THE ALLOT COMPUTER PROGRAM

James A. Ross

April 1979

Prepared for
Office of the Joint Chiefs of Staff

INSTITUTE FOR DEFENSE ANALYSES
SYSTEMS EVALUATION DIVISION
Contract MDA903 79 C 0018
Task T-8-041

IDA Log No. HQ 79-21150
Copy 47 of 50 Copies

79 05 11 052

UNCLASSIFIED
The work reported in this document was conducted under contract MDA903 79 C 0018 OUSD(Eng) the Office of the Joint Chiefs of Staff. The publication of this IDA report does not indicate endorsement by the Office of the Joint Chiefs of Staff nor should the contents be construed as reflecting the official position of that agency.

Approved for public release; distribution unlimited.
THE ALLOT COMPUTER PROGRAM

JAMES A. ROSS

Institute for Defense Analyses
400 Army-Navy Drive
Arlington, Virginia 22202

Office of the Under Secretary of Defense for Research and Engineering
Washington, D.C. 20301

DoD-IDA Management Office
400 Army-Navy Drive
Arlington, Virginia 22202

Approved for Public release; distribution unlimited.

Target value relative force size (RFS)
nuclear target vulnerability (VN) system circular error probable (CEP)
equivalent megatonnage (EMT)
countermilitary potential (CMP)

ALLOT is a computer program that uses linear programming techniques to determine the optimum laydown of a nuclear arsenal against a target base. The arsenal (on input to the model) consists of a number of weapon classes, each characterized by yield, burst height, CEP, prelaunch survivability, weapon system reliability, and penetration probability. The target base (also an input) consists of a number of target classes, each characterized by a target value, a target vulnerability number, and a target base.
ACKNOWLEDGEMENT

The author wishes to express his appreciation for the invaluable assistance provided by LtCol. Donald J. Berg, USAF, JCS/SAGA.
CONTENTS

I. Introduction ........................................... 1
II. The Relative Force Size ............................... 5
III. Operation ............................................... 9
IV. Input Requirements ..................................... 11
V. Sample Cases ............................................ 21

APPENDICES

A. Mathematical Theory
B. ALLOT Linear Programming Algorithm
C. Listing of ALLOT Program
Chapter I
INTRODUCTION

ALLOT is a computer program that uses linear programming techniques to determine the optimum laydown of a nuclear arsenal against a target base. It was written by LtCol. Donald J. Berg, USAF, JCS/SAGA, and modified by Dr. James A. Ross of the Institute for Defense Analyses (IDA). It has been validated against the SIOP. A brief description of ALLOT is presented below.

The nuclear arsenal (input) comprises a number of different weapon classes, each of which is characterized by yield, height of burst, circular error probable, prelaunch survivability, weapon system reliability, and penetration probability. All weapons within a given class are considered to be identical.

Similarly, the target base (also input) consists of a number of different target classes, each characterized by a value, a VN number,¹ and a target radius. In addition, the user may, if desired, input minimum damage goals to be achieved against individual target classes, or groups of classes.

Once all the inputs have been assembled, ALLOT allocates the nuclear arsenal against the target base to produce an optimum laydown according to whichever of the following specifications is chosen by the user.

NOTE: The views expressed herein are those of the author only. Publication of this Note does not indicate endorsement by IDA or the Joint Chiefs of Staff.

¹Those readers not familiar with the nuclear target vulnerability (VN) system used by the Defense Intelligence Agency (DIA) are referred to the Physical Vulnerability Handbook—Nuclear Weapons, DIA publication AP-550-1-2-69-INT, published 1 June 1969, with changes dated through 1 June 1976.
The total target value destroyed is to be a maximum.

The total number of weapons employed is to be a minimum.

The total megatonnage employed is to be a minimum.

The total equivalent megatonnage (EMT) employed is to be a minimum. The EMT of a weapon is defined as follows. Let $Y$ be the yield of the weapon, in megatons. Then the EMT equals $Y^{1/2}$ if $Y$ is greater than 1, or $Y^{2/3}$ if $Y$ is less than 1.

The total countermilitary potential (CMP) employed is to be a minimum. For the purposes of the ALLOT model, the CMP of a weapon is defined as follows. Let $Y$ be the yield of the weapon, in megatons, and let $CEP$ be the circular error probable, in feet. Define $C = 10^8 \cdot Y^{2/3}/CEP^2$ if $Y$ is greater than or equal to 0.2, or $C = 10^6 \cdot Y^{a/5}/CEP^2$ if $Y$ is less than 0.2. Then the CMP equals $C$ unless $C$ is greater than 3, in which case the CMP is simply set equal to 3.

The Relative Force Size (RFS) is to be computed. RFS is defined and discussed in Chapter II.

The chief ALLOT modification performed by IDA is the inclusion of the capability to compute RFS. This measure has gained widespread interest recently, and has been used both in the Secretary of Defense Annual Report and the Consolidated Guidance. ALLOT formerly had no capability to compute RFS; hence, its inclusion is considered to be a valuable addition. Other modifications performed by IDA, aside from auxiliary programming changes required to facilitate the RFS computation within the existing ALLOT framework, include coding changes designed to accommodate greater numbers of weapon classes and target classes than was previously possible.

The version of ALLOT currently being used on IDA's CDC 6400 computer allows up to 20 weapon classes and up to 20 target classes. (The version in JCS/SAGA allows considerably more.) Computer running time depends to a large extent on $N$, the maximum number of weapons allowed to be assigned to a target. If $NW$ is the number of weapon classes, and $NT$ is the
number of target classes, then the running time \( t \) on IDA's computer is given approximately by

\[
t = 0.001 (NW)^n (NT)(NW+NT/n)!
\]

where \( t \) is expressed in minutes. Some examples are given below.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( NW )</th>
<th>( NT )</th>
<th>( t )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>&lt;0.5</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>15</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>20</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>15</td>
<td>28</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td>15</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>10</td>
<td>33</td>
</tr>
</tbody>
</table>
Chapter II
THE RELATIVE FORCE SIZE MEASURE

Relative Force Size (RFS) is defined as the ratio of two force levels:

\[
RFS = \frac{\text{forces available}}{\text{forces required to meet specified damage goals}}
\]

In other words, the forces required equal the forces available divided by RFS. To put it another way, if all the forces in the weapon arsenal were scaled down proportionally by a factor RFS, and if these scaled forces were then all optimally allocated against the target base, then the specified damage goals would just be met.

For example, if the target base consisted of 50 identical targets, and if the weapon arsenal consisted of 50 identical weapons, each with a single-shot probability of kill (SSPK) of 0.50 against a target, and if the damage goal were to destroy 50 percent of the total target base, then the RFS would be 1.00 (see Table 1). On the other hand, if the goal were 25 percent, the RFS would be 2.00; and if the goal were 75 percent, the RFS would be 0.50.

\[1\text{For the purposes of this paper, SSPK is defined as (probability of arrival) x (probability of kill given an arrival), where probability of arrival is defined as (prelaunch survivability) x (weapon system reliability) x (penetration probability). All warheads are assumed to be independent, and no fratricide is allowed. The effect of an enemy attack absorbed before the force is launched may be reflected in prelaunch survivabilities, but post-attack retargeting is not permitted.}\]
Table 1. RFS CALCULATIONS BASED ON ONE WEAPON TYPE AND ONE TARGET TYPE

<table>
<thead>
<tr>
<th>Weapons Available</th>
<th>Targets</th>
<th>SSPK</th>
<th>Overall Goal</th>
<th>Weapons Required</th>
<th>RFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>50</td>
<td>0.50</td>
<td>50%</td>
<td>50a</td>
<td>1.00</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>0.50</td>
<td>25%</td>
<td>25b</td>
<td>2.00</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>0.50</td>
<td>75%</td>
<td>100c</td>
<td>0.50</td>
</tr>
</tbody>
</table>

*One weapon on every target.
*bOne weapon on every other target.
*cTwo weapons on every target.

In general, of course, there are more target types and more weapon types. In the examples in Table 2, there are two target types (each with 50 targets) and two weapon types (each with 50 weapons). The SSPKs for weapon type 1 against target types 1 and 2 are assumed to be 0.10 and 0.90 respectively, and those for weapon type 2 are assumed to be 0.50 and 0.50.

Table 2. RFS CALCULATIONS BASED ON TWO WEAPON TYPES AND TWO TARGET TYPES

<table>
<thead>
<tr>
<th>Weapons Available</th>
<th>Targets</th>
<th>SSPKs</th>
<th>Overall Goals</th>
<th>Weapons Required</th>
<th>RFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1</td>
<td>Type 2</td>
<td>Type 1</td>
<td>Type 2</td>
<td>Wpn 1/Tgt 1</td>
<td>Wpn 1/Tgt 2</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>0.10</td>
<td>0.90</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>0.10</td>
<td>0.90</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>0.10</td>
<td>0.90</td>
</tr>
</tbody>
</table>

*One weapon on every type 2 target.
*bOne weapon on every type 1 target.
*cOne weapon on every other type 2 target.
*dOne weapon on every other type 1 target.
*eTwo weapons on every type 2 target.
*fTwo weapons on every type 1 target.
As the numbers of target types and weapon types continue to increase, it becomes more and more difficult to evaluate the RFS by simple inspection. The general problem is amenable to solution by linear programming techniques, however, provided a bound is placed on the maximum number of weapons that can be assigned to any one target. In ALLOT this upper bound is three. For problems of practical interest, this limitation has not been a serious restriction.
Chapter III
OPERATION

The types of problems solved by ALLOT can always be expressed in the following mathematical form: Find the values of the variables $x_1, x_2, x_3, \ldots, x_{m-1}, x_m; y_1, y_2, y_3, \ldots, y_{n-1}, y_n$, such that, for given $a_{ij}$ and $b_i$,

$$
\sum_{j=1}^{n} a_{ij} y_j + x_i = b_i \quad (i = 1, 2, \ldots, m),
$$

where $x_i \geq 0$ for all $i$, $y_j \geq 0$ for all $j$, and $x_1$ is a minimum.

Thus, operation of the ALLOT model in essence consists of (1) reading the inputs, (2) making preliminary calculations, (3) determining the values of the $a_{ij}$ and the $b_i$, (4) solving the basic equations, and (5) printing the results. The subroutines in which these steps are actually carried out are indicated below.

