPP OFF        (Default = OFF)
```

3. **EXPENDABLE.** This option identifies assets that are expended as they are used. Two forms are available: Expenditure by time and expenditure by repair completion.

EXPENDABLE

Option 1, expenditure by time:
Format: (R1)
asset name, rate of usage by time (per minute) (REAL)

Option 2, expenditure by repair completion:
Format: (HR)
$asset name

NOTE: For assets identified as expendable under option 1, the amount that is assessed as expended at any time point depends upon the amount of mission time spent since the previous update and the effectiveness of the unit during that time. (Mission time is that time which follows a reconstitution and extends until interrupted by a lethality event.) Assets identified as expended during repair or decontamination activities must have HOMELINKs which appear in the consuming repair SUBCHAINs. (See REPAIR, section A.8.) LINK parameters should describe amount needed for one repair. (See LINKS, section E.2.)

4. **FAILURE RATE.** This option specifies the rate at which assets undergo spontaneous (reliability-type) failures. Equipment can fail so as to require light, medium, or infeasible repair; personnel can only have dead failures.

Format: (R1)
FAILURE RATE
asset name, mtbf, fl, fm

where:

mtbf is the mean time (intrvl) between any failures (REAL)
fl is the fraction of failures requiring light repair (REAL)
fm is the fraction requiring medium repair (REAL)

(NOTE: If fl and fm are not specified, all failures are taken as dead)

Option: Preload pool of ongoing repairs at time 0:
Format: (HR)
PREFAIL,ON

(NOTE: When PREFAIL is ON (the default case), light and medium repairs are prestarted, simulating an ongoing process at the initial time of the study. This option avoids having too many items available at time zero, too many failing afterwards, and a delay in the repair return rate.)

Option: If mtbf is entered as a negative number, the value of mtbf is taken as a probability of not being present at time 0. Requires PREFAIL option ON.

5. **GRANULARITY.** This option causes the iterative portion of the optimal allocation processor to consider allocating an asset in specified portions, thus decreasing the possibility of over-allocating to one task at the detriment of others. Since the optimization algorithm has built-in checks against such over-allocation, this option is not used except as a code development and diagnostic tool.

Format:
GRANULARITY
asset name, grnl

where:
grnl is the largest increment per allocation step (REAL)

6. **LOSSES.** This option causes prespecified assets to disappear at prespecified times in an encounter. This option has been used, e.g., in a detailed study in which an a priori decision to remove some assets at a point in the encounter was part of the scenario.

Format: (R1)
LOSSES
asset name, time, number
where:
time is the (clock) time of removal (REAL)
number is the number of assets removed (INTEGER)

7. **REINFORCEMENTS.** The opposite of LOSSES, this option causes prespecified assets to appear at prespecified times in an encounter. This option has been used, e.g., in a detailed study in which an a priori decision to reinforce some assets at a point in the encounter was part of the scenario.

Format: (R1)
REINFORCEMENTS
asset name, time, number
where:
time is the (clock) time of reinforcement (REAL)
number is the number of assets added (INTEGER)

8. **REPAIR.** This option inputs data relative to the repair of damaged or failed equipment. Included in the inputs are the subtasks (LINKs or SUBCHAINs) needed for repair, the mean time and standard deviation in the mean time for repair, and the penalty (0.- 1.) that the commander would be willing to take in his immediate mission in order to fix the item if the need for the item were the choke point in the mission. AURA also uses the penalty value to help prioritize possible repairs at various levels.

****

The REPAIR ALGORITHM

It is important to understand the ways in which a repair can be commissioned and role played by the penalty value. If the unit's future capability can be improved by repairing an item, the commander will consider adding the repair of the item to his current mission and optimize the allocation of assets to perform this augmented mission. A repair ordered this way is called a NEEDED repair. Note that, when a NEEDED repair is being done, the reported unit effectiveness may be determined by the ability to do the repair, not by the ability to do the actual unit mission. The code notes the difference in unit effectiveness between doing only its actual mission versus doing the repair in addition to its mission. If the loss in mission effectiveness is less than the penalty value specified, then the repair is done as part of the mission: its influence is included in the effectiveness of the unit. Hence, specifying a penalty value of 1.0 implies that a repair, if possible, is more important than anything, since all other accomplishment will be sacrificed, if necessary, to repair the damaged item. Note, moreover, that such a penalty is never required for an essential item, since mission accomplishment when the item is damaged is already low; thus the loss in effectiveness during repair cannot be great.

The other way of commissioning repair work is on an as-available (AS-AVL) basis. After the mission (or repair-augmented mission) requirements have been allocated, the commander assigns any remaining repair assets to do any repairs that can be done. These repairs are considered in numerical order by asset ID number; however, the top priority repair level is considered for every asset before considering lower levels.

Both of these ways of implementing repair activity are automatically considered by AURA if the REPAIR option is used.

****

As implied above, repairs can be specified on three levels: 0 = contaminated, 1 = light repair needed, and 2 = medium repair needed. It is assumed that light or medium repair can be applied to the specified item regardless of the source of need i.e. either failure or combat damage. See FAILURE for specification of equipment failures and Appendix A for specification of combat damage probabilities. NOTE: Repairable combat damage requires BOTH that the item appear as a repairable under this mnemonic AND that the item has exactly three kill criteria specified in the lethality file.

Format:
```
(R1)
REPAIR
asset name
$cp1,lvl1,pnlt1,mt1,sd1,xrp1,yrp1
$cp2,lvl2,pnlt2,...
...
```
where:

- cpI is the LINK or SUBCHAIN needed to perform level I repair on the named asset
- lvlI is the level of repair being described (INTEGER)
- pnltI is the acceptable mission penalty for level I repair (REAL)
- mtI is the mean time to accomplish level I repair (REAL)
- sdI is the standard deviation in mtI
- xrpI and yrpI are the coordinates of the location at which repair of this item at this level would take place

NOTE: If an asset name is not followed by repair card(s), a warning is printed. The code then assumes that the user wants combat damage levels checked and tallied, but knows that there is no repair available.

Option: GENERAL REPAIR. This option allows specification of general repair LINKs or SUBCHAINs, i.e. capabilities which must be satisfied ONCE in order for any repairs to be conducted.

Format:
```
(R1)
GENERAL REPAIR
$gnrl cp, lvl
```
where:

- gnrl cp is the LINK or SUBCHAIN needed for any repair at level lvl
- lvl is the level for which gnrl cp applies

Option: MAXIMUM NUMBER. This option allows specifying the maximum number of repairs which can be going on at any time.

Format: (R1)
MAXIMUM NUMBER, maxrp

where:

maxrp is the maximum number of repairs that can be ongoing at any one time (INTEGER). Default, maxrp = 50.

Option: NO REPAIR. This option allows specifying those CHAINs in which no repair or decontamination is allowed. NOTE: CHAIN input must precede REPAIR input if this option is used.

Format: (R1)
NO REPAIR,nch1,nch2,nch3,...

where:

nchI are the CHAINs which do not allow repair or decontamination.

9. **SECONDARY EXPLOSIVE.** This option allows identifying some assets as being potential sources of secondary explosions (i.e. being detonated by an incoming round and becoming additional lethality sources themselves). To use this option, the "explosive potential" is given a name (e.g. SECONDARY) which appears in the NAMES list as both an ASSET and a WEAPON. Similarly, the name appears in the conventional lethality file (see Appendix A) as both a target for other warheads, where data describes probability of detonation, and as a warhead, where data describes the effect of a detonation against all other targets in the unit. This option is used to associate the "explosive potential" with the appropriate (real) assets (e.g. ammunition stacks).

Format: (HR)
SECONDARY EXPLOSIVE
secondary explosive name, associated asset1, ...

10. **TIREDNESS.** This option allows input of the level (in SLUNITS) below which an asset is allowed to stop work and rest.

Format: (R1)
TIREDNESS
asset name, slunit level (Default: Any level)

B. DEPLOYMENT INPUTS

The following data sets input parameters related to particular (geographical) deployment points. Note that this also includes data that define codes for any status that is location dependent (like MOPP posture) and also data which are associated to that status (such as degradation due to MOPP posture).

1. **DEGRADATION.** This option allows the user to associate a MOPP code number and a TOXIC KILL CRITERIA (T.K.C.) code number with a performance degradation value. When the code is evaluating the effectiveness of individuals in doing tasks, it degrades the contributions according to the current MOPP posture and the T.K.C. of each deployment point. Thus, the degradation of a job due to the wearing of MOPP is input via the T.K.C. and toxic posture (and alternate toxic posture, if used). (See DEPLOYMENT, section B.2 and T.K.C., section B.7.).

```
Format:  (R0)
         DEGRADATION
         MOPP code, tkc, dgf
where:
```

MOPP code is defined under the MOPP option (section B.3) (INTEGER)

tkc is defined under the T.K.C. option (section B.7) (INTEGER)

dgf is the degradation factor for the specified MOPP posture and job difficulty (T.K.C.) (REAL)

2. **DEPLOYMENT.** This option indicates location, number, "kill criteria" and posture of assets. A supplementary continuation line may be used to indicate alternate postures and times-to-change-posture. This option is also used to locate places at which DUMMYLINK jobs would be done. (See section II.B.7.). Each set of data defines a TARGET POINT.

Format: (R1)
DEPLOYMENT
asset name, x, y, nmbr, ckc, nkc, tkc, cpst, npst, tpst

where:

- asset name is UNIQUE name of asset or DUMMYLINK at target point
- x, y are coordinates in UNIT Coordinate system (REAL)
- nmbr is the number of (identical) assets at the point (REAL)
- ckc is the conventional kill criteria code (INTEGER) (See APPENDIX A, Conventional Lethality Data.)
- nkc is the nuclear kill criteria (currently has no effect)
- tkc is the toxic kill criteria code (INTEGER) (See, e.g.,T.K.C.)
- cpst is the conventional posture code (INTEGER)
- npst is the nuclear posture code (INTEGER) (See SHIELDING)
- tpst is the toxic posture code (INTEGER) (See MOPP)

Option: Alternate posture:
Format: (R0)
$cpst*,npst*,time1 or
$tpst*,time2 or
$cpst*,npst*,time1,tpst*,time2

where:

- cpst* is the alternate conventional posture (INTEGER)
- npst* is the alternate nuclear posture (INTEGER)
- time1 is the mean time (intrvl) required to change to cpst*,npst* (REAL)
- tpst* is the alternate toxic posture (INTEGER)
- time2 is the time (intrvl) needed to change to tpst* (REAL)

(NOTE: If cpst*,npst* are specified, conventional and nuclear posture change begins at arrival of first round. See ALARM for start of toxic posture change.)

Option: MOPP ALL. This option provides a short cut to give all personnel the same alternate MOPP posture with the same mean time to change.

Format: (R1)
MOPP ALL, tpst*, time2

NOTES: The MOPP ALL card can be inserted anywhere in the deployment input set. If used, it supersedes any individual alternate MOPP postures.

Option: \OFFSET. This option allows adding specified values to the x and y coordinates of all deployment points which follow in the runstream. This option provides an easy method of displacing a section of a deployment. As denoted by the leading backslash, this is an embedded procedure command. (See section II.B.6.).

Format: (R1)
\OFFSET, x-displacement, y-displacement (REAL)

NOTE: The offset generated by this command is ADDITIVE to the offset generated by the OFFSET command or by the OFFSET card in the execute stream. (See sections II.A and IV.B.4.).

3. **MOPP.** This option allows the user to define the toxic posture (MOPP) codes (used in DEPLOYMENT for initial and alternate toxic posture) and specify a transmission factor for each. The transmission factor is used to reduce the dosage received by an individual or contamination received by a piece of equipment for an asset in the specified toxic posture. This command is also used to set a number of options dealing with the assumption of alternate MOPP posture.

Format: (R1)
MOPP
verbal description (<13 letters), tpst, tfv, tfp

where:

- tpst is the toxic posture code number (INTEGER)
- tfv is the fraction of vapor dose still received in posture (REAL)
- tfp is the fraction of percutaneous dose still received in posture (REAL)

If tfp is not given, tfp is set to tfv.

(NOTE: Codes 0 - 4 default to "OPEN", "MOPP I", "MOPP II", etc., with transmissions decreasing from 1.0 to 0.0)

Options: These set various MOPP change parameters:
Format: (R1)

ALL CLEAR YES, or ALL CLEAR NO. This option automatically returns unit individuals to initial MOPP posture when the last contaminant evaporates or is removed by decontamination.
Default is ALL CLEAR YES.

PROXIMITY, dist (REAL). This option allows specifying the (x and y) distance from a warhead within which an asset must be in order to "feel threatened" by an incoming round and change MOPP posture. (See ROUND YES, immediately below.)
Default, dist = infinity.

RECONSTITUTION YES, or RECONSTITUTION NO. This option causes all personnel to assume alternate MOPP posture during a reconstitution while contaminant is present.
Default is RECONSTITUTION NO.

RECOVERY *TIME*, trcvr *(REAL)*. *This option* inputs the time (intrvl) needed to realize that contamination is no longer present and to reassume initial MOPP posture.
Default, trcvr = 30.

ROUND YES, tfalse (REAL) or ROUND NO. This option controls whether individuals assume alternate MOPP posture

upon any incoming round. If ROUND YES, the value tfalse is the (false alarm recognition) time (intrvl) needed to recognize that none of the incoming rounds were toxic and to return to initial MOPP posture.
Default, ROUND YES, tfalse = 10.

TIME SPREAD, sigma (REAL). This option inputs the fractional standard deviation in time needed to change MOPP posture.
Default, sigma = 0.2.

4. **OFFSET.** This option reads x and y offset values which are added to the x and y coordinates of every deployment point before the analysis commences. This allows the user to use one generic deployment of a unit, centered around 0.,0., and displace the entire deployment to specific (battlefield) locations.

Format: (R0)
OFFSET
x-offset (REAL), y-offset (REAL) in UNIT coordinate system

NOTE: Offsets can also be input via the computer's normal input channel as part of program execution. In case of conflict, offsets input via the execution stream take precedence. (See EXECUTION, section A.) The offset input via this command or via the execution stream is ADDITIVE to any offset input via the \OFFSET option under DEPLOYMENT. (See section B.2.).

5. **REST.** This option specifies the places to which personnel assets deploy when assigned to rest. (See FATIGUE.) If no rest location is specified for an asset, it is assumed that the asset will rest at his duty station.

Format: (R1)
REST
asset name, x, y, ckc, nkc, tkc, cpst, npst, tpst

Option: Alternate postures:
Format: (R0)
$cpst*,npst*,time1 or
$tpst*,time2 or
$cpst*,npst*,time1,tpst*,time2

(NOTE: See DEPLOYMENT for definition of data. This input is identical to DEPLOYMENT except 1) nmbr is omitted and 2) ASSET NAME need not be unique.)

6. **SHIELDING.** This option allows the user to define nuclear posture code numbers. Shielding associates a verbal description and a transmission matrix with the code number.

Format: (R1)
SHIELDING
verbal description, npst, trns1, trns2, trns3, trns4

where:

npst is the code number (INTEGER, between 4-61) and the transmission factors are defined by:

trns1 = (n,n*)
trns2 = (g,n*) (usually = 0)
trns3 = (n,g*)
trns4 = (g,g*)

where:

n indicates neutron
g indicates gamma and
(a,b*) indicates dosage of type b in the posture due to an incident dosage of type a.

NOTES: If only trns1 is given, it is used for trns1 and trns4; trns2 and trns3 are set = 0. Nuclear posture codes 1, 2 and 3 are reserved for OPEN, OPEN-BUT-THERMALLY-SHIELDED and FOXHOLE, respectively. No vehicle can be associated with posture codes 1 - 3. Codes 4 and 5 default to APC and TANK; however, the user must associate any vehicular blast protection. (See following option.)

Option: An asset (usually a vehicle) can be associated to nuclear posture codes 4 through 61. Doing so gives an individual in the posture the same blast criteria as the associated vehicle.

Format: (R1)
$associated asset name

7. **T.K.C. (TOXIC KILL CRITERIA).** This option allows the user to define his toxic kill criteria code numbers. T.K.C. associates with the code number a verbal description and a chemical dosage multiplier (used to simulate higher (or lower) than normal ratio of dose/dosage, as would be acquired by a person whose task required a higher (or lower) than normal breathing rate). This input also allows specifying heat stress parameters, which are also job dependent. The effect of the TOXIC KILL CRITERION code number in general, and this option in particular, is to allow the user to indicate deployment points at which a difficult (or easy) job is being done. Any individual assigned to that deployment point inherits the difficulties of the job. (See also DEGRADATION, section B.1.).

Format: (R1)
T.K.C.
verbal description (<13 letters), tkc, dm, pcas, tlag, tau

where:

tkc	is the toxic kill criteria code number (INTEGER, 1-20)
dm	is the dosage multiplier(REAL)
pcas	is the probability of heat stress in alternate MOPP (REAL)
tlag	is the heat stress lag time (intrvl) (REAL)
tau	is the characteristic probability growth time (intrvl) (REAL)

(NOTE: pcas, tlag and tau are optional)

C. WEAPON INPUTS

The following data sets input parameters relating to weapon delivery system performance and weapon arrival events.

1. **ACQUISITION PROBABILITY.** This option inputs a single probability number which represents the probability that the unit has been acquired. This option also allows a change in probability via a change event. A random number is drawn against the current probability value and the acquisition status redetermined at the beginning of every replication and upon every change event. Before every lethality event, acquisition status is checked: if in a non-acquired state, lethality events are skipped.

Format: (R0)
ACQUISITION PROBABILITY
pacqr

where:
pacqr is the target acquisition probability

Option: This option can be used to input an ACQUISITION PROBABILITY change event. This allows the user to simulate a point in time at which the probability of target acquisition changes, as would be caused by unit movement. It also causes a new random number to be drawn. Thus, to model independent attacks on parts of a separated unit, use of this option will cause acquisition probabilities to be uncorrelated.

Format: (R0)
time, pacqr

where:

time is the (clock) time at which the change in error occurs (REAL)

2. **AGENT.** This option associates a toxic agent type with a specific weapon.

Format: (HR)
AGENT
weapon name, atyp

where:
atyp is G, V, or H. Default, atyp = G.

3. **CEP ERROR.** This option reads delivery errors expressed in circular error probable (CEP). (CEP is that radius within which one half of the rounds are expected to fall.)

Format: (R1)
CEP ERROR
weapon name, indcep, corcep, hob

where:

indcep is the independent circular error (REAL)
corcep is the volley-correlated circular error (REAL)
hob is the standard deviation in height-of-burst (REAL)

IMPORTANT NOTE: Values of errors > 0 result in errors being drawn from a Gaussian distribution having a shape parameter derived from the input error value. Values < 0 result in errors being drawn from a uniform distribution having a range parameter derived from the input error value. Positive and negative values may be mixed on the same card.

Option: This option can be used to input a CEP ERROR change event. This allows the user to simulate a point in time at which the error in incoming fire changes, as would be caused by unit movement.

Format: (R1)
weapon name, time, indcep, corcep, hob

where:

time is the (clock) time at which the change in error occurs (REAL)

4. **CEP TLE.** This option reads target location errors expressed in circular error probable (CEP).

Format: (R0)
CEP TLE
tlecep

where:

tlecep is the target location error, expressed as a circular error probable (REAL)

NOTE: See IMPORTANT NOTE under CEP ERROR for choice of distributions.

Option: This option can be used to input a CEP TLE change event. This allows the user to simulate a point in time at which the error in target location changes, as would be caused by unit movement.

Format: (R0)
time, tlecep

where:

time is the (clock) time at which the change in error occurs (REAL)

5. **DELIVERY ERROR.** This option inputs weapon delivery errors as standard deviations in RANGE and DEFLECTION.

Format: (R1)
```
DELIVERY ERROR
weapon name, rind, rcor, dind, dcor, hob
```
where:

- rind is the independent error in range (REAL)
- rcor is the volley-correlated error in range (REAL)
- dind is the independent error in deflection (REAL)
- dcor is the volley-correlated error in deflection (REAL)
- hob is the error in height-of-burst (REAL)

NOTE: See IMPORTANT NOTE under CEP ERROR for choice of distributions.

Option: This option can be used to input a DELIVERY ERROR change event. This allows the user to simulate a point in time at which the error in weapon delivery changes, as might be caused by unit movement.

Format: (R1)
```
weapon name, time, rind, rcor, dind, dcor, hob
```
where:

- time is the (clock) time at which the change takes place (REAL)

6. **INCOMING FIRE DIRECTION.** This option orients the incoming fire (RANGE) direction with respect to the UNIT Coordinate system. See discussion of coordinate systems in section B.8.

Format: (R0)
INCOMING FIRE DIRECTION
incang

where:

incang is the angle (degrees) from the UNIT x direction to the RANGE direction (INTEGER, positive if counter-clockwise).
Default, incang = 0.

Option: Incoming fire direction change event.

Format: (R0)
time, incang

where:

time is the (clock) time at which the incoming fire direction changes (REAL).

Option: The incoming fire direction can also be specified as a range of directions, in which case the code will randomly select a new direction within the specified range for each replication.

Format: (R0)
incang1, incang2

where:

incang1 and incang2 are the angles between which the incoming fire directions will lie (INTEGERS)

Option: Incoming fire direction change event, specified as a range.

Format: (R0)
time, incang1, incang2

7. **MISS DISTANCE.** This option inputs a fixed distance by which a weapon will miss its aimpoint. Rounds will be randomly located on a circle of radius 'miss distance' away from the designated burst point.