**OPERATION**

1. Read Input Values
2. Compute SSPKs
3. Print Input Values
4. Compute Values of $b_i$
5. Compute Values of $a_{ij}$

6.* Print Values of $a_{ij}$ and $b_i$
7. Solve Equations by Using Linear Programming Algorithm
8.* Print Formal Linear Programming Solution
9. Print Results

*Optional
**Also uses auxiliary subroutines AX and AMAT.
Further discussion concerning the mathematics involved—including the physical significance of the $x_1$, $y_j$, $a_{ij}$, and $b_i$—is presented in Appendices A and B. A complete listing of ALLOT is presented in Appendix C.
Chapter IV
INPUT REQUIREMENTS

The input requirements for ALLOT are described below.

Control Cards - Control cards are used to signal to the input routine the type of information that follows. In some cases the information is fully contained on the control card itself, and in other cases the control card only denotes the type of information on the cards that follow it. The control cards (discussed below) are:

*IN/OUT — select input/output options
*START — start problem
*WPN — input weapon specifications
*TGT — input target specifications
*NWP — input weapon numbers
*NTG — input target numbers
*CWP — change and/or add weapons
*CTG — change and/or add targets
*ADD — input weapon vectors to be added (see discussion below)
*SUB — input weapon vectors to be deleted (see discussion below)
*FOR — input applicable targets for weapon vectors (see discussion below)
*FTP — input footprinting degradation factor
*COL — include collateral damage option
*END — end of problem

*IN/OUT Card - The *IN/OUT card is used to select input/output options. Its use is optional. When it is used, it must appear directly before any *START card.

CARD FORMAT

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-7</td>
<td>*IN/OUT</td>
<td>--</td>
</tr>
<tr>
<td>10-12</td>
<td>r_1</td>
<td>A3</td>
</tr>
<tr>
<td>14-16</td>
<td>r_2</td>
<td>A3</td>
</tr>
</tbody>
</table>
The possible values for the $f_1$ are:

MAT - print out intermediate results (generally used for debugging purposes only)

NOL - suppress laydown printout

*START Card - The *START card must appear as the first card of every problem. In addition to signaling the beginning of a problem, the *START card is also used to select an upper limit on the number of weapons assigned to a target, and to select the type of laydown desired.

**CARD FORMAT**

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-6</td>
<td>*START</td>
<td>--</td>
</tr>
<tr>
<td>8</td>
<td>N</td>
<td>A1</td>
</tr>
<tr>
<td>10-12</td>
<td>F</td>
<td>A3</td>
</tr>
</tbody>
</table>

N is the upper weapon limit per target. It must not be greater than three. F is used to select the type of laydown desired. The possible "values" for F are:

DE = maximize expected value damage  
WPN = minimize weapons used  
MT = minimize megatons used  
EMT = minimize EMT used  
CMP = minimize CMP used

If F is blank, the program will compute RFS.

*WPN Card - The *WPN card is used to signal that the cards that follow it are weapon specification cards. If this card is not used, then the arsenal from the previous problem is used.

**CARD FORMAT**

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>*WPN</td>
<td>--</td>
</tr>
</tbody>
</table>
**UNCLASSIFIED**

**Weapon Specification Card** - Weapon specification cards specify the values of the parameters associated with each weapon type.

### CARD FORMAT

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>DEFAULT VALUE</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-12</td>
<td>name</td>
<td>blank</td>
<td>2A4</td>
</tr>
<tr>
<td>15-18</td>
<td>type</td>
<td>blank</td>
<td>A4</td>
</tr>
<tr>
<td>19-24</td>
<td>number</td>
<td>none</td>
<td>16</td>
</tr>
<tr>
<td>29</td>
<td>height of burst (gnd=0, opt air=1)</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>31-40</td>
<td>yield (in megatons)</td>
<td>none</td>
<td>F10.0</td>
</tr>
<tr>
<td>41-50</td>
<td>circular error probable (in feet)</td>
<td>none</td>
<td>F10.0</td>
</tr>
<tr>
<td>51-60</td>
<td>prelaunch survivability</td>
<td>1</td>
<td>F10.0</td>
</tr>
<tr>
<td>61-70</td>
<td>weapon system reliability</td>
<td>1</td>
<td>F10.0</td>
</tr>
<tr>
<td>71-80</td>
<td>probability to penetrate</td>
<td>1</td>
<td>F10.0</td>
</tr>
</tbody>
</table>

**#TGT Card** - The #TGT card is used to signal that the cards that follow it are target specification cards. If this card is not used, then the target system from the previous problem is used.

### CARD FORMAT

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>#TGT</td>
<td>--</td>
</tr>
</tbody>
</table>

**Target Specification Card** - Target specification cards specify the values of the parameters associated with each target type.
**CARD FORMAT**

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>DEFAULT VALUE</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-12</td>
<td>name</td>
<td>blank</td>
<td>2A4</td>
</tr>
<tr>
<td>15-18</td>
<td>type</td>
<td>blank</td>
<td>A4</td>
</tr>
<tr>
<td>19-24</td>
<td>number</td>
<td>none</td>
<td>I6</td>
</tr>
<tr>
<td>26-29</td>
<td>VN number</td>
<td>none</td>
<td>I2,A1,I1</td>
</tr>
<tr>
<td>31-40</td>
<td>value</td>
<td>1</td>
<td>F10.0</td>
</tr>
<tr>
<td>41-50</td>
<td>DEMIN—minimum</td>
<td>0</td>
<td>F10.0</td>
</tr>
<tr>
<td></td>
<td>DE allowed on</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>each target of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>that type.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>51-60</td>
<td>DEMAX—maximum</td>
<td>1</td>
<td>F10.0</td>
</tr>
<tr>
<td></td>
<td>DE allowed on</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>each target of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>that type.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>61-70</td>
<td>DEAVE—minimum</td>
<td>0</td>
<td>F10.0</td>
</tr>
<tr>
<td></td>
<td>average DE to be</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>achieved on that</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>target type.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>71-80</td>
<td>DEDLT—target</td>
<td>0</td>
<td>F10.0</td>
</tr>
<tr>
<td></td>
<td>radius (nautical</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>miles)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*NWP Card* - The *NWP card is used to change the number of each name of weapon used in the arsenal from the previous problem. If the previous problem had NW weapon names, the NW numbers in 2014 format must immediately follow the *NWP card.*

**CARD FORMAT**

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td><em>NWP</em></td>
<td>--</td>
</tr>
</tbody>
</table>

This is followed by a card with NW items in 2014 format.

---

1A series of n consecutive target specification cards with negative values of DEAVE, followed by a card with a positive DEAVE value, D, indicates that a fraction D of the total value of all the targets in the set of n+1 target specification cards must be destroyed.

14
UNCLASSIFIED

*NTG Card - The *NTG card is used to change the number of each of the NT target names in the same manner the *NWP card is used to change the number of each of the NW weapon names.

CARD FORMAT

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>*NTG</td>
<td>--</td>
</tr>
</tbody>
</table>

This is followed by a card with NT items in 2014 format.

*CWP Card - The *CWP card is used to change a weapon specification or add a new one to the arsenal of the previous problem, without having to input the entire arsenal.

CARD FORMAT

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>*CWP</td>
<td>--</td>
</tr>
</tbody>
</table>

The *CWP card is immediately followed by weapon specification cards. If the name and type field on a card (other than blanks) agree with the name and type field for a weapon type in the previous arsenal, then the old parameters are replaced by the new ones. If the name and type fields do not agree with any in the previous arsenal, then the weapon type is added to the arsenal. When a weapon is added, its type field must either agree with the type field of the last weapon in the previous arsenal or else it must be an entirely new type.

*CTG Card - The *CTG card is used to change or add targets to the previous target system in the same manner the *CWP card changes or adds weapons.

CARD FORMAT

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>*CTG</td>
<td>--</td>
</tr>
</tbody>
</table>
**ADD Card** - The *ADD card signals that the cards that follow it define weapon vectors that are to be added to the allowed set of weapon vectors.

**CARD FORMAT**

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>*ADD</td>
<td>--</td>
</tr>
</tbody>
</table>

**SUB Card** - The *SUB card signals that the cards that follow it define weapon vectors that are to be deleted from the set of allowed weapon vectors.

**CARD FORMAT**

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>*SUB</td>
<td>--</td>
</tr>
</tbody>
</table>

**ADD/SUB Specification Card** - These cards define the weapon vectors to be added to or deleted from the set of allowed weapon vectors.

**CARD FORMAT**

<table>
<thead>
<tr>
<th>COLS</th>
<th>PAR</th>
<th>FORMAT</th>
<th>COLS</th>
<th>PAR</th>
<th>FORMAT</th>
<th>COLS</th>
<th>PAR</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>r₁</td>
<td>A₁</td>
<td>30</td>
<td>r₃</td>
<td>A₁</td>
<td>54</td>
<td>r₅</td>
<td>A₁</td>
</tr>
<tr>
<td>7</td>
<td>n₁</td>
<td>II</td>
<td>31</td>
<td>n₃</td>
<td>II</td>
<td>55</td>
<td>n₅</td>
<td>II</td>
</tr>
<tr>
<td>9-16</td>
<td>t₁</td>
<td>2A₄</td>
<td>33-40</td>
<td>t₃</td>
<td>2A₄</td>
<td>57-64</td>
<td>t₅</td>
<td>2A₄</td>
</tr>
<tr>
<td>18</td>
<td>r₂</td>
<td>A₁</td>
<td>42</td>
<td>r₄</td>
<td>A₁</td>
<td>66</td>
<td>r₆</td>
<td>A₁</td>
</tr>
<tr>
<td>19</td>
<td>n₂</td>
<td>II</td>
<td>43</td>
<td>n₄</td>
<td>II</td>
<td>67</td>
<td>n₆</td>
<td>II</td>
</tr>
<tr>
<td>21-28</td>
<td>t₂</td>
<td>2A₄</td>
<td>45-52</td>
<td>t₄</td>
<td>2A₄</td>
<td>69-76</td>
<td>t₆</td>
<td>2A₄</td>
</tr>
</tbody>
</table>

The r₁'s are only used in the *SUB section and are relational operators, where G signifies greater than, < signifies less than, and any other character or a blank signifies equality.

Weapon vectors are, by definition, weapon combinations allowed to be assigned to a target. The *START card indicates that all combinations with N or less weapons are allowed. The purpose of the *ADD and *SUB cards is to permit additions and/or subtractions to the group of allowed weapon vectors.
The $n_i$'s are numbers of weapons. The $t_i$'s are either the name of a weapon (name field), the type of weapon (type field), or the word OTHERS left justified in the field. Anything following a blank, an invalid $t_i$ field, or the word OTHERS is ignored. A $t_6$ field may be used only for OTHERS.

Examples of use:

*ADD

1 ICBM 1 B52 SRAM
2 SLBM 1 OTHERS

*SUB 1 ICBM 0 TRIDENT
1 POLARIS 1 B52 SRAM 0 OTHERS
0 POSEIDON 1 OTHERS

The OTHERS clause is handled somewhat differently in the *ADD and *SUB sections. In the *ADD section, if the OTHERS clause is not present, then zero OTHERS is assumed. On the other hand, if the OTHERS clause is not present in the *SUB section, it is interpreted as "don't care about" OTHERS. Thus, in the first example of the *SUB section above, all weapon vectors with one ICBM and more than zero TRIDENTs will be deleted, regardless of how many other weapons may be in the weapon vector. In the second *SUB section example, however, only the weapon vector with one POLARIS, one B52 SRAM, and no other weapons will be deleted.

*FOR Card - The *FOR card is used in an *ADD or *SUB section to indicate to which targets an ADD/SUB specification applies.

CARD FORMAT

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>*FOR</td>
<td>--</td>
</tr>
<tr>
<td>9-16</td>
<td>$t_1$</td>
<td>2A4</td>
</tr>
<tr>
<td>21-28</td>
<td>$t_2$</td>
<td>2A4</td>
</tr>
<tr>
<td>33-40</td>
<td>$t_3$</td>
<td>2A4</td>
</tr>
<tr>
<td>45-52</td>
<td>$t_4$</td>
<td>2A4</td>
</tr>
<tr>
<td>57-64</td>
<td>$t_5$</td>
<td>2A4</td>
</tr>
<tr>
<td>69-76</td>
<td>$t_6$</td>
<td>2A4</td>
</tr>
</tbody>
</table>
The $t_i$'s are either the name of a target (name field), the type of a target (type field), or ALL EXC. The ALL EXC clause can only be used in $t_1$, and is interpreted as for all except $t_2, \ldots, t_6$. All fields after a blank or invalid field are ignored. All specification cards between the *ADD or *SUB card and the first *FOR card are applied to all targets. Once a *FOR card is used, it applies to all the specification cards that follow it in that section until another *FOR card is encountered.

Examples of use:

*ADD
*FOR HARD NUC STOR
  1 ICBM 1 B52 SRAM
*FOR ALL EXC SS-18
  2 SLBM 1 OTHERS
*SUBL
  1 ICBM GO TRIDENT
*FOR AIRFIELD
  1 POLARIS 1 B52 SRAM 0 OTHERS
  GO POSEIDON L1 OTHERS

In the above examples, the first specification in the *SUB section applies to all targets, and both of the last two apply only to AIRFIELD.

*FTP Card - The *FTP card is used to signal that the cards that follow it are footprint degradation factor cards. If this card is not used, then the degradation factors from the previous problem are used. Until the first employment of an *FTP card, it is assumed that there is no degradation in weapon effectiveness due to footprinting.

CARD FORMAT

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>*FTP</td>
<td>--</td>
</tr>
</tbody>
</table>
Footprint Degradation Factor Card - Footprint degradation factor cards indicate relative effectiveness of the specified weapon name against the specified target name. (The relative effectiveness is by definition 1.0 if there is no degradation in effectiveness due to footprinting.)

**CARD FORMAT**

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>t&lt;sub&gt;1&lt;/sub&gt;</td>
<td>I2</td>
<td>27-28</td>
<td>w&lt;sub&gt;4&lt;/sub&gt;</td>
<td>I2</td>
</tr>
<tr>
<td>3-4</td>
<td>w&lt;sub&gt;1&lt;/sub&gt;</td>
<td>I2</td>
<td>29-32</td>
<td>f&lt;sub&gt;4&lt;/sub&gt;</td>
<td>F4.0</td>
</tr>
<tr>
<td>5-8</td>
<td>f&lt;sub&gt;1&lt;/sub&gt;</td>
<td>F4.0</td>
<td>33-34</td>
<td>t&lt;sub&gt;5&lt;/sub&gt;</td>
<td>I2</td>
</tr>
<tr>
<td>9-10</td>
<td>t&lt;sub&gt;2&lt;/sub&gt;</td>
<td>I2</td>
<td>35-36</td>
<td>w&lt;sub&gt;5&lt;/sub&gt;</td>
<td>I2</td>
</tr>
<tr>
<td>11-12</td>
<td>w&lt;sub&gt;2&lt;/sub&gt;</td>
<td>F4.0</td>
<td>37-40</td>
<td>f&lt;sub&gt;5&lt;/sub&gt;</td>
<td>F4.0</td>
</tr>
<tr>
<td>13-16</td>
<td>f&lt;sub&gt;2&lt;/sub&gt;</td>
<td>F14.0</td>
<td>...</td>
<td>t&lt;sub&gt;10&lt;/sub&gt;</td>
<td>I2</td>
</tr>
<tr>
<td>17-18</td>
<td>t&lt;sub&gt;3&lt;/sub&gt;</td>
<td>I2</td>
<td>73-74</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>19-20</td>
<td>w&lt;sub&gt;3&lt;/sub&gt;</td>
<td>I2</td>
<td>75-76</td>
<td>w&lt;sub&gt;10&lt;/sub&gt;</td>
<td>I2</td>
</tr>
<tr>
<td>21-24</td>
<td>f&lt;sub&gt;3&lt;/sub&gt;</td>
<td>F14.0</td>
<td>77-80</td>
<td>f&lt;sub&gt;10&lt;/sub&gt;</td>
<td>F4.0</td>
</tr>
<tr>
<td>25-26</td>
<td>t&lt;sub&gt;4&lt;/sub&gt;</td>
<td>I2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Each t<sub>i</sub> is a target name number, each w<sub>i</sub> is a weapon name number, and each f<sub>i</sub> is the relative effectiveness of weapon name w<sub>i</sub> against target name t<sub>i</sub>. There is no limit to the number of weapon name/target name combinations permitted, although only 10 such combinations can be indicated on a single card. However, any number of successive cards is permitted. If n weapon name/target name combinations are desired, then the columns corresponding to t<sub>n+1</sub> and w<sub>n+1</sub> must be left blank. (This may necessitate the inclusion of a blank card. For example, if n equals 10, blanks in columns corresponding to t<sub>11</sub> and w<sub>11</sub> are required.)

*COL Card - The *COL card is used to signal that the collateral damage feature in ALLOT is to be activated. Until the first employment of the *COL card, it is assumed that there is no collateral damage. Once this card has been employed, however, the collateral damage feature cannot be deactivated.
This feature assumes that the total damage resulting from a detonation is proportional to the cube root of the weapon yield. It occurs whenever the target attacked has a VN number of 15 or less.

**CARD FORMAT**

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>*COL</td>
<td>--</td>
</tr>
</tbody>
</table>

*END Card - The *END card denotes the end of the problem statement.

**CARD FORMAT**

<table>
<thead>
<tr>
<th>COLS</th>
<th>PARAMETER</th>
<th>FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-4</td>
<td>*END</td>
<td>--</td>
</tr>
</tbody>
</table>

**Miscellaneous**

(a) The *WPN, *TGT, *CWP, and *CTG section must precede all *ADD, *SUB, *FTP, and *COL sections. Other than this restriction, the various sections may appear in any order.

(b) Internally, weapon vectors are added according to the ADD cards before deletions according to the SUB cards take place.

(c) The name fields within the *WPN section must be unique.

(d) The name fields within the *TGT section must be unique.

(e) The type field within a section cannot be the same as a name field within that same section.

(f) All WPN or TGT specification cards having the same type field must be grouped together.

(g) If the number of target or weapon types is changed, then it is necessary to redefine the *ADD and *SUB sections.

(h) Null *ADD and *SUB sections are allowed. That is, it is permissible to have an *ADD or *SUB card followed by no specification cards in order to delete previous specifications.
Chapter V
SAMPLE CASES

To indicate the form of the output of ALLOT, and to assist the user in employing the model, a run consisting of three sample cases is presented below. The first two cases concern computations of RFS and include the two optional intermediate printouts normally used only for debugging purposes. The third case concerns a problem in which DE is maximized.

The input deck for the run is displayed below, followed by the complete output. The total running time for these three sample cases on the IDA computer was about a minute.
## INPUT DECK

<table>
<thead>
<tr>
<th>Card</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IN/OUT</td>
<td>MAT</td>
</tr>
<tr>
<td>1</td>
<td>START</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>STOP</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>GPN</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>WPN1</td>
<td>1001</td>
</tr>
<tr>
<td>1</td>
<td>WPN2</td>
<td>1002</td>
</tr>
<tr>
<td>1</td>
<td>WPN3</td>
<td>1003</td>
</tr>
<tr>
<td>1</td>
<td>WPN4</td>
<td>1004</td>
</tr>
<tr>
<td>1</td>
<td>WPN5</td>
<td>1005</td>
</tr>
<tr>
<td>1</td>
<td>WPN6</td>
<td>1006</td>
</tr>
<tr>
<td>1</td>
<td>WPN7</td>
<td>1007</td>
</tr>
<tr>
<td>1</td>
<td>WPN8</td>
<td>1008</td>
</tr>
<tr>
<td>1</td>
<td>WPN9</td>
<td>1009</td>
</tr>
<tr>
<td>1</td>
<td>WPN10</td>
<td>1010</td>
</tr>
<tr>
<td>2</td>
<td>TOT1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TOT2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TOT3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TOT4</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TOT5</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TOT6</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TOT7</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TOT8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TOT9</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TOT10</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>#ADD</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>#FOR</td>
<td>ORJ1</td>
</tr>
<tr>
<td>3</td>
<td>#SUB</td>
<td>ORJ1</td>
</tr>
<tr>
<td>3</td>
<td>#END</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>#ADD</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>#FOR</td>
<td>ORJ2</td>
</tr>
<tr>
<td>4</td>
<td>#SUB</td>
<td>ORJ2</td>
</tr>
<tr>
<td>4</td>
<td>#END</td>
<td></td>
</tr>
</tbody>
</table>

### Note
- The image contains a table with input data for a computational or simulation program.
- The table includes fields for WPN (Work Package Numbers), TOT (Totals), and some additional data such as ORJ (Operations Reference Numbers).
- The values are likely to be placeholders or specific values relevant to the context of the program's input deck.
SAMPLE CASE 1