```
Format:  (R1)
         MISS DISTANCE
         weapon name, mdis, hob
where:

         mdis  is the miss distance (REAL)
         hob   is the error in height-of-burst (REAL)
```

NOTE: hob is optional. If used, see IMPORTANT NOTE under CEP ERROR for choice of distributions.

8. **RELIABILITY OF WEAPONS.** This option causes randomly selected rounds or volleys to fail to function.

```
Format:  (R1)
         RELIABILITY OF WEAPONS
         weapon name, rrnd, rvol
where:

         rrnd   is the single warhead reliability (REAL)
                (Default = 1.)
         rvol   is the single warhead reliability (REAL)
                (Default = 1.)

         NOTE:  If only one value is given, it is assumed to be
         rrnd
```

9. **ROUND.** This option inputs the parameters necessary to specify an attack by a single round.

```
Format:  (R1)
         ROUND
         weapon name, time, apx, apy, apz
where:

         time  is the (clock) time-of-arrival of the round (REAL)
         apx   is the intended x-coordinate (UNIT Coordinate sys-
               tem) (REAL)
         apy   is the intended y-coordinate (UNIT Coordinate sys-
               tem) (REAL)
         apz   is the intended height-of-burst (REAL)
```

10. **TLE.** This option reads target location errors expressed as standard deviations.

Format: (RO)
TLE
tlex, tley

where:

tlex is the x-coordinate of the target location error (REAL)

tley is the y-coordinate of the target location error (REAL)

NOTE: See IMPORTANT NOTE under CEP ERROR for choice of distributions.

Option: This option can be used to input a TLE change event. This allows the user to simulate a point in time at which the error in target location changes, as would be caused by unit movement.

Format: (RO)
time, tlex, tley

where:

time is the (clock) time at which the change in error occurs (REAL)

11. **VOLLEY.** This option inputs parameters necessary to specify an attack by a volley of weapons. The intended pattern is assumed to be a line of the specified width and at the specified angle with respect to the INCOMING FIRE direction. A cluster-type munition (scatter about a single point) can be modeled by specifying a volley pattern width of 0.

Format:
```
(R1)
VOLLEY
weapon name, time, papx, papy, papz, nrd, ang, width
```

where:

- time — is the (clock) time of the attack (REAL)
- papx — is the intended x-coord. of the pattern midpoint (REAL)
- papy — is the intended y-coord. of the pattern midpoint (REAL)
- papz — is the intended height-of-burst of the rounds (REAL)
- nrd — is the number of rounds in the volley (INTEGER)
- ang — is the angle (degrees) of the pattern line with respect to the incoming direction. (REAL)
- width — is the width of the pattern line (REAL)

Option: This option, used after a normal volley data card (above), creates multiple volleys with the specified time between volleys, each one "stepped" in the specified direction by the specified distance. This option allows easy modeling of a moving barrage.

Format:
```
$ nvol, dtm, dir, dis
```

where:

- nvol — is the number of ADDITIONAL volleys (INTEGER)
- dtm — is the time-between-volleys (intrvl) (REAL)
- dir — is the angle (degrees) of movement of the intended pattern midpoint measured counter-clockwise from the x direction (in the UNIT Coordinate system) (REAL)
- dis — is the distance of movement (REAL)

12. **WIND DIRECTION.** This option orients the wind (RANGE) direction with respect to the UNIT Coordinate system. See discussion of coordinate systems in section II.B.9.

Format: (R0)
WIND DIRECTION
windang

where:

windang is the angle from the UNIT x direction to the RANGE direction (INTEGER, positive if counter-clockwise).
Default, windang = 0.

Option: Wind direction change event.
Format: (R0)
time, windang

where:

time is the (clock) time at which the wind direction changes (REAL).

Option: The wind direction can also be specified as a range of angles, in which case a new wind direction is randomly selected within the specified range for each replication.

Format: (R0)
windang1, windang2

where:

windang1 and windang2 are the angles within which the wind direction will lie (INTEGERS)

Option: Wind direction change event, specified as a range.
Format: (R0)
time, windang1, windang2

13. **YIELD.** This option inputs the yield of nuclear weapons.

Format: (R1)
YIELD
weapon name, yld1, yld2

where:

yld1 is the blast and thermal yield in kt (REAL)
yld2 is the effective radiative and EMP yield in kt (REAL)

NOTE: If only yld1 is given, it is used for both yld1 and yld2.

D. WEAPON EFFECTS

The following data sets input parameters relating to the effects of weapons on assets.

1. **CONVENTIONAL LETHALITY.** This option causes the AURA code to read a conventional lethality data file via input unit #2. No further data is needed. (See Appendix A for the format of the conventional lethality data file.)

Format: (not applicable)
CONVENTIONAL LETHALITY DATA

2. **DOSE PARAMETERS.** This option allows changing various parameters which control the personnel-response-to-dosage algorithms and the output report of dosages.

Format: (all R1)
DOSE PARAMETERS
options

Option: Set values for dosage "bins" for dosage distribution report in output.

Format: (R1)
DOSE BINS,b1,b2,b3,b4,...b10

where:

b1 is the center value of the first bin etc.
Defaults: Appropriate for nuclear and toxic.

NOTES: There must be 10 increasing bin values. Do not input b1 = 0. b10 may = MAX DOSE, but may not exceed it.

Option: Set maximum dosage to be considered as instant casualty

Format: (R1)
MAX DOSE, maximum dosage value (REAL)
Defaults: = 200. Gy (Nuclear); = inf. (Toxic)

Option: Set minimum dosage to be considered for lethality, ETI.

Format: (R1)
MIN DOSE, minimum dosage value (REAL)
Defaults: = 4.5 Gy (Nuclear); = 1.0 (Toxic)

Option: Turn on/off ETI, PCI and dose-related degradation of performance algorithms. Not of general usefulness.

Format: (R1)
NUCNTL, control code number (INTEGER)
(See INFO source file (BRL) for particulars.)
Default: All algorithms operant.

3. **NUCLEAR VULNERABILITY.** This option causes the AURA code to read a nuclear vulnerability data file via input unit #3. No further data is needed. (See Appendix A for the format of the nuclear vulnerability data file.)

Format: (not applicable)
NUCLEAR VULNERABILITY DATA

4. **PERSISTENCE.** This option allows changing the persistence time for chemical agents on specified assets from the standard (uniform) persistence time produced by the toxic dissemination code (NUSSE3). The change is affected by specifying the ratio of time-to-evaporate/diffuse from the asset to time-to-evaporate as output by NUSSE3.

```
Format:  (R1)
         PERSISTENCE
         weapon name1
         asset1, frc11
         asset2, frc12
         .....
         weapon name2
         asset1,frc21
         etc.
where:

         frcIJ is the ratio of evap time for agent from weapon I
               on the asset J to the NUSSE3 output evap time.
```

NOTE: TOXIC DISPERSION command must precede this option.

5. **THERMAL.** This option allows the user to specify the atmospheric thermal transmissivity and the type of uniform being worn. These parameters affect the calculation of thermal casualties.

```
Format:  (all HR)
         THERMAL
         options

Option: Thermal transmissivity
Format:  (HR)
         ATMOSPHERE, quality
where:

         quality must be GOOD, AVERAGE, or POOR
         Default is AVERAGE

Option: Type of uniform
Format:  (HR)
         UNIFORM, type
where:

         type  must be SUMMER or WINTER
         Default is SUMMER for initial posture, WINTER for alter-
               nate MOPP posture
```

6. **TOXIC DISPERSION.** This option causes the AURA code to read a toxic dispersion data file via input unit #4. No further data is needed. (See Appendix A for the format of the toxic dispersion data file.)

Format: (not applicable)
TOXIC DISPERSION DATA

E. INDIVIDUAL TASK INPUTS

The following data sets input parameters that model the specific tasks that are performed by individuals and equipments within the unit (LINKs) and alterations to such tasks (e.g., task-dependent degradations).

1. **FATIGUE.** This option allows the user to specify that different jobs (LINKs) may be more or less demanding than others, both in terms of the need for personnel to be rested and the drain upon personnel who are engaged in the job.

Format:
```
(R1)
FATIGUE
LINK name, rfr, rd
```
where:

rfr is the relative fatigue rate (REAL)
rd is the relative demand-for-stored-rest (REAL)

2. **LINKS.** This option inputs data on basic subtasks including:

1. relationships between number of effective assets allocated to an individual task and effectiveness of task performance (see figure 1),

2. limitations on numbers of ENTITIES (i.e., actual number of personnel or items of equipment assigned, regardless of relative worth of each entity) which may be assigned to task

3. substitutes, i.e. assets which may be assigned to a task other than the task's HOMELINK assets (which are automatically assignable) (See HOMELINK, section II.B.7)

NOTE: All LINKs to which assets may be assigned must be deployed, either through being a HOMELINK for an asset that is deployed or by being a specifically deployed DUMMY LINK (See DUMMY LINK in section II.B.7 and DEPLOYMENT in section B.2.) However, also see note on DUMMYLINKs, below.

Format: (R1)
LINKS
LINK name, cap100, maxeff, entmax

where:

LINK name is any allowed name (see section II.B.5). If LINK name is also the name of an asset, this entry defines the *HOMELINK for* that asset.

cap100 is number of eff. assets needed for maximum effectiveness (REAL)

maxeff is maximum effectiveness IN PERCENT (INTEGER) Default, maxeff = 100

entmax is maximum number of items that may be assigned to LINK.
entmax is taken as an absolute value unless an ASSOCIATED LINK is defined, in which case entmax is taken as the number per item in the ASSOCIATED LINK (See ASSOCIATED LINK immediately below.) (REAL) Default, entmax = unlimited.

Option: Substitutes. Substitutes are defined by sets of three cards.

Format: (HR)/(R1)/(R1)
$subnm1, subnm2, subnm3, ...
$E, eff1, eff2, eff3, ...
$T, tim1, tim2, tim3, ...

where:

subnmI is the name of the Ith substitute (need not be a unique name)

effI is the effectiveness of the Ith substitute (relative to normal assets implied in specifying cap100) (REAL)

timI is the time (intrvl) needed for the Ith asset to substitute

NOTE: It is essential that each substitute card be followed by an effectiveness card and a time card. If the number of substitutes exceeds the capacity of the first substitute card, additional substitutes may be specified by following the first set of three cards with other sets.

Option: Specify lower bounds on LINK effectiveness

Format: (R1)

$M, cap0, mineff

where:

cap0 is number of eff. assets below which effectiveness is minimum (REAL) Default, cap0 = 0.

mineff is minimum effectiveness IN PERCENT (INTEGER) Default, mineff = 0

Option: Specify granularity (maximum amount of any asset) for allocation into LINK in any one installment. Useful, for example, to cause allocation scheme to consider other LINKs before totally allocating any asset to one LINK.

Format: (R1)

$G, grnlnk

where:

grnlnk is the maximum amount to be allocated in any one pass (REAL). Default, grnlnk = 0.5

Option: Default Granularity. This option allows changing the default value for the LINK granularity (the maximum amount the will be substituted into a LINK during any one pass of the allocation algorithm). This command changes the granularity for **ALL** LINKs which do not have a granularity specified via the $G option, no matter where the command is placed in the runstream.

Format: (R1)

GRANULARITY, new default (REAL)

Option: Specify an ASSOCIATED LINK. An ASSOCIATED LINK is another subtask whose potential fulfillment controls the entities assignable to the current subtask. For example, if there can be only two operators per system X, this option can be used as follows: System X would be defined as a normal LINK. The operator LINK would be defined with 2. for the value of entmax on card 1 and System X as the associated LINK.

Format:
    $A, associated LINK name

NOTE on DUMMY LINKS: A dummy LINK without substitutes is allowed. (Such a construct can be used, for example, to provide a "sure path" through a COMPOUND LINK.) Such LINKs need not be deployed.

3. **SUBLETHAL DOSE DEGRADATION.** This option associates specified LINKs with particular dose-time degradation sets which can be used, e.g., to degrade dosed individuals assigned to physically demanding jobs more severely than those in cognitive jobs. Two degradation sets are built into AURA which describe: a gunner (degradation set number 0), typical of most jobs and an ammunition loader (degradation set number -1), typical of a physically demanding job. The default for all LINKs (without use of this option) is degradation set 0. NOTE: Degradation code number -999 indicates no degradation.

Format: (R1)
    SUBLETHAL DOSE DEGRADATION
    LINK name, code number (INTEGER)

Option: Read new degradation data from input unit #11. This option also allows reading additional degradation sets to which new degradation set numbers (1 - 5) are given. These sets are then available for association with specific LINKs. NOTE: Format for the data on unit #11 is given in Appendix B.

Format: (HR)
    READ

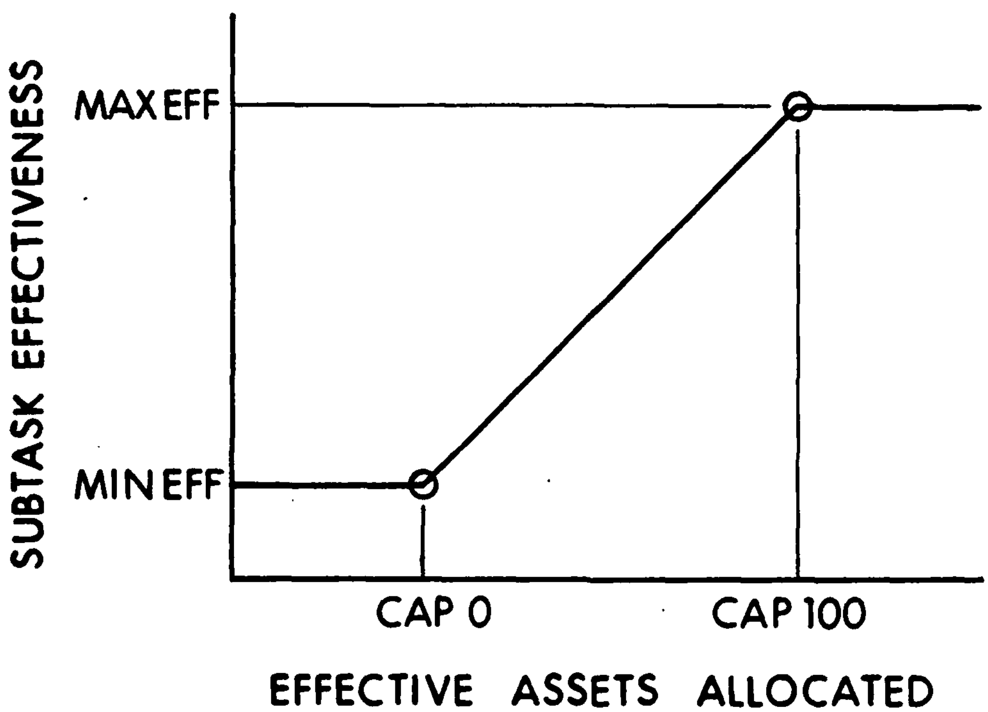

Figure 1. General Form of a Link Effectiveness Curve

F. UNIT FUNCTION INPUTS

The following data sets input parameters which model the functional structure of a unit. (See FUNCTIONAL STRUCTURE, section II.B.7 and Appendix B.)

1. **CHAINS.** This option allows the (overall) "ANDing" together of sets of operations (called SEGMENTS) to satisfy the requirements of missions, one CHAIN per mission. Segments may be any previously defined LINKs, CREWs, SUBCHAINs, ORLINKs or COMPOUND LINKs.

```
Format:  (HR)
         sg11, sg12, sg13, sg14, ....
         sg21, sg22, sg23, .....
         etc.
where:
         sgIJ is the Jth segment in CHAIN I.
```

Option: A continuation line which begins with the word TIME (may be abbreviated to T) is interpreted as a set of (clock) times during which the preceding CHAIN is operant. Thus the available missions for a unit may change during the encounter.

```
Format:  (R1)
         $T, str1, stp1, str2, stp2, ....
where:

         strI  is the (clock) time that a CHAIN becomes operant
         stpI  is the corresponding time that it ceases to be
               available
```

2. **COMPOUND LINKS.** This option inputs data for COMPOUND LINKs (CPLINKs), i.e. functional structures which are weighted summations of parts (called CPPARTs). A CPPART may be a simple LINK, a CREW, a SUBCHAIN or an ORLINK. NOTE: A COMPOUND LINK name MUST begin with the ! character.

```
Format:  (R1)
         COMPOUND LINK
         compound link name1 (must begin with !)
         cppart11, wt11
         cppart12, wt12
         ....
         compound link name2
         cppart21, wt21
         ....
where:
```

compound link name is of the form !xxxxx (e.g. !LOADING TECHNIQUE). (See section II.B.5.)
cppartIJ is the name of the jth part of CPLINK
wtIJ is the weight of cppart IJ

NOTE: The weights for any CPLINK usually sum to 1. A warning message is printed if summation is not 1., but run will proceed.

3. **CREW.** This option inputs data to define functional structures (called CREWs) that represent sets of subtasks that must work together to accomplish part of a mission. The relationship between tasks in a CREW are generalized parallel structures, as opposed to strict ANDs as are those in a SUBCHAIN. (See SUBCHAIN, below.) Individual crew jobs must be LINKs. Each crew position has two parameters, alp and bet, associated with it. In addition, each crew has a parameter K. NOTE: A CREW name must begin with the = character.

```
Format:  (R1, R0, R1, R1, ....)
         CREW
         crew name1 (must begin with =)
         N1, K1
         LINK1, alp1, bet1
         LINK2, alp2, bet2
         ....
         crew name2
         N2, K2
         ....
```

where:

crew name is of the form =xxxxx (e.g. =GUN CREW). (See section II.B.5.)

Nj is the number of members in the jth crew (INTEGER)

Kj is the crew parameter for the jth crew (REAL)

lnkI is the name of the Ith LINK in the crew

alpI and betI are the parameters for the Ith task in the crew

4. **ORLINKS.** This option inputs data to define functional structures (called ORLINKs) that represent mutually exclusive choices for the accomplishment of part of a mission. The choices, called "branches", may be SUBCHAINs, CREWs or simple LINKs. NOTE: An ORLINK name must begin with the + character. 1 and 23.

```
Format:  (HR)
         ORLINKS
         orlink name, br1, br2, br3, .... (name must begin with +)
```

where:

orlink name is of the form +xxxx (e.g. +COMMO). (See section II.B.5.)

brI, the Ith branch, is the name of a LINK, CREW or SUBCHAIN

5. **SUBCHAINS.** This option inputs data to define functional structures (called SUBCHAINs) that represent sets of subtasks that must work together to accomplish part of a mission. The relationship between elements in a SUBCHAIN are simple ANDs: the value of a SUBCHAIN equals the value of its weakest member. In contrast, see CREWs, above. The elements of a SUBCHAIN may be CREWs and/or simple LINKs. NOTE: A SUBCHAIN name must begin with the * character.

Format: (HR)
SUBCHAINS
subchain name, lnk1, lnk2, lnk3, ... (name must begin with *)

where:

subchain name is of the form *xxxx (e.g. *FORKLIFT TEAM). (See section II.B.5.)

lnkI is the name of the Ith LINK or CREW in the SUBCHAIN

G. PROGRAM CONTROLS

The following data sets input parameters that control the running of the code, the decision logic used, the outputs produced and scenario parameters such as the length of (clock) time of the encounter.

1. **DECISION RULES.** This command allows resetting certain default conditions and values that control the rules by which the allocation algorithm chooses assets for assignment into the various LINKs. The allocation algorithm (commander) normally considers versatility before considering the order in which the user listed assets on a substitution card. (See LINKS, section E.2.) This command allows choosing whether versatility or user-input-order is to be the first consideration. (See ALLOCATION DECISION RULES, section II.B.8.).

Format: (HR)
DECISION RULES
PRIORITY, first consideration

where:
first consideration is VERSATILITY (default) or ORDER

Option: Significance. This option allows specifying the fractional improvement needed before the allocation algorithm will ignore differences in order and versatility. (e.g., signif = 0.5 means that an asset having 1.5 times the effectiveness of the current best choice will automatically become current-best-choice, regardless of versatility or order.)

Format: (R1)
SIGNIFICANCE, signif

where:
signif is the required improvement (REAL).
Default, signif = 0.005.

Option: Sickness level. This option allows specifying the degraded level of performance (e.g., due to sickness or fatigue) that an asset must show in order to be preempted in its HOMELINK by a substitute. (e.g., sicklv = 0.75 means that if a HOMELINK asset were at effectiveness 0.5 and a substitute were available at effectiveness greater than 0.67 (= 0.5 / 0.75), then the HOMELINK asset would be replaced in his job.)

Format: (R1)
SICKLV, sicklv

where:
sicklv is the required degradation (REAL). Default, sicklv = 0.75.

Option: Finish repair. This option allows specifying the relative worth in finishing an ongoing NEEDED repair rather than starting another NEEDED repair. (e.g., fnshrp = 2.0 means that, if an ongoing repair has 0.6 of an item left to fix, the code will calculate the anticipated gain based upon receiving 1.2 (= 2.0 * 0.6) items. This would be compared to the anticipated gain from any other repair which might be initiated in order to solve a mission-accomplishment limitation. The repair which promises the most gain is the one selected by the optimization algorithm to add to the mission requirements to calculate the mission cost of (NEEDED) repair. Note that this parameter only influences NEEDED repairs, as defined in section A.8.).

Format: (R1)
FINISH REPAIR, fnshrp

where:
fnshrp is the relative gain (REAL). Default, fnshrp = 2.0

2. **GO.** This command indicates that all data has been entered and the simulation and analysis should begin.

Format: (not applicable)
GO

3. **HEADING.** This option allows the user to write a message at the beginning of the AURA output and again at the beginning of the results. This message is in addition to any message entered via the computer normal input channel during execution. (See EXECUTION, section A.)

Format:
HEADING
message

4. **INTERNAL RECONSTITUTION TIMES.** This option inputs a matrix of times (intervals) following a lethality or reconstitution event at which outputs are desired. Every lethality event (see ROUND and VOLLEY, sections C.9 and C.11) and reconstitution event (see RECONSTITUTION EVENT, section G.7) causes the internal clock to reset. Then, at the end of interval 1, interval 2, etc., the code updates all time dependent factors, reallocates assets and compiles all statistics to be used in the final results. In this way, the user sets up the time points at which results will be reported.

Format: (R0)
INTERNAL RECONSTITUTION TIMES
tm1, tm2, tm3, ... (maximum, 47 times)

where:
tmI is the Ith time point (intrvl) after a lethality event

5. **MODE.** This option controls certain choices in operation of the code.

Option: CODE mode. Used in code development to track internal parameters.

Format: (HR)
MODE
CODE, ON or OFF (Default = OFF)

Option: CULL mode. In CULL mode, incoming rounds are screened (in the ROUND and VOLLEY input routines) to predetermine if the round has a potential of affecting a target point. This allows using one, standard, large, scenario-wide threat in the run-streams for all targets in the the scenario: AURA will cull out only those weapon employments which might affect the unit being studied in each runstream. NOTE: If running in CULL mode, weapon employment data (ROUND or VOLLEY) may not be followed by weapon characteristic data (DEPLOY, DELIVERY, TLE, CONVENTIONAL, or TOXIC.)

Format: (HR)
CULL, ON or OFF (Default = OFF)

Option: DEBUG mode. Causes the code to process input data but not execute. Used to debug runstreams.

Format: (HR)
DEBUG, ON or OFF (Default = OFF)

Option: PRIORITY mode. Changes the interpretation of COMPOUND LINKs. In PRIORITY mode, parts of COMPOUND LINKs (CPPARTs) are considered in order of entry AND failure to improve one cppart stops the optimization process. If the following restrictions apply:

a. unit structure consists of a single COMPOUND LINK whose CPPARTs are all simple SUBCHAINs
b. every LINK is modeled as a 0.- 1. step function (i.e. a job is either at 100% or else 0%)
c. all substitutes are 100% capable
d. degradation of assets is not played
e. multiple assets cannot be assigned to a single job

then PRIORITY mode causes the AURA optimization algorithm to emulate the AMORE linear program allocation algorithm.

Format: (HR)
PRIORITY, ON or OFF (Default = OFF)

Option: STOCHASTIC mode. Causes all lethality assessments to be stochastically determined. In STOCHASTIC mode, the AURA lethality routines draw random numbers against calculated probabilities to determine damage or kill. (In normal mode, fractional kills are tallied.)

Format: (HR)
STOCHASTIC, ON or OFF (Default = OFF)

Option: TIME BEFORE ZERO. This option allows resetting the assumed time that has elapsed before start of analysis, thus limiting the substitutions that can be assumed for the time = 0. reconstitution. The default time, infinite, allows all needed substitutions to be made before commencement of analysis.

Format: (R1)
TIME BEFORE ZERO, time (intrvl) (REAL)

6. **OUTPUT.** This option controls the printing of optional outputs from an AURA run. The definition, interpretation and use of these outputs is the subject of a report to be published.

##

The OUTPUT Options

BINS	FATIGUE	PRINT
CASUALTIES *	INPUT LISTING	RANDOM NUMBER
CHAIN	ITERATION	RECONSTITUTION
DEPLOYMENT	LETHALITY	REPAIR REPORT
DOSE *	LINK SUMMARY	SUMMARY
DUMP8	OPTIMIZE	TIMER
DUMP9	POSTURE *	WEAPON
ETIPCI		

* - indicates options affected by PARTICULAR ASSETS option

Option: BINS. Reports the contents of all radiation accumulation bins for every asset at reporting times that fall within the specified intervals.

Format: (R1)
BINS, strt1, stp1, strt2, stp2, ...

where:

strtI is the start time (clock) for the Ith interval (REAL)
stpI is the ending time (clock) for the Ith interval (REAL)
Default, no bin reports output.

Option: CASUALTIES. Reports all casualties, contaminations, ETI episodes and expenditures as they occur. If WEAPON report is not ON, this option also describes incoming warheads which cause immediate casualties. (See PARTICULAR ASSETS option.)

Format: (HR)
CASUALTIES, ON or OFF (Default, OFF)

Option: CHAIN. Prints line-printer depiction of unit functional structure.

Format: (HR)
CHAIN, ON or OFF (Default, ON)

Option: DEPLOYMENT PLOT. Prints line-printer depiction of unit deployment, including initial wind and incoming fire directions. (See SELECTIVE PLOT option.)

Format: (HR)
DEPLOYMENT PLOT, ON or OFF (Default, ON)

Option: DOSE. Reports all nuclear or toxic doses as received (cumulated to the nearest reporting time). (See PARTICULAR ASSETS option.)

Format: (HR)
DOSE, ON or OFF (Default, OFF)

Option: DUMP8. Causes reallocation information (time, effectiveness, weak LINKs, strongest CHAIN) to be written onto output unit #8 at every reporting time.

Format: (HR)
DUMP8, ON or OFF (Default, OFF)

Option: DUMP9. Causes incoming weapon information, including actual burst points for every warhead in every replication, to be written onto output unit #9. This information is used by the BRL graphical post-analysis utility programs to aid in analysis of results.

Format: (HR)
DUMP9, ON or OFF (Default, OFF)

Option: ETIPCI. If ON, causes at-end average of nuclear dose effects (average performance degradations, Early Transient Incapacitation episodes and Permanent Incapacitation occurrences) to be printed. If FULL, also causes intermediate information to be written onto output unit #10.

Format: (HR)
ETIPCI, ON or FULL or OFF (Default, ON)

Option: FATIGUE. If RECONSTITUTION (see below) is ON or PARTIAL, this option prints out the fatigue status of all assets.

Format: (HR)
FATIGUE, ON or OFF (Default, OFF)

Option: INPUT LISTING. Causes the code-interpreted input data to be printed at beginning of output.