The first sample case, an RFS computation, involves an allocation in which most target classes are limited to a maximum of one weapon per target. An ADD card allows OBJ3 targets to receive one SLBM weapon and one ICBM weapon, but this is the only exception to the one-weapon-per-target limit.
<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE NUMBER</th>
<th>N/M</th>
<th>YIELD</th>
<th>CEF</th>
<th>MSR</th>
<th>PLS</th>
<th>WSR</th>
<th>RADIUS</th>
<th>MIN DE</th>
<th>MAX DE</th>
<th>AVE DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP3</td>
<td>1-104</td>
<td>1</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>MP4</td>
<td>1-104</td>
<td>2</td>
<td>100</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>MP5</td>
<td>1-104</td>
<td>3</td>
<td>100</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>MP6</td>
<td>1-104</td>
<td>4</td>
<td>100</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>MP7</td>
<td>1-104</td>
<td>5</td>
<td>100</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>MP8</td>
<td>1-104</td>
<td>6</td>
<td>100</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>MP9</td>
<td>1-104</td>
<td>7</td>
<td>100</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>MP10</td>
<td>1-104</td>
<td>8</td>
<td>100</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>MP11</td>
<td>1-104</td>
<td>9</td>
<td>100</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>MP12</td>
<td>1-104</td>
<td>10</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

**Totals**

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE NUMBER</th>
<th>N/M</th>
<th>YIELD</th>
<th>CEF</th>
<th>MSR</th>
<th>PLS</th>
<th>WSR</th>
<th>RADIUS</th>
<th>MIN DE</th>
<th>MAX DE</th>
<th>AVE DE</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>0-103</td>
<td>1</td>
<td>100</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>S2</td>
<td>0-103</td>
<td>2</td>
<td>100</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>S3</td>
<td>0-103</td>
<td>3</td>
<td>100</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>S4</td>
<td>0-103</td>
<td>4</td>
<td>100</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>S5</td>
<td>0-103</td>
<td>5</td>
<td>100</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>S6</td>
<td>0-103</td>
<td>6</td>
<td>100</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>S7</td>
<td>0-103</td>
<td>7</td>
<td>100</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>S8</td>
<td>0-103</td>
<td>8</td>
<td>100</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>S9</td>
<td>0-103</td>
<td>9</td>
<td>100</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>S10</td>
<td>0-103</td>
<td>10</td>
<td>100</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

**Totals**

UNCLASSIFIED
### Table 1: Weapon Vectors

<table>
<thead>
<tr>
<th>Type</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The solution is limited to the following weapon vectors plus all vectors of the following types. For all possible weapon vectors with 1 or less weapons:

1. V1
2. V2
3. V3
4. V4
5. V5
6. V6
7. V7
8. V8
9. V9
10. V10

**UNCLASSIFIED**
### Target Results

<table>
<thead>
<tr>
<th>Target</th>
<th>Type</th>
<th>Value Damaged</th>
<th>Percentage</th>
<th>Targets Damaged</th>
<th>Percentage</th>
<th>Weapons Used</th>
<th>Megatons Used</th>
<th>EMT Used</th>
<th>CMP Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 TGT1</td>
<td>ORJ1</td>
<td>70.00</td>
<td>62.25</td>
<td>65.30</td>
<td>2.95</td>
<td>70.00</td>
<td>55.30</td>
<td>4.20</td>
<td>1.60</td>
</tr>
<tr>
<td>2 TGT2</td>
<td>ORJ1</td>
<td>6.00</td>
<td>5.00</td>
<td>6.00</td>
<td>0.00</td>
<td>6.00</td>
<td>5.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3 TGT3</td>
<td>ORJ2</td>
<td>6.00</td>
<td>5.00</td>
<td>6.00</td>
<td>0.00</td>
<td>6.00</td>
<td>5.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4 TGT4</td>
<td>ORJ3</td>
<td>60.00</td>
<td>52.00</td>
<td>60.00</td>
<td>2.55</td>
<td>60.00</td>
<td>52.00</td>
<td>2.55</td>
<td>1.05</td>
</tr>
<tr>
<td>5 TGT5</td>
<td>ORJ3</td>
<td>65.00</td>
<td>55.00</td>
<td>65.00</td>
<td>2.55</td>
<td>65.00</td>
<td>55.00</td>
<td>2.55</td>
<td>1.05</td>
</tr>
<tr>
<td>6 TGT6</td>
<td>ORJ3</td>
<td>65.00</td>
<td>55.00</td>
<td>65.00</td>
<td>2.55</td>
<td>65.00</td>
<td>55.00</td>
<td>2.55</td>
<td>1.05</td>
</tr>
<tr>
<td>7 TGT7</td>
<td>ORJ4</td>
<td>65.00</td>
<td>55.00</td>
<td>65.00</td>
<td>2.55</td>
<td>65.00</td>
<td>55.00</td>
<td>2.55</td>
<td>1.05</td>
</tr>
<tr>
<td>8 TGT8</td>
<td>ORJ4</td>
<td>65.00</td>
<td>55.00</td>
<td>65.00</td>
<td>2.55</td>
<td>65.00</td>
<td>55.00</td>
<td>2.55</td>
<td>1.05</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
<td>355.00</td>
<td>298.17</td>
<td>355.00</td>
<td>298.17</td>
<td>355.00</td>
<td>298.17</td>
<td>355.00</td>
<td>298.17</td>
</tr>
</tbody>
</table>

### Weapon Results

<table>
<thead>
<tr>
<th>Weapon</th>
<th>Type</th>
<th>Weapons Used</th>
<th>Percentage</th>
<th>Weapons Remaining</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>WPH1</td>
<td>1RMH</td>
<td>65.92</td>
<td>43.94</td>
<td>659.28</td>
<td>43.94</td>
</tr>
<tr>
<td>WPH2</td>
<td>1RMH</td>
<td>76.95</td>
<td>51.93</td>
<td>76.95</td>
<td>51.93</td>
</tr>
<tr>
<td>WPH3</td>
<td>1RMH</td>
<td>76.95</td>
<td>51.93</td>
<td>76.95</td>
<td>51.93</td>
</tr>
<tr>
<td>WPH4</td>
<td>1RMH</td>
<td>76.95</td>
<td>51.93</td>
<td>76.95</td>
<td>51.93</td>
</tr>
<tr>
<td>WPH5</td>
<td>1RMH</td>
<td>76.95</td>
<td>51.93</td>
<td>76.95</td>
<td>51.93</td>
</tr>
<tr>
<td>WPH6</td>
<td>1RMH</td>
<td>76.95</td>
<td>51.93</td>
<td>76.95</td>
<td>51.93</td>
</tr>
<tr>
<td>WPH7</td>
<td>1RMH</td>
<td>76.95</td>
<td>51.93</td>
<td>76.95</td>
<td>51.93</td>
</tr>
<tr>
<td>WPH8</td>
<td>1RMH</td>
<td>76.95</td>
<td>51.93</td>
<td>76.95</td>
<td>51.93</td>
</tr>
<tr>
<td>WPH9</td>
<td>1RMH</td>
<td>76.95</td>
<td>51.93</td>
<td>76.95</td>
<td>51.93</td>
</tr>
<tr>
<td>WPH10</td>
<td>1RMH</td>
<td>76.95</td>
<td>51.93</td>
<td>76.95</td>
<td>51.93</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
<td>659.28</td>
<td>43.94</td>
<td>659.28</td>
<td>43.94</td>
</tr>
<tr>
<td>Layout</td>
<td>Number</td>
<td>Unit</td>
<td>WPAMHS</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>------</td>
<td>---------</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Target 1</td>
<td>tot1</td>
<td>wn</td>
<td>76,55</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>17.19</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>6.26</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
<td></td>
<td>100.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Target 2</td>
<td>tot2</td>
<td>wn</td>
<td>100.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Target 3</td>
<td>tot3</td>
<td>wn</td>
<td>100.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Target 4</td>
<td>tot4</td>
<td>wn</td>
<td>100.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Target 5</td>
<td>tot5</td>
<td>wn</td>
<td>23.53</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>76.47</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
<td></td>
<td>100.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Target 6</td>
<td>tot6</td>
<td>wn</td>
<td>27.47</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>64.93</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
<td></td>
<td>100.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Target 7</td>
<td>tot7</td>
<td>wn</td>
<td>59.36</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>16.56</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>24.08</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
<td></td>
<td>100.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Target 8</td>
<td>tot8</td>
<td>wn</td>
<td>23.36</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>76.55</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Totals</td>
<td></td>
<td></td>
<td>100.00</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
SAMPLE CASE 2

The second sample case, also an RFS computation, uses the same weapon base and target base as the first sample case. However, it allows a maximum of two weapons per target (instead of one), except that attacks on OBJ3 targets by one SLBM weapon and one ICBM weapon are prohibited.
### WAPUN Allocation Mhenen

#### ARSENAL

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>NUMBER</th>
<th>HOR</th>
<th>YIELD</th>
<th>CEP</th>
<th>PLS</th>
<th>NSR</th>
<th>PTP</th>
<th>ENT</th>
<th>CMP</th>
<th>N MT</th>
<th>N ENT</th>
<th>N CMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WPN1</td>
<td>ICAP</td>
<td>150</td>
<td>A</td>
<td>10.00</td>
<td>4000</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td>1.732</td>
<td></td>
<td>650.000</td>
</tr>
<tr>
<td>2</td>
<td>WPN2</td>
<td>ICAP</td>
<td>150</td>
<td>A</td>
<td>2500</td>
<td>3000</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td>1.561</td>
<td></td>
<td>375,000</td>
</tr>
<tr>
<td>3</td>
<td>WPN3</td>
<td>ICAP</td>
<td>150</td>
<td>A</td>
<td>7500</td>
<td>2000</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td>1.425</td>
<td></td>
<td>365,000</td>
</tr>
<tr>
<td>4</td>
<td>WPN4</td>
<td>ICAP</td>
<td>150</td>
<td>A</td>
<td>1500</td>
<td>1000</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td>1.225</td>
<td></td>
<td>225,000</td>
</tr>
<tr>
<td>5</td>
<td>WPN5</td>
<td>SLAP</td>
<td>150</td>
<td>G</td>
<td>1500</td>
<td>1000</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td>1.060</td>
<td></td>
<td>150,000</td>
</tr>
<tr>
<td>6</td>
<td>WPN6</td>
<td>SLAP</td>
<td>150</td>
<td>G</td>
<td>1500</td>
<td>1000</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td>1.060</td>
<td></td>
<td>150,000</td>
</tr>
<tr>
<td>7</td>
<td>WPN7</td>
<td>SLAP</td>
<td>150</td>
<td>G</td>
<td>1500</td>
<td>1000</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td>1.060</td>
<td></td>
<td>150,000</td>
</tr>
<tr>
<td>8</td>
<td>WPN8</td>
<td>RMAP</td>
<td>150</td>
<td>G</td>
<td>1500</td>
<td>1000</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td>1.060</td>
<td></td>
<td>150,000</td>
</tr>
<tr>
<td>9</td>
<td>WPN9</td>
<td>BMAP</td>
<td>150</td>
<td>A</td>
<td>1500</td>
<td>1000</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td>1.060</td>
<td></td>
<td>150,000</td>
</tr>
<tr>
<td>10</td>
<td>WPN10</td>
<td>BMAP</td>
<td>150</td>
<td>A</td>
<td>1500</td>
<td>1000</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td>1.060</td>
<td></td>
<td>150,000</td>
</tr>
</tbody>
</table>