```
Format:  (HR)
         INPUT LISTING, word1, word2, word3, ...

where:   word =

         ON        to turn on full beginning output (Default)
         OFF       to turn off all beginning output
         EVENTS    to print event table
         WEAPON    to print weapon identification table
         ASSETS    to print asset identification table
         NUCLEAR   to print nuclear-related parameters
         TOXIC     to print toxic-related parameters
         REPAIR    to print failure and repair parameters
         LINKS     to print LINK definition table
         FUNCTIONAL to print substitution, SUBCHAIN,  ORLINK  and
                   CPLINK tables
         DEPLOYMENT to print deployment table
```

Option: ITERATION. Causes some results to be reported at the end of every replication.

```
Format:  (HR)
         ITERATION, ON or OFF  (Default, OFF)
```

Option: LETHALITY. Causes input units #2, #3 and/or #4 (conventional lethality file, nuclear vulnerability file and toxic dispersion file, respectively) (if used) to be rewound and copied onto the end of the AURA output.

```
Format:  (HR)
         LETHALITY, ON or OFF  (Default, ON)
```

Option: LINK SUMMARY. Causes an at-end report of the number of times the specified LINKs were weak at each reporting time as a function of the CHAIN being optimized.

```
Format:  (HR)
         LINK SUMMARY, LINK1, LINK2, ... (maximum, 12)
or       LINK SUMMARY, OFF  (Default, OFF)
```

Option: OPTIMIZE. Causes a highly detailed report of every step attempted in every optimal reallocation process which occurs during the specified intervals.

Format: (R1)
OPTIMIZE, str1, stp1, str2, stp2, ..., iunwlk

where:

strI is the start time (clock) for the Ith interval (REAL)

stpI is the ending time (clock) for the Ith interval (REAL)

iunwlk is an optional flag to redirect this output (INTEGER)

iunwlk .GE. 0 = output on printer
.LE. 0 = output on output unit #13
(Default, iunwlk=1)

Default, no optimization steps reported.

Option: PARTICULAR ASSETS. Restricts the assets to be included in casualty reports, dosages, contamination reports, etc.

Format: (HR)
PARTICULAR ASSETS, asset name1, asset name2, ...

NOTE: Common names can be used; any number of assets can be included.

Option: POSTURE. Causes a report of all posture changes. If ON, every individual is reported; if PARTIAL, group changes are reported. (See PARTICULAR ASSETS option.)

Format: (HR)
POSTURE, ON or PARTIAL or OFF. (Default, OFF)

Option: PRINT7. Causes all output to be sent to output unit #7.

Format: (HR)
PRINT7, ON or OFF. (Default, OFF)

Option: RANDOM NUMBER. Causes report of the random number seeds at the beginning of each replication.

Format: (HR)
RANDOM NUMBER, ON or OFF (Default, ON)

Option: RECONSTITUTION. Causes output after every reconstitution. If ON, a complete matrix of assets assigned versus LINKs is produced. If PARTIAL, only a summary of results with a report of assignments into the weak LINK is given. If ONCE, a complete matrix of asset assignments is given for the initial arrangement only. If time intervals are specified, a complete matrix of asset assignments is reported for any reconstitution which occurs during the specified intervals.

Format: (HR)
RECONSTITUTION, ON or PARTIAL or ONCE or OFF. (Default, OFF)

or

RECONSTITUTION, str1, stp1, str2, stp2, ...

where:

strI is the start time (clock) for the Ith interval (REAL) stpI is the ending time (clock) for the Ith interval (REAL)

Option: REPAIR REPORT. If ON, causes a report of all repair activities at each reporting time. If FULL, also gives a complete status of damaged items available for repair at each time.

Format: (HR)
REPAIR REPORT, FULL or ON or OFF. (Default, OFF)

Option: SELECTIVE PLOT. Restricts the assets depicted in the line-printer deployment plot. (See DEPLOYMENT PLOT option.)

Format:

SELECTIVE PLOT, asset name1, asset name2, ...

NOTE: Common names can be used; any number of assets can be included. Default: all assets included.

Option: SUMMARY. Gives an at-end sum of survivors vs. time, with standard deviations of the averages over replications. This output eliminates the need to add up the final output results for several assets. The user need only give the assets a common name (see NAMES, Chapter III) and request a summary on the common name via this option. Commonly used to summarize total casualties for PERSONNEL.

Format: (HR)
SUMMARY, common name1, common name2, ... (maximum, 6)
or SUMMARY, OFF (Default, OFF)

Option: TIMER. Allows user to measure the computer time used by various portions of an AURA run.

Format: (HR)
TIMER, opt
where opt is:

ON: times major segments
OFF: no timing
INPUT: times the input and preprocessor routines
OPTIMIZATION: times the various routines in the optimal allocation process
RECONSTITUTION: times over-all reconstitution process (updating, inventory, optimization, etc.)

Option: WEAPON. Causes a report of every weapon arrival (type, time, aimpoint, burstpoint).

Format: (HR)
WEAPON, ON or OFF (Default, OFF)

##

7. **RECONSTITUTION EVENT.** This option causes a reset of the internal time clock to restart a series of output time points. (See INTERNAL, section G.4.)

Format: (R0)
RECONSTITUTION EVENTS
time1, time2, ...

where:

timeI is the Ith (clock) time from which a series of reporting times will be generated. (REAL)

8. **REPLICATIONS.** This option controls the number of replications to be done.

Format: (R0)
REPLICATIONS
number of replications (INTEGER)

9. **SEEDS (Random Number).** This option allows presetting the random number seeds before beginning a run. This option is particularly useful if one wishes to repeat one particular replication of a previous run, perhaps with special output options turned on. Note that there are a number of random number sequences, each with its own seed, maintained in AURA.

Format: (R1)
SEEDS (RANDOM NUMBER)
sdnm, seedvalue

where:

sdnm is:

WEAPON, to set seed(1) which primarily effects weapon delivery processes
OTHER, to set seed(2) which effects processes not specifically seeded
FAILURE, to set seed(3) which effects random failures
STOCHASTIC, to set seed(4) which is used to select kills in stochastic mode. (See MODE, section G.5.)
HEAT CASUALTIES, to set seed(5) which selects heat stress casualties

seedvalue is the desired seed (REAL)
Defaults set by code.

10. **STOP.** This command indicates the end of the runstream file.

Format: (not applicable)
STOP

11. **TRACE.** This option causes reporting of specified LINKs being used or specified LINKs being weak. Useful as an aid to tracking down erratic or inexplicable LINK behavior.

Format: (HR)
TRACE
WEAK LINK, link name, rec
or USES, link name, rec
where:

link name is the name of the link being traced
rec is the reconstitution number of interest (i.e. the reconstitution in which a use or weak LINK occurrence of the specified LINK is to be reported).

or

rec can be the word ANY to report all such occurrences.

V. SUMMARY

This manual has presented the current repertoire of commands for the Army Unit Resiliency Analysis (AURA) computer code. It is clear from the number and diversity of commands that the methodology is extremely flexible, with algorithms designed to model, in-depth, a large, broad, detailed spectrum of battlefield factors.

Not covered in this report are the efforts being pursued at the Ballistic Research Laboratory to facilitate the use of AURA among users of differing technical and computational backgrounds. The first of these efforts are those dealing with aiding input preparation. Already in existence are interactive graphical programs for the generation of deployment data. These are being augmented with graphical packages for the generation of unit functional structure diagrams and data and off-line debugging routines.

Of particular importance is the work being done to create standard data bases for such AURA inputs as weapon characteristics, vulnerability/lethality, unit structures and deployments. It is currently envisioned that these data bases, wherever resident, will be accessible through DoD computer nets to allow user-friendly, interactive input file assembly by non-experts in the various technical areas. A nuclear vulnerability data base has long been available from the Harry Diamond Laboratory. Although quite incomplete, joint efforts by the HDL and the BRL have resulted in a quick, easy-to-use tool for the preparation of nuclear vulnerability data files (input unit #3) for AURA nuclear runs. Work at the BRL has begun on the conventional lethality data base.

Finally, a set of interactive programs, perhaps involving expert system techniques, are planned to facilitate the analysis of AURA results. The currently operational actual-weapon-burst-point graphical extension to the deployment program is the first of these aids.

The goal of these efforts is to make it feasible for one-sided unit-level analyses to be made quickly, easily and uniformly throughout the diverse Army analysis community, using the most current data and algorithms available from the various proponent agencies. It is intended that the publication of this manual will be a first step toward that goal.

APPENDIX A

AURA INPUT FILES

Conventional Lethality (Unit #2)

The conventional lethality file inputs a weapon name (which may be a common name) as it appears in the WEAPON list under the NAMES mnemonic in the runstream file. Following the weapon name are a series of asset names (which also may be common names) from the ASSET list under NAMES. Under each asset name is a list of conditions, followed by the parameters that describe the vulnerability of the asset to the weapon under the conditions listed.

Three conditions are covered in the lethality data: height-of-burst (HOB) of the incoming weapon, posture of the asset and kill criteria. The format of the input routine requires that lethality data be given for all combinations of the three conditions. Thus, if two HOB, four postures and three kill criteria are listed, there must be 24 (2*4*3) lethality data lines.

Lethality data may be in one of eight forms. All data for a single weapon-asset combination must be in the same form; however, different forms may be used for different assets under a single weapon. The data form used for each asset is indicated by an integer code number on the asset name card.

The allowed forms are: a single ellipse, two or three concentric ellipses and the Carleton-von Neumann (bi-variate Gaussian) function. Each of the above can also be input with different elliptical or Gaussian parameters for the positive and negative RANGE (forward-backward) directions. Code numbers (2-10) for the various forms are listed in Table A-1.

TABLE A-1. Conventional Lethality Data Types and Required Data

CODE	DATA TYPE	REQUIRED DATA
2	Carleton-von N	Pk0,sigR,sigD
3	Single ellipse	Pk0,r0R,r0D
5	Double ellipse	Pk0,r0R,r0R,Pk1,r1R,r1D
6	Triple ellipse	Pk0,r0R,r0D,Pk1,r1R,r1D,Pk2,r2R,r2D
7	Asymmetric C-von N	Pk0,sigR+,sigD,sigR-
8	Asymmetric ellipse	Pk0,r0R+,r0D,r0R-
9	Asym. double ellipse	Pk0,r0R+,r0D,r0R-,Pk1,r1R+,r1D,r1R-
10	Asym. triple ellipse	Pk0,r0R+,r0D,r0R-,Pk1,r1R+,....

Note: Code numbers 1 and 4, originally used for data types that are no longer supported, are currently disabled.

Repairable Combat Damage Lethality Data

Data for the probability of repairable combat damage from conventional weapons is input using the same formats described in this section. There are, however, two restrictions on the values of the data input:

1. There must be exactly three kill criteria specified: damage requiring light repairs, damage requiring medium repairs and damage unfixable by the study unit.

2. The probabilities corresponding to light, medium and unfixable must be mutually inclusive; i.e. probability of light damage means "probability of AT LEAST" light damage. Therefore, for any given incoming round (HOB, posture of the target, miss-distance), the probability of light damage must be greater than or equal to the probability of medium damage. Similarly, probability of medium damage includes probability of unfixable.

Format for Conventional Lethality File (Unit #2)

```
weapon name
asset name1, data type code (per Table A-1)
number of HOBs (INTEGER), heights of burst(REAL)
number of postures (INTEGER), verbal descriptions
number of kill criteria (INTEGER), verbal descriptions
data
```

Example of Conventional Lethality Data File

```
WARHEAD1
SUPPLIES, 3
2, 0., 20.
1, IN THE OPEN
1, UNUSABLE
1.0, 19.2, 14.3
1.0, 22.7, 18.1
TRUCK, 8
1, 10.
2, IN THE OPEN, IN THE TREES
3, LITE, MEDIUM, HEAVY DAMAGE
1.0, 22.6, 14.4, 10.7, 0.3, 30.3, 22.3, 12.3
1.0, 18.4, 11.1, 5.9, 0.3, 24.6, 16.6, 8.8
1.0, 10.2, 8.3, 2.2, 0.3, 16.2, 12.2, 5.5
1.0, 18.8, 11.5, 6.6, 0.3, 25.0, 16.6, 9.0
1.0, 10.4, 8.5, 2.8, 0.3, 16.7, 12.6, 5.9
1.0, 6.2, 4.2, 0.9, 0.3, 10.4, 8.5, 2.3
WARHEAD2
SUPPLIES, 5
1, 8.
1, OPEN
1, RUINED
1.0, 19.3, 19.3, 0.3, 22.9, 22.9
TRUCK, 8
.......
.......
END
```

Nuclear Vulnerability Data for Equipment (Unit #3)

Unlike the case of conventional lethality, in which there is a distinct data entry for every combination of weapon, target, height-of-burst, posture and kill criteria, the shallow gradient of nuclear effects allows the insertion of an intermediate step: Weapons produce a set of environments and vulnerability of targets is assessed as a function of environmental strength. The environments included in the current Harry Diamond Laboratory (HDL) nuclear vulnerability data base, and hence calculated within AURA, are blast-overturn (dP*Iq), neutron fluence (TREE), electro-magnetic pulse (EMP) and thermal fluence.

Because all weapon events are converted to environments, only one card (data line) is required for each target. The card contains: the ASSET name (may be a common name) as it appears under ASSETS in the NAMES section of the AURA runstream, a code number which indicates the environments to which the item is susceptible, and the required data. The code number and the required data are taken directly from the HDL NUDACC data base.

The code number defined by HDL is found by adding together a digit assigned to each environment. These digits are:

1 - EMP
2 - TREE
4 - dPIq
8 - Thermal

Thus, the vulnerability data for an item susceptible to EMP and blast-overturn would have code number 5 (= 1 + 4). The data itself is a set of parameters for a shifted log-normal probability distribution.[A-1]. The data required for each environment is presented in the following table.

A-1. William L. Vault, "Vulnerability Data Array: The Agreed Data Base - Final Report (U)," Harry Diamond Laboratories, HDL-TR-1906, (JUL 80), (SECRET).

TABLE A-2. Data Required for Vulnerability Calculations by Environment

ENVIRONMENT	DATA REQUIRED
EMP	log-mean, log-sigma
TREE	transmission, log-mean, log-sigma
dPIq	threshold shift, log-mean, log-sigma
Thermal	damage fluence, dummy variable

Example of Nuclear Vulnerability File (Unit #3)