**TOTALS =** 1500

#### TARGET SYSTEM

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>NUMBER</th>
<th>VMTH</th>
<th>VALUE</th>
<th>N VALUE</th>
<th>MIN DE</th>
<th>MAX DE</th>
<th>AVE DE</th>
<th>RADIUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TG4</td>
<td>ORJ1</td>
<td>100</td>
<td>1e05</td>
<td>1-un</td>
<td>1e00</td>
<td>ø8000</td>
<td>ø8000</td>
<td>ø7000</td>
</tr>
<tr>
<td>2</td>
<td>TG4</td>
<td>ORJ1</td>
<td>100</td>
<td>1e04</td>
<td>1-un</td>
<td>1e00</td>
<td>ø8000</td>
<td>ø8000</td>
<td>ø7000</td>
</tr>
<tr>
<td>3</td>
<td>TG4</td>
<td>ORJ1</td>
<td>100</td>
<td>8-00</td>
<td>1-un</td>
<td>1e00</td>
<td>ø8000</td>
<td>ø8000</td>
<td>ø7000</td>
</tr>
<tr>
<td>4</td>
<td>TG4</td>
<td>ORJ1</td>
<td>100</td>
<td>4-50</td>
<td>1-un</td>
<td>1e00</td>
<td>ø8000</td>
<td>ø8000</td>
<td>ø7000</td>
</tr>
<tr>
<td>5</td>
<td>TG4</td>
<td>ORJ1</td>
<td>100</td>
<td>1-00</td>
<td>1-un</td>
<td>1e00</td>
<td>ø8000</td>
<td>ø8000</td>
<td>ø7000</td>
</tr>
<tr>
<td>6</td>
<td>TG4</td>
<td>ORJ1</td>
<td>100</td>
<td>1-00</td>
<td>1-un</td>
<td>1e00</td>
<td>ø8000</td>
<td>ø8000</td>
<td>ø7000</td>
</tr>
<tr>
<td>7</td>
<td>TG4</td>
<td>ORJ1</td>
<td>100</td>
<td>2-00</td>
<td>1-un</td>
<td>1e00</td>
<td>ø8000</td>
<td>ø8000</td>
<td>ø7000</td>
</tr>
<tr>
<td>8</td>
<td>TG4</td>
<td>ORJ1</td>
<td>100</td>
<td>1-00</td>
<td>1-un</td>
<td>1e00</td>
<td>ø8000</td>
<td>ø8000</td>
<td>ø7000</td>
</tr>
</tbody>
</table>

**TOTALS =** 800
## UP Table

<table>
<thead>
<tr>
<th>TARGET TYPE</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.939896</td>
<td>0.868408</td>
<td>0.782897</td>
<td>0.706805</td>
<td>0.635045</td>
<td>0.566813</td>
<td>0.499984</td>
<td>0.435650</td>
<td>0.372321</td>
<td>0.310000</td>
</tr>
<tr>
<td></td>
<td>0.939898</td>
<td>0.868408</td>
<td>0.782897</td>
<td>0.706805</td>
<td>0.635045</td>
<td>0.566813</td>
<td>0.499984</td>
<td>0.435650</td>
<td>0.372321</td>
<td>0.310000</td>
</tr>
<tr>
<td></td>
<td>0.949984</td>
<td>0.898992</td>
<td>0.820902</td>
<td>0.740900</td>
<td>0.660805</td>
<td>0.586791</td>
<td>0.510630</td>
<td>0.435321</td>
<td>0.370000</td>
<td>0.300000</td>
</tr>
<tr>
<td></td>
<td>0.949984</td>
<td>0.898992</td>
<td>0.820902</td>
<td>0.740900</td>
<td>0.660805</td>
<td>0.586791</td>
<td>0.510630</td>
<td>0.435321</td>
<td>0.370000</td>
<td>0.300000</td>
</tr>
<tr>
<td></td>
<td>0.954357</td>
<td>0.889531</td>
<td>0.810600</td>
<td>0.730600</td>
<td>0.649505</td>
<td>0.572421</td>
<td>0.498312</td>
<td>0.421400</td>
<td>0.350000</td>
<td>0.280000</td>
</tr>
<tr>
<td></td>
<td>0.954357</td>
<td>0.889531</td>
<td>0.810600</td>
<td>0.730600</td>
<td>0.649505</td>
<td>0.572421</td>
<td>0.498312</td>
<td>0.421400</td>
<td>0.350000</td>
<td>0.280000</td>
</tr>
</tbody>
</table>

### The Solution Is Limited to the Following Weapon Vectors

All possible weapon vectors with 2 or less weapons

Minus all vectors of the following types

For the following targets only OR-3

- All possible weapon vectors with
  - 1 Slim weapons
  - 1 Icon weapons
### Problem 2: Optimal Solution for Maximum Value Damaged

#### Target Results

<table>
<thead>
<tr>
<th>Target</th>
<th>Type</th>
<th>Value Damaged</th>
<th>Targets Damaged</th>
<th>Weights Used</th>
<th>Megatons Used</th>
<th>Ext Use</th>
<th>Cmp Use</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Number</td>
<td>Percent</td>
<td>Number</td>
<td>Percent</td>
</tr>
<tr>
<td>TGT1</td>
<td>OBJ1</td>
<td>70.00</td>
<td>70.00</td>
<td>19.00</td>
<td>62.50</td>
<td>0.41</td>
<td>0.11</td>
</tr>
<tr>
<td>TGT2</td>
<td>OBJ1</td>
<td>9.00</td>
<td>9.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>TGT3</td>
<td>OBJ2</td>
<td>9.00</td>
<td>9.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>TGT4</td>
<td>OBJ3</td>
<td>28.00</td>
<td>28.00</td>
<td>8.00</td>
<td>2.80</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>TGT5</td>
<td>OBJ3</td>
<td>65.00</td>
<td>65.00</td>
<td>17.32</td>
<td>54.20</td>
<td>6.63</td>
<td>1.11</td>
</tr>
<tr>
<td>TGT6</td>
<td>OBJ3</td>
<td>65.00</td>
<td>65.00</td>
<td>26.00</td>
<td>23.72</td>
<td>2.32</td>
<td>0.17</td>
</tr>
<tr>
<td>TGT7</td>
<td>OBJ4</td>
<td>55.00</td>
<td>55.00</td>
<td>15.90</td>
<td>12.60</td>
<td>5.46</td>
<td>2.24</td>
</tr>
<tr>
<td>TGT8</td>
<td>OBJ5</td>
<td>84.04</td>
<td>84.04</td>
<td>22.00</td>
<td>33.20</td>
<td>14.67</td>
<td>4.78</td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td></td>
<td><strong>305.00</strong></td>
<td><strong>305.00</strong></td>
<td><strong>719.05</strong></td>
<td><strong>47.96</strong></td>
<td><strong>717.92</strong></td>
<td><strong>47.96</strong></td>
</tr>
</tbody>
</table>

#### Weapon Results

<table>
<thead>
<tr>
<th>Weapon</th>
<th>Type</th>
<th>Weights Used</th>
<th>Weapons Remaining</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Number</td>
<td>Percent</td>
</tr>
<tr>
<td>WPN1</td>
<td>RRM</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>WPN2</td>
<td>RRM</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>WPN3</td>
<td>RRM</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>WPN4</td>
<td>RRM</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>WPN5</td>
<td>RRM</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>WPN6</td>
<td>RRM</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>WPN7</td>
<td>RRM</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>WPN8</td>
<td>RRM</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>WPN9</td>
<td>RRM</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>WPN10</td>
<td>RRM</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td><strong>Totals</strong></td>
<td></td>
<td><strong>719.05</strong></td>
<td><strong>47.96</strong></td>
</tr>
</tbody>
</table>
SAMPLE CASE 3

The third sample case maximizes target value destroyed. It allows a maximum of two weapons per target, and incorporates the collateral damage feature wherein a given weapon is assigned to more than one target.
### ARSENAL

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>NUMBER</th>
<th>MOR</th>
<th>YIELD</th>
<th>CEP</th>
<th>MLS</th>
<th>WSR</th>
<th>PTP</th>
<th>EMT</th>
<th>CMP</th>
<th>N HT</th>
<th>N EMT</th>
<th>N CMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>WPN1</td>
<td>A</td>
<td>700</td>
<td>4,000</td>
<td>3,000</td>
<td>100</td>
<td>1,000</td>
<td>2,236</td>
<td>0.325</td>
<td>250,000</td>
<td>1,545,240</td>
<td>227,424</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WPN2</td>
<td>A</td>
<td>600</td>
<td>3,000</td>
<td>2,000</td>
<td>100</td>
<td>1,000</td>
<td>1,732</td>
<td>0.333</td>
<td>150,000</td>
<td>846,025</td>
<td>166,407</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WPN3</td>
<td>A</td>
<td>500</td>
<td>2,000</td>
<td>1,000</td>
<td>100</td>
<td>1,000</td>
<td>1,000</td>
<td>0.250</td>
<td>100,000</td>
<td>360,000</td>
<td>75,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WPN4</td>
<td>A</td>
<td>400</td>
<td>1,000</td>
<td>500</td>
<td></td>
<td>500</td>
<td>500</td>
<td>0.430</td>
<td>250,000</td>
<td>316,906</td>
<td>139,991</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WPN5</td>
<td>A</td>
<td>300</td>
<td>500</td>
<td>250</td>
<td></td>
<td>250</td>
<td>250</td>
<td>0.342</td>
<td>140,000</td>
<td>239,397</td>
<td>239,397</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TOTALS</strong></td>
<td></td>
<td><strong>2700</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>5690.000</strong></td>
<td><strong>2285.664</strong></td>
<td><strong>846.218</strong></td>
<td></td>
</tr>
</tbody>
</table>

### TARGET SYSTEM

<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>NUMBER</th>
<th>VMTX</th>
<th>VALUF</th>
<th>N VALUE</th>
<th>MIN DE</th>
<th>MAX DE</th>
<th>AVE DE</th>
<th>RADIUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOT1</td>
<td>A</td>
<td>10</td>
<td>1024</td>
<td>1.000</td>
<td>10.00</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>TOT2</td>
<td>A</td>
<td>20</td>
<td>1024</td>
<td>1.000</td>
<td>10.00</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>TOT3</td>
<td>A</td>
<td>25</td>
<td>1024</td>
<td>1.000</td>
<td>10.00</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>TOT4</td>
<td>A</td>
<td>30</td>
<td>1024</td>
<td>1.000</td>
<td>10.00</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>TOT5</td>
<td>A</td>
<td>40</td>
<td>1024</td>
<td>1.000</td>
<td>10.00</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>TOT6</td>
<td>A</td>
<td>50</td>
<td>1024</td>
<td>1.000</td>
<td>10.00</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>TOT7</td>
<td>A</td>
<td>75</td>
<td>1024</td>
<td>1.000</td>
<td>10.00</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>TOT8</td>
<td>A</td>
<td>100</td>
<td>1024</td>
<td>1.000</td>
<td>10.00</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>TOT9</td>
<td>A</td>
<td>150</td>
<td>1024</td>
<td>1.000</td>
<td>10.00</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td><strong>TOTALS</strong></td>
<td></td>
<td><strong>2195</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Problem 3: Optimal Solution for Maximum Value Damaged

#### Target Results

<table>
<thead>
<tr>
<th>Target</th>
<th>Type</th>
<th>Value Damaged</th>
<th>Target Damaged Percent</th>
<th>Weapons Used Number</th>
<th>Weapons Used Percent</th>
<th>Megatons Used Number</th>
<th>Megatons Used Percent</th>
<th>Fmt Used Number</th>
<th>Fmt Used Percent</th>
<th>Cmp Used Number</th>
<th>Cmp Used Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TOT1</td>
<td>7.54</td>
<td>7.74%</td>
<td>6.85</td>
<td>.25</td>
<td>6.95</td>
<td>.12</td>
<td>6.05</td>
<td>.21</td>
<td>1.71</td>
<td>.20</td>
</tr>
<tr>
<td>2</td>
<td>TOT2</td>
<td>7.54</td>
<td>15.49%</td>
<td>13.69</td>
<td>.51</td>
<td>13.69</td>
<td>.24</td>
<td>13.69</td>
<td>.49</td>
<td>3.42</td>
<td>.60</td>
</tr>
<tr>
<td>3</td>
<td>TOT3</td>
<td>7.54</td>
<td>15.46%</td>
<td>17.12</td>
<td>.63</td>
<td>17.12</td>
<td>.30</td>
<td>17.12</td>
<td>.59</td>
<td>4.28</td>
<td>.50</td>
</tr>
<tr>
<td>4</td>
<td>TOT4</td>
<td>7.54</td>
<td>15.46%</td>
<td>27.39</td>
<td>1.01</td>
<td>27.39</td>
<td>.48</td>
<td>27.39</td>
<td>.87</td>
<td>6.05</td>
<td>.81</td>
</tr>
<tr>
<td>5</td>
<td>TOT5</td>
<td>7.54</td>
<td>15.46%</td>
<td>34.26</td>
<td>1.27</td>
<td>34.26</td>
<td>.60</td>
<td>34.26</td>
<td>1.04</td>
<td>6.56</td>
<td>1.01</td>
</tr>
<tr>
<td>6</td>
<td>TOT6</td>
<td>7.54</td>
<td>15.46%</td>
<td>68.47</td>
<td>2.54</td>
<td>68.47</td>
<td>1.20</td>
<td>68.47</td>
<td>2.08</td>
<td>17.12</td>
<td>2.02</td>
</tr>
<tr>
<td>7</td>
<td>TOT7</td>
<td>7.46</td>
<td>14.94%</td>
<td>114.25</td>
<td>4.23</td>
<td>114.25</td>
<td>4.05</td>
<td>114.25</td>
<td>3.56</td>
<td>33.51</td>
<td>3.80</td>
</tr>
<tr>
<td>8</td>
<td>TOT8</td>
<td>7.45</td>
<td>18.62%</td>
<td>146.68</td>
<td>5.21</td>
<td>146.68</td>
<td>4.96</td>
<td>146.68</td>
<td>4.45</td>
<td>40.99</td>
<td>4.83</td>
</tr>
<tr>
<td>9</td>
<td>TOT9</td>
<td>7.36</td>
<td>17.29%</td>
<td>277.47</td>
<td>8.80</td>
<td>277.47</td>
<td>12.52</td>
<td>277.47</td>
<td>8.57</td>
<td>70.93</td>
<td>9.32</td>
</tr>
<tr>
<td>10</td>
<td>TOT11</td>
<td>7.13</td>
<td>12.87%</td>
<td>423.36</td>
<td>15.68</td>
<td>423.36</td>
<td>12.52</td>
<td>423.36</td>
<td>14.67</td>
<td>138.65</td>
<td>16.35</td>
</tr>
<tr>
<td>TOTALS</td>
<td></td>
<td>74.61</td>
<td>150.82%</td>
<td>1683.51</td>
<td>46.13</td>
<td>1683.51</td>
<td>56.55</td>
<td>1799.97</td>
<td>54.74</td>
<td>333.92</td>
<td>39.32</td>
</tr>
</tbody>
</table>

#### Weapon Results

<table>
<thead>
<tr>
<th>Weapon</th>
<th>Type</th>
<th>Weapons Used Number</th>
<th>Weapons Used Percent</th>
<th>Weapons Remaining Number</th>
<th>Weapons Remaining Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>WP1</td>
<td>283.51</td>
<td>40.50</td>
<td>416.49</td>
<td>59.50</td>
</tr>
<tr>
<td>2</td>
<td>WP2</td>
<td>78.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>WP3</td>
<td>480.00</td>
<td>100.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>4</td>
<td>WP4</td>
<td>0.00</td>
<td>0.00</td>
<td>500.00</td>
<td>100.00</td>
</tr>
<tr>
<td>5</td>
<td>WP5</td>
<td>0.00</td>
<td>0.00</td>
<td>700.00</td>
<td>100.00</td>
</tr>
<tr>
<td>TOTALS</td>
<td></td>
<td>1183.51</td>
<td>46.13</td>
<td>1616.49</td>
<td>53.87</td>
</tr>
<tr>
<td>TARGET</td>
<td>NUMBER</td>
<td>DE</td>
<td>WEAPONS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>-----</td>
<td>---------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1  2  3  4  5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET 1</td>
<td>TOT1</td>
<td>3,42</td>
<td>0  0  2  a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS=</td>
<td>10,00</td>
<td>.7540</td>
<td>0  0  7  a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET 2</td>
<td>TOT2</td>
<td>6,05</td>
<td>0  0  2  a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS=</td>
<td>20,00</td>
<td>.7540</td>
<td>0  0  14 a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET 3</td>
<td>TOT3</td>
<td>8,56</td>
<td>0  0  2  a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS=</td>
<td>25,00</td>
<td>.7540</td>
<td>0  0  17 a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET 4</td>
<td>TOT4</td>
<td>13,69</td>
<td>0  0  2  a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS=</td>
<td>40,00</td>
<td>.7540</td>
<td>0  0  27 a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET 5</td>
<td>TOT5</td>
<td>17,12</td>
<td>0  0  2  a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS=</td>
<td>56,00</td>
<td>.7540</td>
<td>0  0  34 a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET 6</td>
<td>TOT6</td>
<td>34,24</td>
<td>0  0  2  a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS=</td>
<td>100,00</td>
<td>.7540</td>
<td>0  0  68 a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET 7</td>
<td>TOT7</td>
<td>4,18</td>
<td>0  0  2  a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS=</td>
<td>208,00</td>
<td>.7457</td>
<td>0  52 62 a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET 8</td>
<td>TOT8</td>
<td>52,34</td>
<td>0  1  1  a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS=</td>
<td>258,00</td>
<td>.7441</td>
<td>0  70 70 a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET 9</td>
<td>TOT9</td>
<td>116,73</td>
<td>0  2  0  a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS=</td>
<td>398,00</td>
<td>.7358</td>
<td>0  237 0 a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TARGET 10</td>
<td>TOT10</td>
<td>71,03</td>
<td>2  0  0  a  0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTALS=</td>
<td>1998,00</td>
<td>.7152</td>
<td>284 140 0 a  0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX A

MATHEMATICAL THEORY
Appendix A

MATHEMATICAL THEORY

The mathematical theory employed in ALLOT is best illustrated by an example. Suppose there are two target classes, consisting of $T_1$ and $T_2$ targets, respectively. Suppose further that there are $W$ identical weapons, each with a single-shot probability of kill (SSPK) of $K_1$ against targets in the first target class, and $K_2$ against targets in the second class. Let $V_1$ and $V_2$ be the values of targets in the first and second target classes, respectively, and assume that it is not permitted to place more than two weapons against any individual target. With these constraints, consider the following two questions:

1. What is the maximum possible value destroyed?
2. If it is desired to achieve average damage levels of at least $D_1$ and $D_2$, against the two target classes, respectively, what is the relative force size (RFS)?

Each of these two examples is amenable to solution by first expressing the problem in a general linear programming form, and then solving it by standard linear programming techniques.¹

¹In general, a noninteger solution will be obtained, although the answers may be truncated or rounded off by the user. In view of the aggregation usually involved in the preparation of the target data base, such activity is not likely in practice to have a significant effect on the accuracy of the solution.
The general linear programming form used in ALLOT is as follows:

Find values of the variables \( x_i \) (\( i=1,2,3,\ldots,m \)) and \( y_j \) (\( j=1,2,3,\ldots,n \)) such that, for given \( a_{ij} \) and \( b_i \)

\[
\sum_{j=1}^{n} a_{ij} y_j + x_i = b_i \quad (i=1,2,3,\ldots,m) \quad (1)
\]

\[
x_i \geq 0, y_j \geq 0 \quad (2)
\]

\[
x_1 = \text{minimum} \quad (3)
\]

The algorithm used to solve these equations is described in Appendix 13. The purpose of the present appendix is to show that each of the two examples presented above can be expressed in the general form of equations (1)-(3).

Consider the first problem - i.e., what is the maximum possible value destroyed? Define the following:

- \( x_1 \) = the total value surviving (i.e., not destroyed)
- \( x_2, y_1, y_2 \) = the numbers of targets in target class 1 assigned 0, 1, and 2 weapons, respectively.
- \( x_3, y_3, y_4 \) = the numbers of targets in target class 2 assigned 0, 1, and 2 weapons, respectively.
- \( x_4 \) = the number of weapons not used.