```
RADIO, 3, 1.6, 2.6, 1.0, 10.68, 3.2
TRUCK, 4, 2.2, 3.5, 2.56
END
```

Toxic Dispersion File (Unit #4)

The toxic dispersion file requires up to three sets of data derived from a dissemination code such as the NUSSE3 [A-2] code from the Chemical Research, Development and ENGINEERING Center (CRDEC). The number of data sets required depends upon the threat environments (contamination, percutaneous, vapor) to be included from each warhead. The first set of data, the contamination outline, gives the width, arrival and evaporation time of contamination as a function of downwind distance. The second set, also taken from the liquid phase of the dissemination calculation, is a grid of contamination density, normalized to a lethal percutaneous dose, for crosswind displacements as a function of downwind distance. The third set is a series of dosage grids, one for each selected time point, with each grid containing an entry for dosage at each crosswind displacement as a function of downwind distance. In the third set, all entries are normalized to a lethal inhalation dose.

It was foreseen that the amount of data needed to fully describe the behavior of a toxic threat was prohibitively ponderous to enter manually. Therefore, the BRL developed a utility code, called PRETOX, which interfaces with the NUSSE3 code to interactively generate the required data sets. Since the only standard outputs available from the NUSSE3 code are for hard-copy print files which are laced with a number of titles, spaces and other reader-friendly formats, the PRETOX code must strip off this information and extract the needed data. Early in the development of AURA, it was observed that a great deal of flexibility could be gained if all chemical dispersion data (depositions and dosages) were normalized before input. PRETOX also accomplishes this task.

To use PRETOX, the user responds to the following prompts.

WHAT TIME UNIT IS BEING USED IN AURA RUN (IN MINUTES)? .E.G. IF RUN IS IN MINUTES, TYPE 1. IF IN HOURS, TYPE 60.

NAME OF WEAPON FOR TOP OF FILE?

CONTAMINATION DATA (Y OR N)?

A-2. Richard Saucier, "A Mathematical Model for the Atmospheric Transport and Diffusion of a Chemical Contaminant," Chemical Systems Laboratory, ARCSL-TR-81071, (NOV 81).

If Y: LOWEST CONTAMINATION LEVEL TO BE CONSIDERED?

PERCUTANEOUS DATA (Y OR N)?

If Y: STANDARD LETHAL DEPOSITION (FOR NORMALIZING)?

PRIMARY VAPOR (Y OR N)?

TOTAL VAPOR (Y OR N)?

The user usually chooses the latter. If either is Y, the following prompts occur:

STANDARD LETHAL DOSE (FOR NORMALIZING)?

HEIGHTS READ: h1, h2, h3, ...
where h1, h2,.. are the heights at which NUSSE3 calculated dosages

WHICH ONE DO YOU WANT?

The program then writes the required data in AURA format onto output file (unit #4), reports to the user the number of data points included in each data set and exits.

On the output file, PRETOX also includes certain pieces of extra information such as the extrema of the dissemination, the agent type and the wind speed. This information is used in AURA to enhance certain outputs which provide quick summaries of the input deposition data.

Dose-Time Performance Degradation Data File (Unit #11)

The dose-time degradation matrices built into AURA should be sufficient for any routine application of AURA to the nuclear or chemical battlefield. However, special circumstances might arise in which the user will want to prescribe a dose-time dependent behavior onto certain job positions in the unit. Provisions have therefore been made to input additional dose-time performance degradation matrices and associate them to a degradation code number. Input of the data is made through input unit #11 when so directed by the SUBLETHAL DOSE DEGRADATION command. The user can then use the SUBLETHAL DOSE DEGRADATION command as described in the main body of this report to associate the degradation set to the desired LINKs.

Format for Dose-Time Degradation File (Unit #11)

```
18 character description, code number for the degradation
set
number of dose points (ND) and their values
number of time points (NT) and their values

Data for dose1  ( A set of NT degradation values )
Data for dose2  ( " )
.........
Data for dose ND
END
```

APPENDIX B

The Army Unit Resiliency Analysis (AURA) Methodology

Introduction

In the mid 1970s, the U.S. Army analysis community was faced with the problem of quantifying the expected survival of certain combat capabilities in the event of a European war. The study team that was formed to address the problem consisted of 10-20 different agencies, with specialities ranging from signature and acquisition to vulnerability to unit structure and manpower. It was soon apparent that a major hurdle in producing an over-all evaluation was the need to incorporate the widely different, highly detailed technical data into a coherent analysis. It was to fill that need that the large, integrated family of methodologies known as AURA was conceived.

AURA, the Army Unit Resiliency Analysis methodology, is a large, interconnected collection of analysis models which provides a detailed evaluation of the ability of an Army unit to accomplish a series of missions in a combat scenario. Briefly, AURA is an event sequenced, one-sided combat simulation methodology. The methodology consists of an (expanding) number of highly detailed models from the various technical communities interfaced into a large, time-dependent event playing and optimization routine. The interfaces are varied, involving such diverse kill probabilities as lethal footprints for conventional munitions, log normal kill probabilities for nuclear effects, toxic chemical dispersions and evaporations, MOPP degradation, reliability, and target acquisition probabilities. The optimization is a dedicated, non-linear routine which models the (commander's) reallocation of surviving, degraded assets in order to minimize the choke points in the optimal functional path. The logic process required the development of a general model for the functional structure of a military unit: Such a model was developed and forms an essential part of the AURA methodology.

The Radiation Engineering Branch, Vulnerability/Lethality Division, BRL, is the developer and a primary user of AURA. However, many other agencies now run the associated code, as do a number of contractors. The AURA code has been run on UNIVAC, VAX, IBM, CRAY and Data General computers, as well as the CDC series on which it resides at BRL.

An example of a recent expansion of the technical models incorporated in AURA occurred as part of a Defense Nuclear Agency (DNA) sponsored program. Under the DNA's Intermediate (Nuclear) Dose Program (IDP), the BRL extended the AURA family of methodologies to include the detailed models of radiation effects upon personnel performance and incapacitation. These models, based upon data from medical, historical, accident and animal-experimentation sources, were integrated into the main AURA code with the same degree of detail as were all the other factors which impact upon the resiliency of a combat unit on the integrated battlefield. As a result, the effects of IDP data are weighed equally with tactical, doctrinal, scenario-dependent, as well as other technical factors.

This summary paper describes the parts of the AURA methodology in varying detail, concentrating on the areas - such as the AURA Asset Allocation Algorithm (the Commander's decision model) - which were derived especially for AURA, and giving references to the standard methodologies which have been incorporated into the AURA family of methodologies.

Overview of AURA

As stated above, AURA is a family of methodologies covering a broad spectrum of technical areas. Figure B-1 is a depiction of several of the areas. As shown, the various models are interfaced together into a combat simulation through a (large) computer program which is also called AURA. Operation of the computer code involves the following stages:

A. The user inputs the runstream data, which includes:

   1. Scenario

   2. Threat

   3. Lethality

   4. Unit

      a. Mission(s)

      b. Assets

      c. Organization and Operation

B. The code processes the data and sets up the simulation.

C. The code runs the simulation several times.

   1. Lethality events cause damage, contamination, dosages, etc.

   2. Time dependent phenomena are updated

   3. Reconstitution events cause the commander to reallocate his surviving, degraded assets in order to optimize his unit's mission accomplishment

D. The code outputs total and averaged statistics on:

   1. Mission accomplishment

   2. Asset survival, degradations, dosages, etc.

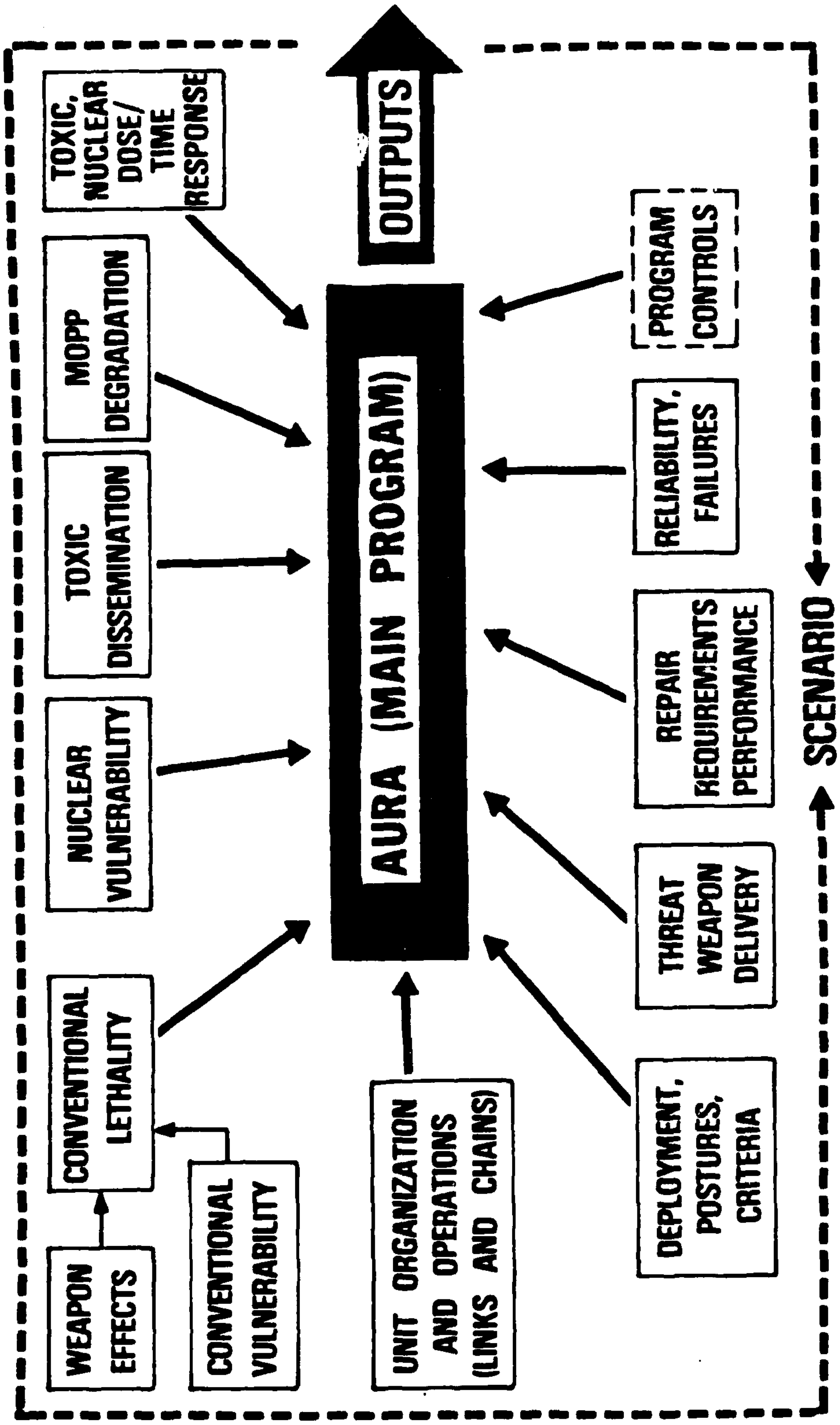

Figure B-1. The AURA Family of Methodologies

3. Reasons for shortcomings

4. Decisions made

5. Other items and actions, selected by the user

The process outlined above is depicted in Figure B-2.

Lethality Models

AURA uses the following standard weapon effect methodologies:

Conventional Lethality - JTCG/ME FULL SPRAY methodology[B-1]

Nuclear Vulnerability - NUDACC Methodology[B-2]

Toxic Dispersion - NUSSE3 Methodology[B-3]

Detailed descriptions of the above methodologies is beyond the scope of this paper. The interested reader is referred to the references cited.

In order to insure that weapons and weapon effects are faithfully reproduced in AURA, a great deal of effort was invested in developing the interfaces for the above weapon effects methodologies. For example, the unique characteristics of fragmenting munitions against targets requires that detailed "lethal footprints" be used for every weapon / target / height-of- burst / posture / kill-criterion. In the case of nuclear weapons, the data can be simplified by separating the effects into the two step process : weapon-produces-environment and environment-kills-target. However, five different environments (blast, thermal, neutron fluence, electromagnetic pulse, and total dosage) must be considered. Finally, toxic weapons result in dosages and contaminations which are highly time and posture dependent, as well as indirect effects such as causing a

---

B-1. "Computer Program for General Full Spray Materiel MAE Computations", Joint Technical Coordinating Group for Munitions Effectiveness, 61 JTCG/ME-79-1-1

B-2. William L. Vault,"Vulnerability Data Array: The Agreed Data Base - Final Report (U)", Harry Diamond Laboratories, HDL-TR-1906, (JUL 80). (SECRET)

B-3. Richard Saucier, "A Mathematical Model for the Atmospheric Transport and Diffusion of a Chemical Contaminant", Chemical Systems Laboratory, ARCSL-TR-81071 (NOV 81)

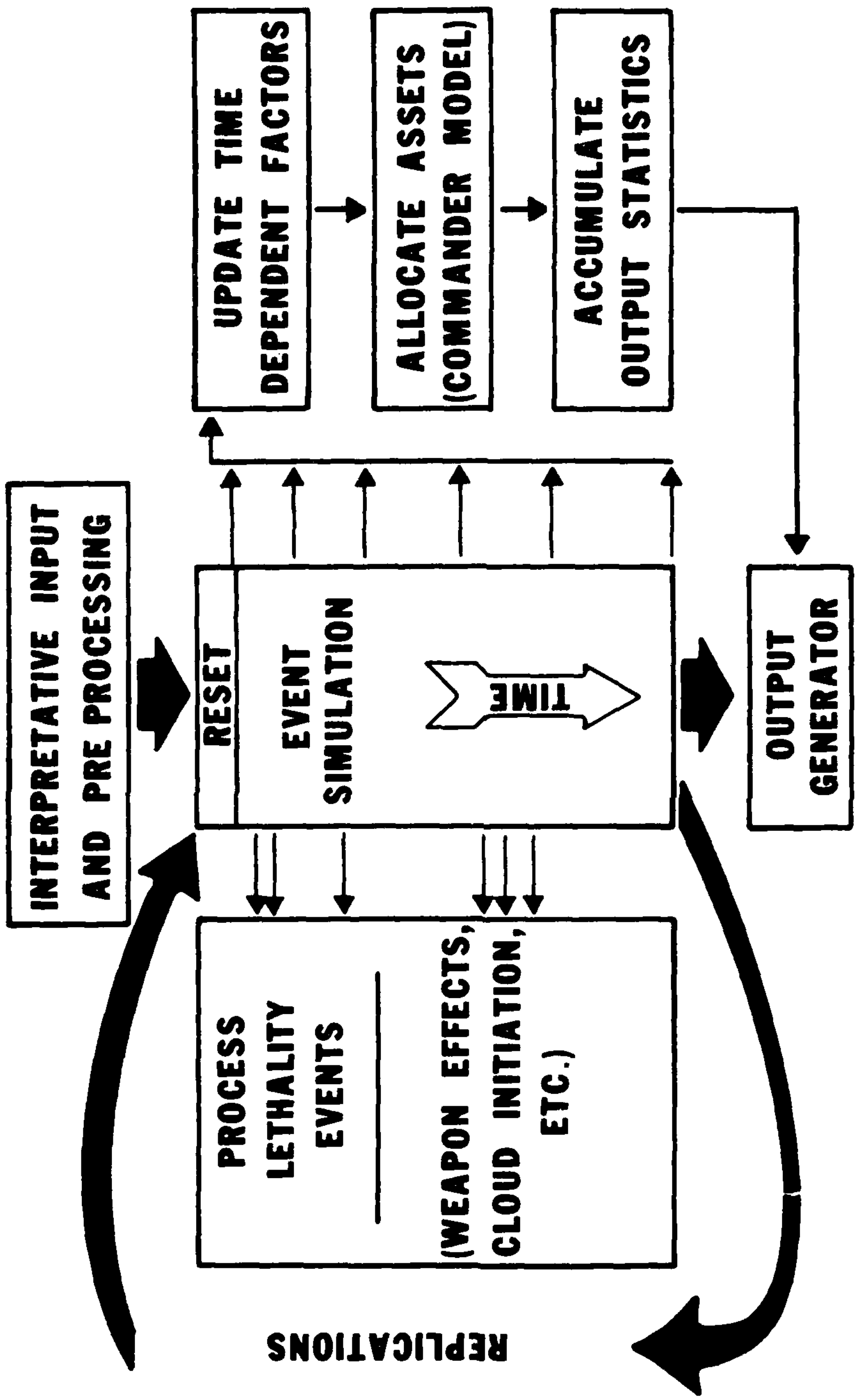

Figure B-2. AURA Code Flowchart

diversion of assets for decontamination activity and causing individual performance decrements due to the wearing of chemical protective clothing. The AURA family contains extensive off-line models, interfacing and utility programs and incorporated algorithms to assure that all of these factors are realistically modeled in every applicable AURA study.

The AURA Asset Allocation Algorithm

The formulation of the asset allocation model in AURA was guided by a number of essential factors:

1. Asset allocation is driven by mission accomplishment. Optimum allocation of surviving, possibly degraded assets is that allocation which results in the best accomplishment of the overall job or jobs that the unit must do. The asset allocation model is therefore inseparably tied to the unit functional description model.

2. Military units vary widely, especially in their functional descriptions. To be of general use, a functional description model must be flexible enough to describe many different possible relationships between assets. These relationships should be user-specifiable during input preparation.

3. To be easily usable, the elements of the functional model should have a recognizable association with actual elements. Similarly, although generally more difficult to describe, the relationships between model tasks should intuitively resemble the relationships between tasks actually done in the unit.

4. A somewhat subtle, but pervasive factor: the asset allocations model must be compatible with the mathematical behavior of the input data. Thus, for example, if assets are to be degraded (such as a man working at half the normal rate), then the asset allocation model cannot be intrinsically an integer model.

5. Of course, the output must be quantitative and must reflect the ability of the unit to perform the specified mission. The reason for any shortfall must be traceable (audit trail).

Guided more or less formally by these factors, we formulated the AURA Asset Allocation Algorithm (AAAA). The first step was to define a fundamental building block and introduce quantification. Following some work done for the Theater Nuclear Force Survivability Study by BDM (in a model called CCD), we chose the individual subtask, which we call a LINK, as the building block. Fundamental in CCD is the assumption that the ability to do a subtask depends upon the amount of assets allocated to that subtask.

From that point, we deviated from CCD. First, we generalized the functional relationship between the effectiveness of a subtask with respect to overall mission completion and the amount of assets allocated to the form shown in Figure B-3. To illustrate the flexibility of the form, note that the three graphs in Figure B-4 are just different cases of the same function, viz., a straight line from the left to the point (N_o, E_o), a slanted line up to the point (N_{100}, MAX), and a straight line off to the right. The curves in Figure B-4 describe different kinds of actual jobs:

TOP CURVE: There is an optimum amount of assets (number of men, e.g.), shown as N_{100}, at which the subtask effectiveness reaches its maximum. Allocating more assets doesn't add anything to the task, while taking away assets reduces it.

MIDDLE CURVE: Same as TOP CURVE, except this job requires a non-zero minimum number of assets (N_o) to begin to get any effectiveness in the task.

BOTTOM CURVE: Some "tasks" have a residual effectiveness, even with no assets assigned. For example, a well-trained crew can function without a supervisor. Thus, although the N_{100} amount of supervisory assets may be required for maximum performance, the effective supervisory function drops only to E_o at 0 assets.

The next step was to generalize the meaning of "amount of assets" to include degradations. Thus, putting a less than fully capable man into a subtask, or degrading the performance of a man (by putting him into protective clothing, by imposing radiation sickness, etc.) was equated to allocating less than a whole asset to the subtask. Since the effectiveness curves (Figure B-3) are continuous, this generalization required no change in the functional structure. (Note, however, that the use of continuous (non-integer) functions was made possible by the development of the non-integer (and non-linear) optimization algorithm described below.)

This mathematical model of a subtask, which we call a LINK, meets the requirements for a quantitative basic building block. It is easily associated with real jobs (driving a tank, receiving radio messages), and smoothly allows for non-integer assets. The parameters are also possible to evaluate. In evaluating an artillery battery, for example, one can ask: How many gunners are needed (=N_{100})? Can they do this mission (= MAX)? What happens if there are none (= E_o, N_o)?

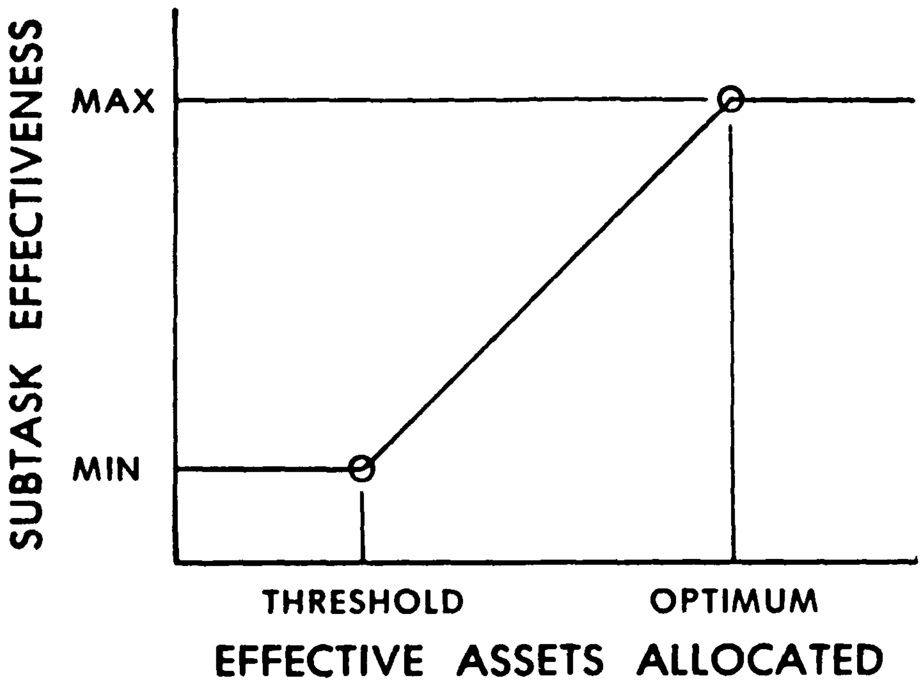

Figure B-3. General Form of a Link Effectiveness Curve

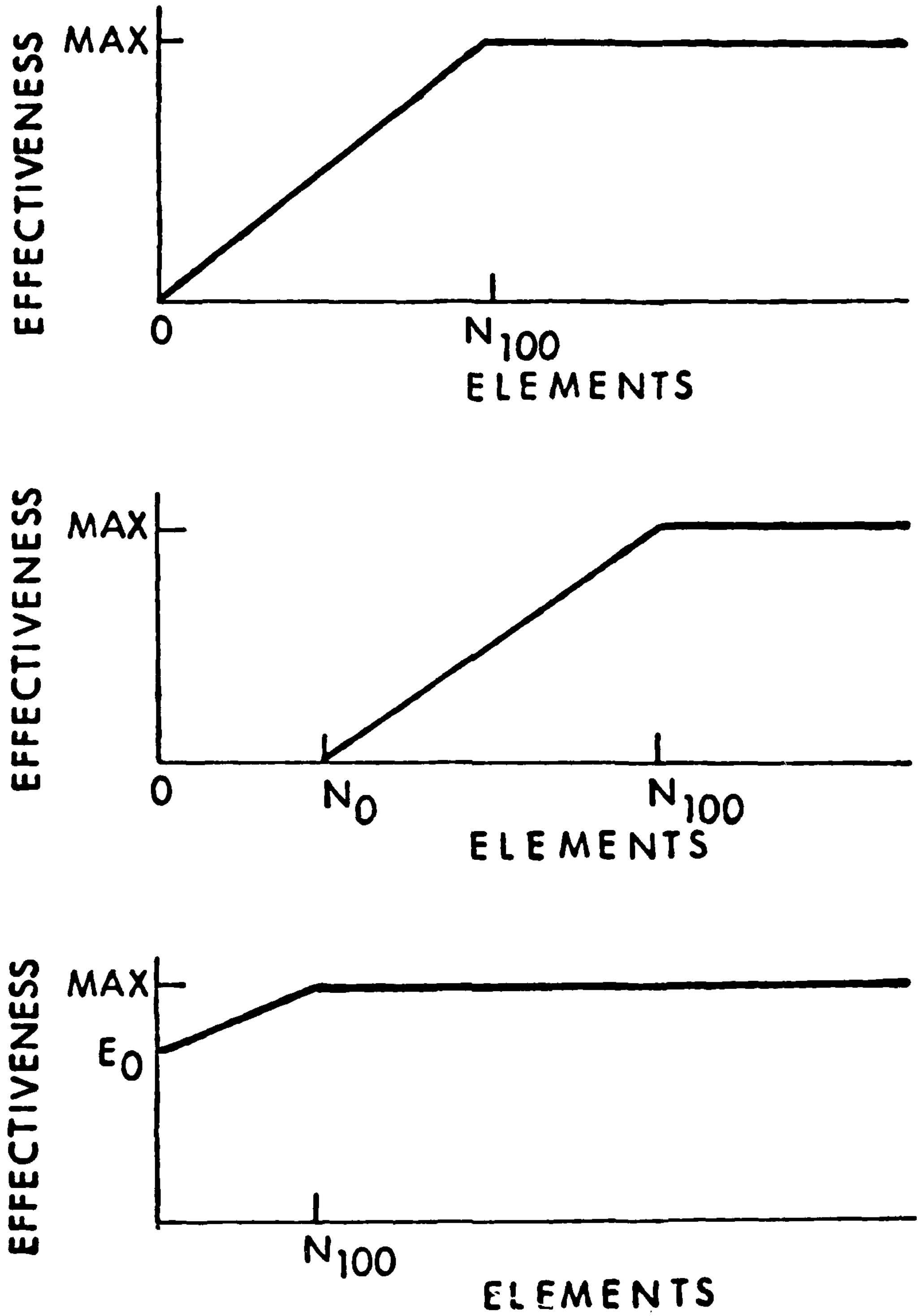

Figure B-4. Three Examples of Link Effectiveness Curves

While evaluating the LINK parameters, it is also convenient to ask: Who normally does this job? Who can substitute? How well (how much slower) does the substitute perform? How long does it take for the substitute to take over? This data forms the substitution matrix, which is another fundamental part of AAAA. Note that substitutes are governed by time to substitute and relative ability, as well as operational and physical degradations.

The next step in building AAAA was to account for the different relationships that subtasks (LINKs) could have to each other with respect to overall mission accomplishment. First, some tasks require others to also be effective in order to contribute to the overall mission. For example, the job that a forklift operator does is useless without the job that the forklift itself does, and visa-versa. Mathematically, this can be thought of as an AND relationship between the forklift and forklift operator LINKs. In AURA, LINKs having an AND relationship are said to form a SUBCHAIN. The user specifies the SUBCHAINs he wishes to form via his input runstream.