Then the requirements that the total numbers of targets in each class are \( T_1 \) and \( T_2 \), respectively, can be written

\[
y_1 + y_2 + x_2 = T_1 \quad (4)
\]

\[
y_3 + y_4 + x_3 = T_2 \quad (5)
\]
The requirement that the total number of weapons equals \( W \) is expressed as

\[
y_1 + 2y_2 + y_3 + 2y_4 + x_4 = W
\]  

(6)

The total value surviving, \( x_1 \), is given by

\[
x_1 = V_1[x_2 + y_1(1-K_1) + y_2(1-K_1)^2]
+ V_2[x_3 + y_3(1-K_2) + y_4(1-K_2)^2]
\]  

(7)

where \( V_1 \) and \( V_2 \) are the individual target values, and \( K_1 \) and \( K_2 \) are the SSPKs. If the expressions given in equations (4) and (5) for \( x_2 \) and \( x_3 \) are substituted into equation (7), the result is

\[
x_1 = V_1[(T_1 - y_1 - y_2) + y_1(1-K_1) + y_2(1-K_1)^2]
+ V_2[(T_2 - y_3 - y_4) + y_3(1-K_2) + y_4(1-K_2)^2]
\]  

(8)

Rearranging terms in equation (8) gives

\[
V_1[l-(1-K_1)]y_1 + V_1[l-(1-K_1)^2]y_2
+ V_2[l-(1-K_2)]y_3 + V_2[l-(1-K_2)^2]y_4 + x_1 =
V_1T_1 + V_2T_2
\]  

(9)

Equations (9), (4), (5), and (6) comprise the desired set of conditions. Comparison with equations (1)-(3) shows that
This completes the discussion of the first problem. The technique shown is quite general and is easily extended to accommodate cases involving additional target classes and/or weapon classes.

Consider now the second problem—i.e., what is the RFS? In this case, the procedure followed by ALLOT is to first determine the forces required to meet the specified goals, and then determine the ratio of these forces to those available.

Use the same definitions as before, except for $x_1$. Redefine $x_1$ as the inverse of the RFS:

$$x_1 = \frac{1}{\text{RFS}}$$

$x_2, x_3, x_4, y_1, y_2, y_3, y_4$ = same definitions as in first problem
Define, in addition,

\[ y_5 = \frac{1}{RFS} \]

\[ x_5 = \text{average damage in excess of required average damage } D_1 \text{ to targets in target class 1} \]

\[ x_6 = \text{average damage in excess of required average damage } D_2 \text{ to targets in target class 2} \]

Equations (4) and (5) are still applicable and are therefore retained. In equation (6), it is necessary to replace \( W \), the number of weapons available, by \( W y_5 \), the number of weapons actually required:

\[ y_1 + 2y_2 + y_3 + 2y_4 + x_4 = W y_5 \quad (10) \]

The equations describing \( x_5 \) and \( x_6 \) deal with the total value surviving in the two target classes, respectively, and are somewhat similar in form to equation (7):

\[ V_1 T_1 (1-D_1-x_5) = V_1 [x_2 + y_1 (1-K_1) + y_2 (1-K_1)^2] \quad (11) \]

\[ V_2 T_2 (1-D_2-x_6) = V_2 [x_3 + y_3 (1-K_2) + y_4 (1-K_2)^2] \quad (12) \]

With substitutions from equations (4) and (5), and rearrangement of terms, equations (11) and (12) become

\[ \frac{[-1+(1-K_1)] y_1 + [-1+(1-K_1)^2] y_2 + x_5}{T_1} = -D_1 \quad (13) \]

\[ \frac{[-1+(1-K_2)] y_3 + [-1+(1-K_2)^2] y_4 + x_6}{T_2} = -D_2 \quad (14) \]

Finally, the definitions of \( x_1 \) and \( y_5 \) require that

\[ -y_5 + x_1 = 0 \quad (15) \]
Accordingly, equations (15), (4), (5), (10), (13), and (14) comprise the desired set of conditions. Comparison with equations (1)-(3) shows that

\[
\begin{align*}
  a_{14} &= 0, & a_{34} &= 1, & a_{55} &= 0 \\
  a_{15} &= 1, & a_{35} &= 0, & a_{61} &= 0 \\
  b_1 &= 0, & a_{41} &= 1, & a_{62} &= 0 \\
  b_2 &= T_1, & a_{42} &= 2, & a_{63} &= \frac{[-1+(1-K_2)]}{T_2} \\
  b_3 &= T_2, & a_{43} &= 1, & a_{64} &= \frac{[-1+(1-K_2)^2]}{T_2} \\
  b_4 &= 0, & a_{44} &= 2, & a_{65} &= 0 \\
  b_5 &= D_1, & a_{45} &= -W \\
  b_6 &= D_2, & a_{51} &= \frac{[-1+(1-K_1)]}{T_1} \\
  a_{11} &= 0, & a_{52} &= \frac{[-1+(1-K_1)^2]}{T_1} \\
  a_{12} &= 0, & a_{53} &= 0 \\
  a_{13} &= 0, & a_{54} &= 0 \\
  a_{31} &= 0, & a_{32} &= 0, & a_{33} &= 1
\end{align*}
\]

This completes the discussion of the second problem. Once again, the technique is quite general and is easily extended to accommodate cases involving additional target classes and/or weapon classes.

The algorithm used to solve the general set of equations (1)-(3) is presented in Appendix B.
APPENDIX B

ALLOW LINEAR PROGRAMMING ALGORITHM
Appendix B

ALLOT LINEAR PROGRAMMING ALGORITHM

Consider the following linear programming problem: Find values of the variables \( x_i \) (\( i = 1, 2, 3, \ldots, m \)) and \( y_j \) (\( j = 1, 2, 3, \ldots, n \)) such that, for given \( a_{ij} \) and \( b_i \),

\[
\sum_{j=1}^{n} a_{ij} y_j + x_i = b_i \quad (1)
\]

\[ x_i \geq 0, \quad y_j \geq 0 \quad (2) \]

\[ x_i = \text{minimum} \quad (3) \]

(A trial solution is \( x_i = b_i, \quad y_j = 0 \))

This is, in fact, the desired answer if \( b_i \geq 0 \) for all \( i \), and if \( a_{ij} \leq 0 \) for all \( j \). If these conditions are not met, however, further work is necessary.

Given equations (1)-(3), the steps of the algorithm used in ALLOT\(^1\) to solve this problem are as follows:

1. If \( b_i \geq 0 \) for all \( i \leq 2 \), go to step 7.
2. Find an \( i \leq 2 \) for which \( b_i \leq 0 \). Call it \( a \).
3. Find the value of \( j \) for which \( a \alpha_j \) is the smallest. Call it \( \beta \). Then \( a_{a \beta} \leq a_{a \beta} \) for all \( j \).
4. If \( a_{a \beta} \geq 0 \), the problem as stated above in equations (1)-(3) has no solution.

\(^1\)This algorithm is a variant of the well-known simplex algorithm.

B-1
5. Consider all $l \geq 2$ for which $b_1 \geq 0$, and $a_{1l} > 0$. Of these, find the value of $l$ for which $(b_1/a_{1l})$ is the smallest. Call it $\gamma$. (If there is no $l \geq 2$ such that $b_1 \geq 0$ and $a_{1l} > 0$, set $\gamma$ equal to $a$.)


7. (Via step 1). If $a_{lj} \leq 0$ for all $j$, the problem is solved. The solution is $x_1 = b_1$, $y_j = 0$.

8. If there exists an $a_{lj} > 0$, then find the value of $j$ for which $a_{lj}$ is the largest. Call it $\beta$. Then $a_{1\beta} > a_{lj}$ for all $j$, and $a_{1\beta} > 0$.

9. Consider all $l \geq 2$ for which $a_{1l} > 0$. Of these, find the value of $l$ for which $(b_1/a_{1l})$ is the smallest. Call it $\gamma$. (If there is no $l \geq 2$ for which $a_{1l} > 0$, the problem as stated above in equations (1)-(3) is unbounded.)

10. (Via step 6 or step 9). Consider equation (1) with $i = \gamma$:

\[ \sum_j a_{\gamma j} y_j + x_\gamma = b_\gamma \]  

11. Divide both sides of this equation by $a_{1\beta}$:

\[ \sum_{j \neq \beta} (a_{\gamma j}/a_{1\beta}) y_j + (1/a_{1\beta}) x_\gamma + y_\beta = (b_\gamma/a_{1\beta}) \]  

1.e., $y_\beta = (b_\gamma/a_{1\beta}) - \sum_{j \neq \beta} (a_{\gamma j}/a_{1\beta}) y_j - (1/a_{1\beta}) x_\gamma \)  

12. Use equation (6) to substitute for $y_\beta$ in the remainder of equations (1):

\[ \sum_{j \neq \beta} [a_{1j} - (a_{1\beta} a_{\gamma j}/a_{1\beta})] y_j - (a_{1\beta}/a_{1\beta}) x_\gamma + x_1 = b_1 - (a_{1\beta} b_\gamma/a_{1\beta}) (1 \neq \gamma) \]  

B-2
13. Define $u_1 \equiv x_1$ for all $i \neq \gamma$; $u_\gamma \equiv y_\beta$; $v_j \equiv y_j$ for all $j \neq \beta$; and $v_\beta \equiv x_\gamma$.

14. Define $e_{ij} \equiv a_{ij} - (a_{i\beta} a_{j\gamma}/a_{\gamma\beta})$ for all $i \neq \gamma$ and $j \neq \beta$; $e_{i\beta} \equiv -(a_{i\beta}/a_{\gamma\beta})$ for all $i \neq \gamma$; $e_{j\gamma} \equiv (a_{j\gamma}/a_{\gamma\beta})$ for all $j \neq \beta$; and $e_{\gamma\beta} = (1/a_{\gamma\beta})$.

15. Define $d_1 \equiv b_1 - (a_{1\beta} b_{\gamma}/a_{\gamma\beta})$ for all $i \neq \gamma$; and $d_\gamma \equiv b_\gamma/a_{\gamma\beta}$.

16. With the above definitions equations (1)-(3) may be written via equations (5)-(7) as:

$$
\sum_j e_{ij} v_j + u_1 = d_1
$$

(9)

$$
u_1 \geq 0, \quad v_j \geq 0
$$

(10)

$$u_1 = \text{Minimum}
$$

(11)

(Equation (11) follows because $\gamma \geq 2$, and therefore $u_1 = x_1$.)

17. Equations (9)-(11) have exactly the same form as equations (1)-(3). Thus, the above procedure—i.e., steps (1)-(16)—may be applied again. The next step, therefore, is to go back to step 1 and iterate (using the $e_{ij}$, $v_j$, $u_1$, and $d_1$, respectively, instead of the $a_{ij}$, $y_j$, $x_1$, and $b_1$).