Another possible relationship between LINKs and/or SUBCHAINs is an exclusive OR: In such cases the user wants the code to choose the best of several alternate ways of accomplishing some function, where each choice (branch) may be composed of a single subtask (LINK) or a SUBCHAIN. For example, it may be possible to use the forklift team (SUBCHAIN composed of forklift and forklift operator) to load a truck, or to handload it manually (i.e., by using a crew of men.) In AURA, the specification of alternative procedures is done via the ORLINK construct.

(Note: The ORLINK is used for alternative *procedures*, that is, combinations of subtasks, which perform the same function. The use of alternative *personnel* or equipment to perform the *same* procedures is automatically done by the optimization algorithm for every subtask. In AURA, a great deal of care was taken to differentiate between subtasks (LINKs) and the people/equipment which perform those subtasks. The distinction between ORLINKs (choice of tasks) and substitutions (choice of performers) is just one manifestation of that differentiation.)

There are some jobs which involve a number of procedures that are additively related. For example, consider loading a truck with 75 percent light and 25 percent heavy items, which requires two different loading procedures. Clearly the relationship between the two procedures is not an OR (only one is chosen), nor is it an AND (no capability unless both are accomplished). Rather, the total fraction of the truck loaded is a weighted sum of the light and heavy loading capabilities, where the weighting factors, 0.75 and 0.25, reflect the relative demands. To model this relationship, AURA has a structure called COMPOUND LINKs (CPLINKs). CPLINK parts can be ORLINKs, SUBCHAINs, CREWs and/or simple LINKs.

The next higher level of aggregation is the CHAIN, a series of ANDs. The segments of a CHAIN can be CPLINKs, ORLINKs, SUB-CHAINs, CREWs and/or simple LINKs. This level of ANDs is convenient for assembling a complete mission, which requires command AND control AND communication AND transportation AND.........

Finally, a unit can be given a number of missions to consider at any given time. Each mission is described by a CHAIN. Each CHAIN has a series of time intervals during which the associated mission is to be done. If two time intervals overlap, implying two missions competing for the unit's attention during that time, AURA chooses the CHAIN which can be done most effectively. This, in effect, gives the user the capability of an overall OR relationship between CHAINs.

The hierarchy of relationships is depicted in Figure B-5 and summarized in Table B-1. Note that, in all cases, the constructs (CHAINs, ORLINKs, etc.) can be made of combinations of any of the lower echelon constructs. This ability to combine simple tasks, quantified in terms of effective assets, into a wide variety of complex structures gives AAAA the flexibility to be applied to a broad spectrum of unit types and missions.

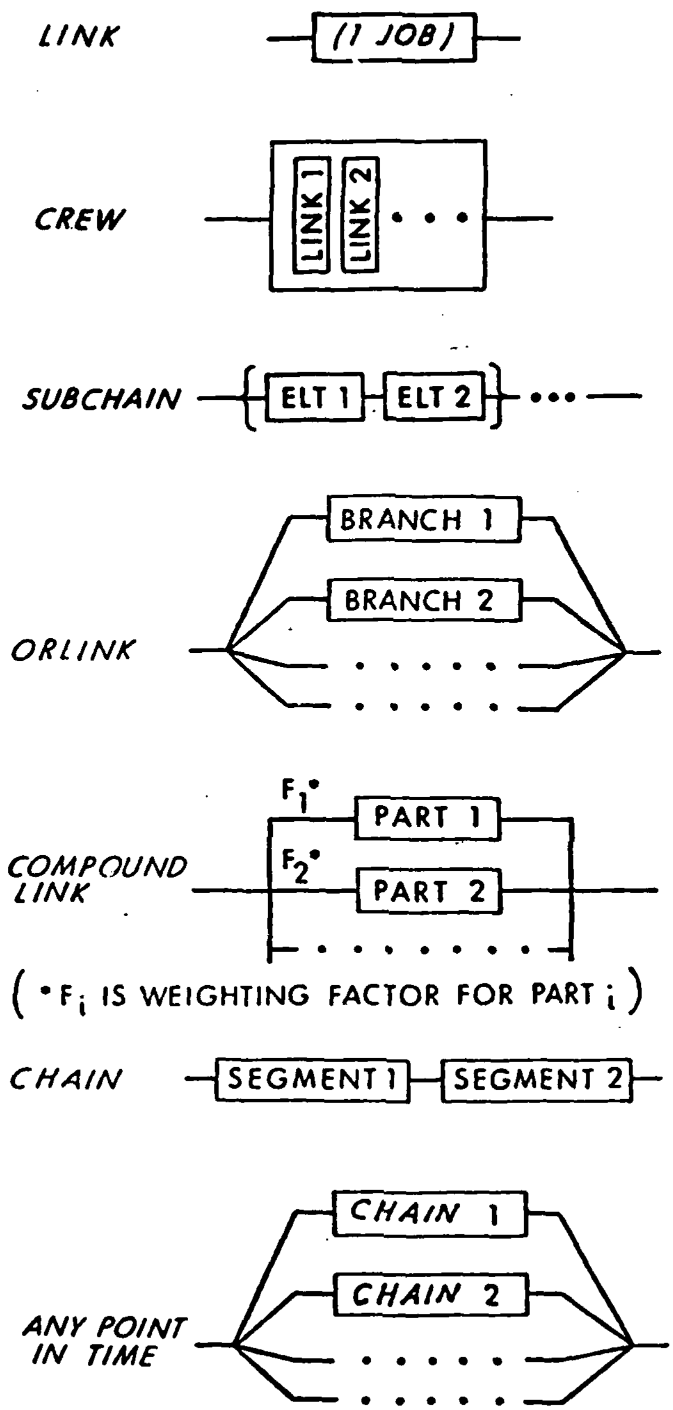

Figure B-5. Hierarchy of Relationships between Combinations of Jobs

TABLE B-1. HIERARCHY OF RELATIONAL OPERATORS

CONSTRUCT	OPERATOR	OPERANDS	Operand May Be
Point in Time	XOR	CHAINs	CHAIN
CHAIN	AND	Segments	COMPOUND LINK ORLINK SUBCHAIN CREW LINK
COMPOUND LINK	Weighted Sum	CP Parts	ORLINK SUBCHAIN CREW LINK
ORLINK	XOR	Branches	SUBCHAIN CREW LINK
SUBCHAIN	AND	Elements	CREW LINK
CREW	"Parallel"	Positions	LINK
LINK	Fundamental building block		

Application to an Example Unit

It is helpful to consider applying this structure to an example unit. Consider a small, hypothetical supply unit. The mission of the unit is to load trucks on order at a certain ratio. Two weight classes of items, heavy and light, are to be loaded: the heavy items, which comprise 25 percent of each load, must be loaded with a crane; the light items can be loaded by hand or by forklift. The order to fill the trucks is received by radio or telephone. Personnel are required to receive the order, man the forklift and crane teams, drive the truck, and handload if required. Handloading, however, can never accomplish more than 80 percent of the required rate, and requires more than one person.

There is also a loadmaster, who supervises the operation. However, the unit has functioned together long enough to work at 60 percent of the required rate even if the loadmaster's job is not done.

A set of LINK effectiveness curves to describe this unit is shown in Figure B-6. Note that most jobs in this simple example are of the (1., 100; 0., 0) form (1 asset for 100 percent effectiveness; 0 assets for 0 percent effectiveness). Exceptions to this are this are the LOADMASTER (1., 100; 0., 65) and the handloading MEN (5., 80; 1., 0).

For this example, only one mission was given. The CHAIN to accomplish that mission is shown in Figure B-7. As described above, the mission requires receipt of the message, a radio or telephone, 0.75 light and 0.25 heavy load capability and a truck. The LOADMASTER is also ANDed into the CHAIN. However, referring to Figure B-6, one notes that the absence of a loadmaster asset reduces the value of the LOADMASTER LINK only down to 0.6, thus limiting his effect on the unit.

Figure B-7 is a graphical depiction of a CHAIN. When augmented by a decision rule, the figure also depicts the value function upon which the AURA optimization process is based. That decision rule is:

> **The EFFECTIVENESS of a "CHAIN" is EQUAL to the EFFECTIVENESS of the WEAKEST SEGMENT.**

When applied to the example depicted in Figure B-7, this decision rule implies, e.g., that a unit which had only one-third of its trucks would only fulfill one-third of its mission, as long as all other capabilities were greater than one-third. In particular, even if the ability to receive messages was degraded to one-half the prescribed rate, the unit would only be able to load the available one-third prescribed number of trucks.

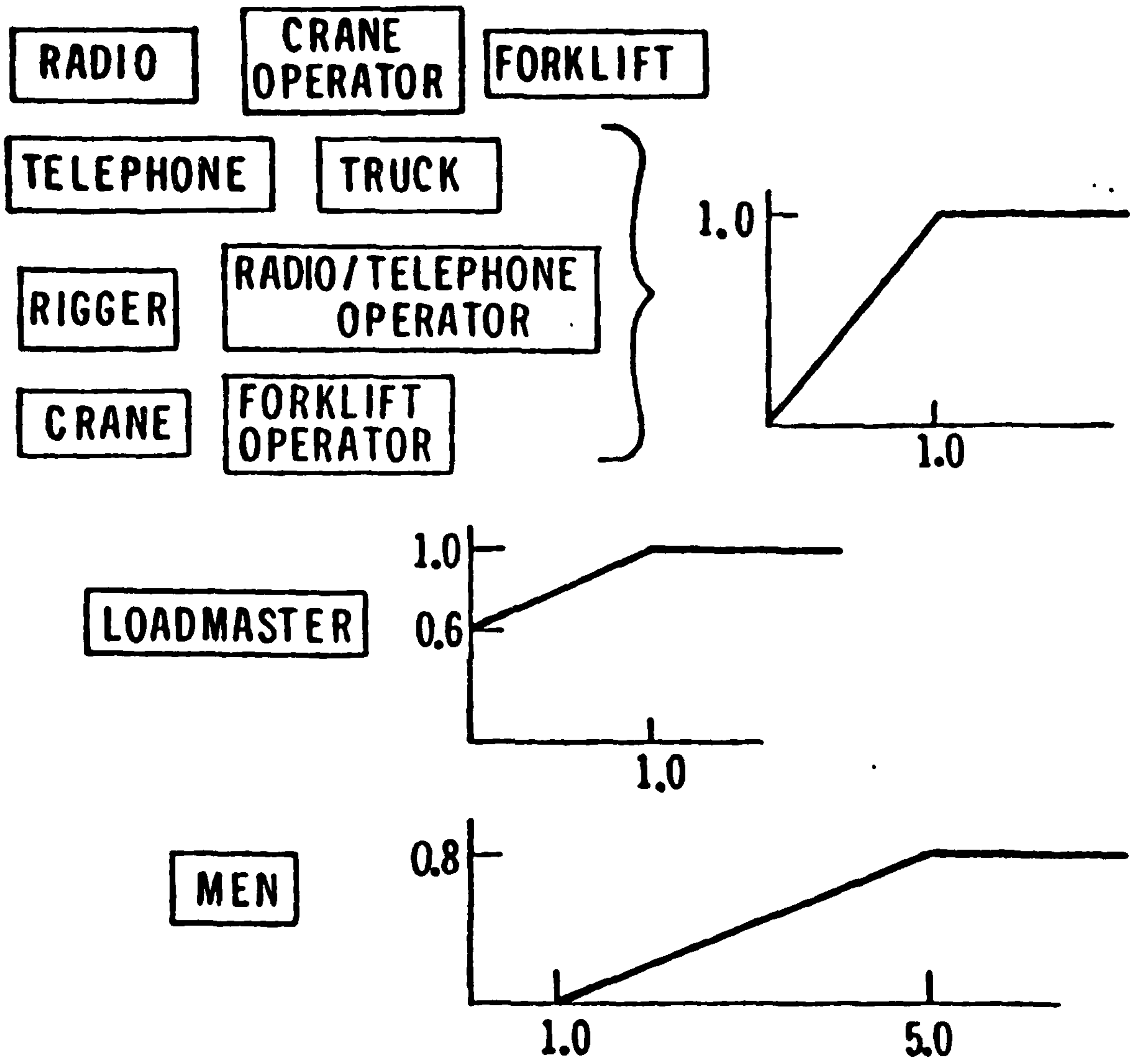

Figure B-6. Link Effectiveness Curves for the Example Unit

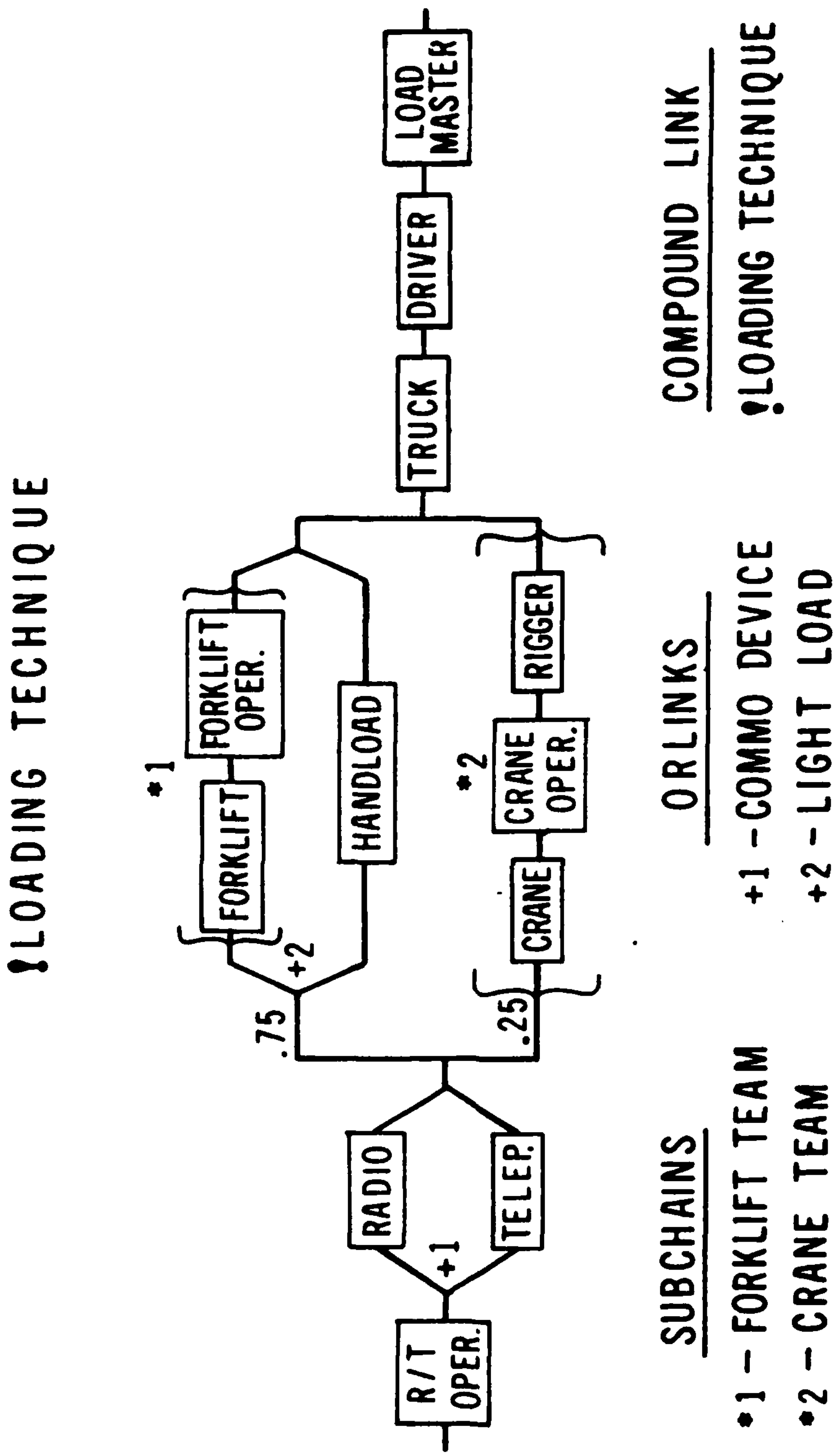

Figure B-7. CHAIN for the Example Unit

Mathematical Description of the AURA Asset Allocation Algorithm

The mathematical optimization algorithm used in AURA to allocate assets is a sizable extension of the GREEDY Algorithm. Basically, the GREEDY algorithm solves a maximization problem by a sequence of single steps. At each step, the algorithm selects that choice, consistent with the imposed constraints, which produces the maximum gain in the value function (the function whose value is to be maximized). In the case of AURA, the value function is the effectiveness of the unit, which is dictated by the value of the choke point: The GREEDY algorithm therefore indicates allocation of asset(s) to the weakest segment until its capability has improved above that of some other segment, which then becomes the weakest. This process continues until a point is reached at which no further allocation can be made. Intuitively, this process corresponds to a commander considering his mission's activity demands one at a time, in order of decreasing stringency, and allocating just enough assets to satisfy each demand before considering the next one. The GREEDY algorithm has been found to be a good model of human decision making in several important classes of problems[B-4].

Unfortunately, like greedy humans, the GREEDY algorithm guarantees the optimal solution for only a limited class of value functions and constraints[B-5]. In AURA, a unit whose members were completely cross-trained would fall into that class. However, in more common units, it is possible to "fool" the greedy commander, which mathematically corresponds to making a series of allocations which lead to a local maximum. The most likely way of doing this is to specify a unit structure and substitution effectiveness matrix in such a way that the algorithm chooses asset A over asset B early in the optimization process, then cannot fill a task that only A can do later in the sequence. To avoid this error, AURA incorporates three processes - preventative look-ahead, local dynamic look-ahead, and first-order look-back - which intuitively correspond to "experience", "limited foresight", and "limited error correction".

In AURA, preventative look-ahead is accomplished in two ways. First, a preprocessor evaluates the versatility (i.e. the number of possible job assignments) of each asset. In all subsequent allocation events, assets are considered in inverse order of versatility. In effect, the commander assigns his least

---

B-4. Eugene Lawler, "Combinatorial Optimisation: Networks and Matroids", Holt, Rinehart & Winston (1975))

B-5. Papadimitriou, C. and Steiglitz, K., "Combinatorial Optimization: Algorithms and Complexity", Prentice Hall (1982)) useful assets first, reserving his more-assignable ones for later. Secondly, before every allocation event, the algorithm counts the actual number of remaining assets which could possibly be assigned to each segment, then orders the segments for consideration in increasing order of potential assignees. This corresponds to the commander knowing that some jobs may be hard to fill and considering those jobs ahead of more redundant ones.

Local dynamic look-ahead is applied to improving segments which require a number of LINKs (job positions) to be filled before any gain is realized. In the truck loading example, the commander can load light parts by using a forklift and operator or by assigning several men to handload. When AURA considers the first alternative, the assignment of an operator is made tentatively; only when the forklift is found to be assignable and the gain from the two assignments is recognized is the set of assignments made "firm". Should the forklift not be available, the operator remains available for assignment to the handloading crew. This "look-ahead" is applied within every segment optimization step; however, it is not done from segment to segment.

Finally, first order look-back is applied as follows. If a point is reached at which a needed asset is unavailable, the algorithm checks all previous assignments of assets which could satisfy the need. If such a previous assignment is found and there is an unassigned asset available which could be substituted for the needed one, a "switch" is made: the unassigned asset takes the place of the needed one, and the needed one becomes available for reassignment to the current choke point. This look-back is limited to one level, however; C cannot replace B so that B can replace A so that A can become available.

(The above discussion did not address the REPAIR/DECONTAMINATION model incorporated into AURA. In AURA, the commander considers the possibility of improving unit performance through repair/decontamination activity. In order to effect repairs, the commander must allocate the personnel and equipment to repair/decon tasks IN ADDITION to the tasks required for his mission. After determining an optimum allocation of resources to perform this augmented mission, he weighs the immediate cost in mission performance versus the possible gain in deciding whether or not to include the repair/decon activity. Thus, the REPAIR/DECONTAMINATION model constitutes a fairly complex look-ahead process. However, unlike the other processes discussed above, inclusion of repair/decontamination alternatives is optional.)

The success of the algorithm in solving actual unit assignments has improved over the past six years as the "commander has become smarter" (i.e., as the above look-ahead and look-back features were added.) Mistakes by the algorithm in actual practice have become fairly rare, and are usually traceable to unlikely arrangements of skills (e.g. a senior person with a unique, non-essential capability and no capability to do the tasks of his juniors), in conjunction with a complex unit functional structure. As with any analysis tool, however, the final judgement lies in the hands of the analyst. For this reason, AURA also includes several output options which can be used to analyze results in detail, including the commander decisions which lead to the results.

Allocation Algorithm Decision Rules

The decision rules followed by the asset allocation algorithm (the "commander") in assigning assets to LINKs are as follows:

HOMELINK. A LINK is filled by its HOMELINK asset (an asset having the same name as the LINK) if one is available. If no HOMELINK asset is available, the commander will attempt to fill the LINK with a substitute. Also, if the available HOMELINK asset is degraded (e.g. because of sickness or fatigue) below a user-settable level (sicklv) and if there is a substitute available at a performance level more than (1/sicklv) greater than the best HOMELINK asset, then a substitute will be selected.

SUBSTITUTES. A potential substitute does not become available until the elapsed time (time since the need for a substitute developed) exceeds the substitute's (user-specified) substitution time. (See LINKS, section IV.E.2.) If more than one substitute is available in the elapsed time involved, a particular substitute is chosen by the following criteria.

1. Any potential substitute which is more than a user-settable level (signif) less effective than the best substitute is automatically dropped from consideration.

2. The commander will assign a less versatile asset in preference to assigning a more versatile asset. (Versatility, an integer number, is defined as the number of LINKs to which an asset can be assigned. AURA internally predetermines the versatility of each asset by analysis of the substitution matrix.)

3. The allocation algorithm numbers the substitutes for a particular LINK in the order in which they were named (see LINKS, section IV.E.2). The commander will assign an lower-numbered substitute in preference to a higher-numbered one. (Note, however, that several assets may have equal order numbers, since several substitutes may be specified by the same common name.)

The normal operation of the allocation algorithm is to take the decision criteria in the order presented above: a decision passes to the next criterion only if there is a "tie" in all preceding criteria.

The decision rules and values can be modified by the user. As stated above, the user can set the values of signif and sicklv. Furthermore, the order of consideration of criteria 2 and 3 (VERSATILITY and USER-INPUT-ORDER) can be reversed.

Finally, note that any decision made by the above rules can be over-ruled by a correction made through the look-back capability of the algorithm, as described in the preceding section.

Current Efforts in AURA Development

Current work at BRL and elsewhere is increasingly aimed at interfacing AURA outputs, especially for combat support and combat service support units, into broader scaled, 2-sided wargames. Such simulations generally concentrate upon the direct engagement of combat units. Thus, those elements involved in direct fire may be modeled individually, with parameters such as acquisition probability, first round hit/kill, etc. available for each firing element. However, those elements which function as a unit, such as artillery or ammunition resupply units, must generally be modeled only as "point targets" for indirect fire, whose residual capability to fire or supply is based upon simple linear attrition models. It has become increasingly clear that such units are far more complicated, and important, than can be shown by such simple models. A solution to the problem of getting the important details of combat support and combat service support units into 2-sided models lies in the ability of AURA to model such details in off-line runs and interface those details via look-up tables and semi-empirical curves. Recent progress suggests that such interfaces will succeed in both portraying the important intra-unit factors and providing affordably small requirements upon the 2-sided models.

REFERENCES

1. J. Terrence Klopcic, *Input Manual for the Army Unit Resiliency Analysis (AURA) Methodology*, BRL-TR-2670, (SEP 85). AD A159327.

2. Reference: J.T. Klopcic and L.K. Roach, *An Introduction to the Use of the Army Unit Resiliency Analysis (AURA) Methodology: Volume I,* BRL-MR-3384. (SEP 84).

A-1. William L. Vault, "Vulnerability Data Array: The Agreed Data Base - Final Report (U)," Harry Diamond Laboratories, HDL-TR-1906, (JUL 80), (SECRET).

A-2. Richard Saucier, "A Mathematical Model for the Atmospheric Transport and Diffusion of a Chemical Contaminant," Chemical Systems Laboratory, ARCSL-TR-81071, (NOV 81).

B-1. "Computer Program for General Full Spray Materiel MAE Computations", Joint Technical Coordinating Group for Munitions Effectiveness, 61 JTCG/ME-79-1-1

B-2. William L. Vault,"Vulnerability Data Array: The Agreed Data Base - Final Report (U)", Harry Diamond Laboratories, HDL-TR-1906, (JUL 80). (SECRET)

B-3. Richard Saucier, "A Mathematical Model for the Atmospheric Transport and Diffusion of a Chemical Contaminant", Chemical Systems Laboratory, ARCSL-TR-81071 (NOV 81)

B-4. Eugene Lawler, "Combinatorial Optimisation: Networks and Matroids", Holt, Rinehart & Winston (1975))

B-5. Papadimitriou, C. and Steiglitz, K., "Combinatorial Optimization: Algorithms and Complexity", Prentice Hall (1982))

DISTRIBUTION LIST

No. of Copies	Organization
10	C.I.A. OIR/DB/Standard GE-47 HQ Washington, DC 20505
12	Administrator Defense Technical Info Center ATTN: DTIC-FDAC Cameron Station, Bldg 5 Alexandria, VA 22304-6145
2	Director Defense Nuclear Agency ATTN: STBE (Dr. D. Auton, Dr. R. Young) Washington, DC 20305
2	Commander Defense Nuclear Agency/Field Command ATTN: FCPR (MAJ Fleming) FCPR (CPT Thorton) Kirtland AFB, NM 87115
1	HQDA DAMA-ART-M Washington, DC 20310
2	DA/ODCSOPS ATTN: DAMO-ZDF (Mr. Wolpert) DAMO-ZDF Pentagon, Washington, DC 20315
1	DA/ODCSOPS ATTN: DAMO-ZXG (Mr. Patterson) Washington, DC 20310-0404
1	Commander U.S. Army Materiel Command ATTN: AMCDRA-ST 5001 Eisenhower Avenue Alexandria, VA 22333-0001
1	Commander U.S. Army ARDEC ATTN: SMCAR-MSI Dover, NJ 07801-5001

No. of Copies	Organization
1	Commander U.S. Army ARDEC ATTN: SMCAR-TSS Dover, NJ 07801-5001
1	Commander U.S. Army ARDEC ATTN: SMCAR-TDC Dover, NJ 07801-5001
1	Commander U.S. Army ARDEC ATTN: SMCAR-RAR (Mr. Ostuni) Dover, NJ 07801-5001
1	Commander U.S. Army ARDEC ATTN: SMCAR-LC (Mr. Brooks) Dover, NJ 07801-5001
1	Director U.S. AMCCOM ARDEC CCAC Benet Weapons Laboratory ATTN: SMCAR-CCB-TL Watervliet, NY 12189-4050
1	Commander U.S. Army Armament, Munitions and Chemical Command ATTN: AMSMC-IMP-L Rock Island, IL 61299-7300
1	Commander U.S. Army Aviation Research and Development Command ATTN: AMSAV-ES 4300 Goodfellow Blvd St. Louis, MO 63120-1798
1	Director U.S. Army Air Aviation Research and Technology Activity Ames Research Center Moffett Field, CA 94035-1099

DISTRIBUTION LIST

No. of Copies	Organization
1	Commander U.S. Army Communications-Electronics Command ATTN: AMSEL-ED Fort Monmouth, NJ 07703-5301
1	Commander CECOM R&D Technical Library ATTN: AMSEL-IM-L (Reports Section) Fort Monmouth, NJ 07703-5000
5	Commander U.S. Army Harry Diamond Laboratories 2800 Powder Mill Road ATTN: DELHD-NW-RA Adelphi, MD 20783
1	Commander U.S. Army Missile Command Research, Development and Engineering Center ATTN: AMSMI-RD Redstone Arsenal, AL 35898-5241
1	Director U.S. Army Missile and Space Intelligence Center ATTN: AIAMS-YDL Redstone Arsenal, AL 35898-5500
1	Commander U.S. Army Tank Automotive Command ATTN: AMSTA-TSL Warren, MI 48090
8	Director U.S. Army TRADOC Analysis Center - WSMR ATTN: ATAA-SL (Mr. Pena) ATAA-SL (Mr. Kirby) ATAA-SL (Mr. Fernandez) ATAA-SL (Mr. Billingsley) ATAA-SL (Mr. Belk) ATAA-SL (Mr. Leach) ATAA-SL (Mrs. Hodde) ATAA-SL (Mr. Baldauf) White Sands Missile Range, NM 88002-5022

No. of Copies	Organization
1	Director U.S. Army TRADOC Analysis Center ATTN: ATOR-TSL White Sands Missile Range, NM 88002-5502
2	Director U.S. Army TRADOC Analysis Center - FLVN ATTN: ATRC-FTC (CPT Perry) Ft. Levenworth, KS 66027
1	Commander HQ TRADOC ATTN: ATCD-AS (Mr. Radda) Bldg 11 Ft. Monroe, VA 23651
2	Commander TRADOC Research Element TREM P.O. Box 8692 ATTN: MAJ R. Wimberley Monterey, CA 93943
2	Commander U.S. Army Aviation Logistics Center ATTN: ATSP-CD (CPT White) ATSP-CD (Mr. Pollack) Ft. Eustis, VA 23604
3	Commander U.S. Army Combined Arms Combat Developments Agency ATTN: ATZL-CAS-SP (CPT DeWulf) ATTN: ATZL-CAP (COL Henderson) ATTN: ATZL-CAP (CPT Grand) Ft. Leavenworth, KS 66027
4	U.S. Army Nuclear and Chemical Agency ATTN: MONA-WE 7500 Backlick Road, Bldg 2073 Springfield, VA 22150-3198
2	Commander U.S. Army Logistics Center ATTN: ATCL-OS (Mr. Cameron) ATCL-OS (CPT Buck) Ft. Lee, VA 23801

DISTRIBUTION LIST

No. of Copies	Organization
3	Commander Concepts Analysis Agency ATTN: CSCA (Mr. D. Hurd) CSCA (LTC Barrett) CSCA (Mr. Chandler) 8120 Woodmont Avenue Bethesda, MD 20014
1	Commandant U.S. Army Air Defense Artillery School ATTN: ATZK-CD Bldg. 5800 Ft. Bliss, TX 79916
1	Commandant U.S. Army Armor School Directorate of Combat Developments ATTN: ATZK-CP-SD (Mr. Hekemeyer) Ft. Knox, KY 40121
1	Commander U.S. Army Engineering School Ft. Belvoir, VA 22060
1	Commander U.S. Army Field Artillery School Directorate of Combat Developments Ft. Sill, OK 73503
3	Commandant U.S. Army Infantry School ATTN: ATSH-CD-CSO-OR (Mr. Derrico) Ft. Benning, GA 31905-5400
1	Commandant U.S. Army Transportation School Directorate of Combat Development Ft. Eustis, VA 23604
3	Commander U.S. Army Missile and Munitions Center and School ATTN: ATSK-CD-CS Redstone Arsenal, AL 35898

No. of Copies	Organization
1	Commander U.S. Army Quartermaster School ATTN: ATSM-CDC Ft. Lee, VA 23801
1	Commander U.S. Army Signal Center & School Directorate of Combat Developments Ft. Gordon, GA 30905
3	Commandant U.S. Army Chemical School Directorate of Combat Developments ATTN: ATZN-CM-CC (Mr. T. Collins) ATZN-CM-CC (LTC Gary Stratton) ATZN-CM-CC (Mr. B. Gould) Ft. McClellan, AL 36201
2	Commander U.S. Army Development and Employment Agency ATTN: MODE-ORD (Mr. Fabian) Ft. Lewis, WA 98433-5000
1	Commander Armed Forces Radiobiological Research Institute ATTN: Mr. William Jackson National Naval Medical Center Bethesda, MD 20014
2	Commander Combined Arms Center Scores Working Group ATTN: ATZL-CAD-LN Ft. Leavenworth, KS 66027
2	Commander Naval Surface Weapons Center Code 31-TJY (Mr. Yencha) Code 31-TJY (Mr. P. Kirk) Dahlgren, VA 22448
1	AFWL/SUL Kirtland AFB, NM 87117
1	Air Force Armament Laboratory ATTN: AFATL/DLODL (Tech Info Center) Eglin AFB, FL 32542-5438

DISTRIBUTION LIST

No. of Copies	Organization
1	McLean Research Center ATTN: Mr. W. Gray Parks 6845 Elm Street McLean, VA 22101
1	Office of the Secretary of Defense Joint Chemical Warfare Joint Test Force Suite 300, Five Skyline Place 5111 Leesburg Pike Falls Church, VA 22041
2	Institute for Defense Analysis ATTN: Mr. Graham McBryde, Ms. M. Charren 1801 N. Beauregard Street Alexandria, VA 22311
1	AFELM, The Rand Corporation ATTN: Library-D 1700 Main Street Santa Monica, CA 90406

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: AMXSY-D
AMXSY-MP (H. Cohen)
AMXSY-J (J. Blomquist)
AMXSY-GD (R. Mezan)
AMXSY (Mr. M. Miller)
AMXSY (Ms. C. Harley)
AMXSY (Ms. D. Smoot)
AMXSY-C (M. Carroll)
AMXSY-G (M. Sandmeyer)
Cdr, USATECOM
AMSTE-SI-F
Cdr, CRDEC, AMCCOM
SMCCR-RSP-A
SMCCR-MU
SMCCR-SPS-IL
SMCCR-MSS (T. Himmelheber/J. Seigh)
SMCCR-PPD (J. Baker)
SMCCR-RSP (L. Sturdivan)
Cdr, HEL
W. DeBellis
O. Zubal
Cdr, USAOC&S
ATSL-CD-CS

USER EVALUATION SHEET/CHANGE OF ADDRESS

This Laboratory undertakes a continuing effort to improve the quality of the reports it publishes. Your comments/answers to the items/questions below will aid us in our efforts.

1. BRL Report Number__________________________Date of Report____________

2. Date Report Received____

3. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which the report will be used.)____________

4. How specifically, is the report being used? (Information source, design data, procedure, source of ideas, etc.)____________________________________

5. Has the information in this report led to any quantitative savings as far as man-hours or dollars saved, operating costs avoided or efficiencies achieved, etc? If so, please elaborate.____

6. General Comments. What do you think should be changed to improve future reports? (Indicate changes to organization, technical content, format, etc.)

CURRENT ADDRESS

Name

Organization

Address

City, State, Zip

7. If indicating a Change of Address or Address Correction, please provide the New or Correct Address in Block 6 above and the Old or Incorrect address below.

OLD ADDRESS

Name

Organization

Address

City, State, Zip

(Remove this sheet, fold as indicated, staple or tape closed, and mail.)

FOLD HERE

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-5066

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 12062 WASHINGTON, DC

POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
U.S. Army Ballistic Research Laboratory
ATTN: SLCBR-DD-T
Aberdeen Proving Ground, MD 21005-9989

FOLD HERE

END
DATE
FILMED
9-88
DTIC