The above is the algorithm used in ALLOT to solve the allocation problem. The detailed coding uses a matrix framework to save computer storage, but in all other respects is identical to the description given here.
APPENDIX C

LISTING OF ALLOT PROGRAM
PROGRAM ALLOT(INPUT, OUTPUT, TAPE=INPUT, TAPE=OUTPUT)
COMMON /BLKA/ B(I), C(201), N, W, Y
COMMON /BLKB/ NP, NT, NV, NV, UR, J
COMMON /BKLD/ P(I), F(I)
COMMON /BLK/ J(I), A(I)
COMMON /BLK/ N(I, J), N(I, J), N(I, J)
COMMON /BLK/ L(I, J), I(I, J), I(I, J)
COMMON /BLK/ NAME(I), NAME(I), NAME(I)
COMMON /BLK/ V(I), V(I), V(I)
COMMON /BLK/ U(I), U(I), U(I)
COMMON /BLK/ P(I), P(I), P(I)
COMMON /BLK/ I(I), I(I), I(I)
COMMON /BLK/ NAME(I), NAME(I), NAME(I)
COMMON /BLK/ V(I), V(I), V(I)
COMMON /BLK/ U(I), U(I), U(I)
COMMON /BLK/ P(I), P(I), P(I)
DIMENSION IOCONT(U)
EQUIVALENCE (PUN, IOCONT(U))
LOGICAL PUN, USE, CONTINUE, NAME(I), TITLE(I), OBJCON(I), IDGR
*...
END
SUBROUTINE INPUT

C NOTE -- DEUTL IS THE TARGET RADIUS IN NAUTICAL MILES.

C

COMMON /ALKA/ B(IS),U(IS),X(IS),Y(IS),Z(IS)
COMMON /ARK/ NP,NV,NV,IOB
COMMON /ALD/ PB(IS),PS(IS),F(IS),Z(IS)
COMMON /ALE/ PND(IS),C(IS),W(IS),U(IS),V(IS),T(IS)
COMMON /ALF/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /BLK/ BKL(IS),D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALG/ B(I1),V(I1),S(I1),G(I1),U(I1),D(I1)

COMMON /ALK/ B(I2),U(I2),V(I2),S(I2),T(I2),C(I2)

COMMON /ALK/ BKL(I1),D(I1),N(I1),S(I1),T(I1),C(I1)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ B(IS),U(IS),X(IS),Y(IS),Z(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)

COMMON /ALK/ N(IS),V(IS),NP(IS),IM(IS),I(M),Y(IS),Y(IS),T(IS)

COMMON /ALK/ D(IS),N(IS),S(IS),T(IS),C(IS)
C-3

UNCLASSIFIED
UNCLASSIFIED

DO 5310 I=1,10
   IF(LC=0.LO.LUN(I),141)=1
   CONTINUE
   IF(N.EQ.10) N=N+1
   IF(N.NE.3) GO TO 8000
5320 JSW=0
   DO 5330 J=1,9
   OBJCON(I)=.FALSE.
   DO 5330 J=1,9
   IF(LFLU(I).EQ.LO.LO.LJ(I)) GO TO 5340
   OBJCON(I)=.TRUE.
   JSW=1
5330 CONTINUE
   IF(JSW.EQ.0) NZ=1
   IF(JSW.EQ.0) OBJCON(I)=.TRUE.
   GO TO 400
400 READ(195A,942) ICONT
   C
   C DECODE CONTROL CARD
   C
   4010 DO 5420 I=1,9
   IF(CONT.EQ.LCUN(I)) GO TO (11000,1010,1007,1017,2100,2100,2500,2500,2500)
   18201,1
5420 CONTINUE
   IF(N.EQ.0) N=1
   IF(N.EQ.0) CONT=2
   GO TO 540
50 CONTINUE
   C
   C INPUT WEAPON INFORMATION
   C
   1000 CODE=.FALSE.
   JSW=1
   NWC=1
1001 I=1
   READ(199A,940) CONT,(NAME(J),J=1,10),TYPE(I),NUME(I),D1,D2,
   HOB(I),WLD(I),CEP(I),PLM(I),WDR(I),PTP(I)
   IF(CONT.EQ.IBLANK) GO TO 5410
   NWC=1
   IF(NW.EQ.20) GO TO 8000
   IF(HOB(I).EQ.0) HOB(I)=1
   IF(PLM(I).EQ.0) PLM(I)=1
   IF(WDR(I).EQ.0) WDR(I)=1
   IF(PTP(I).EQ.0) PTP(I)=1
   IF(YLD(I).EQ.0) YLD(I)=1
   IF(CEP(I).GT.0) CEP(I)=1
   IF(YLD(I).EQ.1) YLD(I)=0
   IF(CEP(I).EQ.0) CEP(I)=0
   GO TO 1005
1005 CONT=CONT+CEP(I)-CEP(I)
   CONT=CONT+WLD(I)-WLD(I)
   IF(YLD(I).EQ.0) YLD(I)=1
   IF(CEP(I).EQ.0) CEP(I)=1
50 CONTINUE
   1000 READ(199A,940) CONT,(NAME(J),J=1,10)
GO TO 3400

INPUT TARGET INFORMATION

1010 CDE=.FALSE.
    JSW=1
    NT=0
    I=0

1011 READ(IUSR911) ICONT, (TNAMn(I), JM12), ITYPE(I), NUM(I), VN(I),
    P0(I), AO(I), (DENm(I), DENAX(I), DEave(I), JEDLT(I)
    IF(ICAL(NN).IE.1) GO TO 9410
    NT=1
    IF(NT.GT.20) GO TO 890
    IF(DENmax(I).LT.EPS) DENmax(I)=0.
    IF(DENmax(I).LT.EPS) DENAX(I)=0.
    IF(ABS(DEave(I)).LT.EPS) DEave(I)=0.
    IF(DEDLT(I).LT.EPS) DEDLT(I)=0.
    IF(V(I).LE.EPS) V(I)=1.
    IF(DENm(I).LE.NNTER) DENm(I)=1.
    GO TO 1811, 3090, JSW

READ NEW TARGET NUMBERS

1017 READ(IUSR910) (NUM(J), JM1, NT)
    GO TO 3400

CHANGE OR ADD WEAPON

4000 CDE=.FALSE.
    JSW=1
    NT=0
    I=0

4010 CONTINUE
    GO TO 1901

4020 CONTINUE
    GO TO 1901

4030 NUM(J) = NUM(I)
    M0(J) = M0(I)
    YM(I) = YM(I)
    LEP(J) = LEP(I)
    LPS(J) = LPS(I)
    WSR(J) = WSR(I)
    WTP(J) = WTP(I)
    EMT(J) = EMT(I)
    CMP(J) = CMP(I)
    I=1
    NM=1
    NT=0
    GO TO 1901

CHANGE OR ADD TARGET

4050 CDE=.FALSE.
    JSW=1
    NT=0
    GO TO 1911
UNCLASSIFIED

C-6
UNCLASSIFIED

C-7
C INPUT FOOTPRINT EFFICIENCY

1400 CONTINUE
READ(USR+,949) (XKK,JXK) DO 1230 K=1,10
1230 CONTINUE
GO TO 1260

C CALLER DAMAGE FACTOR

1400 CONTINUE
DO 1320 J=1,10
XYZPA+YLD(J) DO 1310 NT IF (VH(J),BT,15) GO TO 1310
PTP(J+JT)PNUM(K) 1310 CONTINUE
1320 CONTINUE
GO TO 1340

C OUTPUT WEAPON INFORMATION

C

WRITE(6,921)HP$TITLE
WRITE(6,922) WIN
YLOSUM, ENTSUM, CMPSUM, DO 1290 IM+NW
1290 CONTINUE
VLOM=(NAME(Z(J)),MT(J),Z(J),NMT(J),X(J),X(J)), CEP(J),
LSU=NSUM+NUMW(J)
YLOM,YLSUM+01
ENTSUMP, NTSUM+2
1290 CONTINUE
C OUTPUT TARGET INFORMATION

WRITE(6,923) NTSUM
VSUM=NSUM+NUMW(J)
NTSUM+NTSUM+NUMW(J)
1030 VSUM = VSUM+01
WRITE(6,917)INTSUM,VSUM

C CALCULATE DE TABLE
C
IF(CQ.EQ.10 TO 1030
DO 1033 I=1,NT
IT=1
IF(PD(I).EQ.KG)IT=2
DO 1033 J=1,NW
PA=PLS(J)+QWR(J)*PTP(J)
Y=100.0*YLO(J)
CSAVE=CEP(J)
CEP(J)=SORT(CEP(J),CEP(J),*,M=94576.*6476.*DELT(I)*DELT(I)
CALL UPOXY(VN(I),*4343(I),*943(I),*4343(J),*4343(J),
CEP(J))=SAVE
PK(J))=PA
1033 CONTINUE

C OUTPUT DE TABLE
C
WRITE(6,924)
1042 WRITE(6,925)
IF(JJ,MT,NE)JJ=NN
WRITE(6,926)(I,II,JJ)
DO 1045 J=1,NT
WRITE(6,927)(I,J,JJ)
IF(JJ,NE,NE)=GO TO 1042

C OUTPUT VECTOR SPECS
C
IF(KB.EQ.11)GO TO 0000
WRITE(6,929)
DO 1140 KB=1,2
IF(KC,EQ.10)GO TO 1149
IC=3
IF(KB.EQ.11)WRITE(6,929)
IF(KB,EQ.2)WRITE(6,929)
DO 1130 I=1,NE
K=1
IF(KB,EQ.2)K=1111=I
IF(KB,EQ.2)GO TO 1060
DO 1059 J=1,NT
IF(INAP(J,K),NE,INAP(J,K))GO TO 1060
1059 CONTINUE
GO TO 1109

1060 DO 1095 J=1,NT
IF(INAP(J,K),NE,JB)GO TO 1070
1069 CONTINUE
WRITE(6,944)
GO TO 1109

1070 DO 1095 J=1,NT
IF(INAP(J,K),NE,JB)GO TO 1070
1079 CONTINUE
IF(KB.LT.2)IAP(J,J)=IAP(J,J-1)
DO 120 J=1,2
NC(I,J)=0
NC(I,J)=NC(I,J)+1
120 CONTINUE
IF(IAP(J,JL)=IAP(J,JL))GO TO 1100
CONTINUE
DO 160 J=1,2
IF(L(J,J)=.GT.0)NC(J,J)=NC(J,J)+1
160 CONTINUE
IF(NC(J,J).NE.1)GO TO 1100
CONTINUE
1100 WRITE(6,931)I,J
WRITE(6,941)I,J,NB(I,J)
WRITE(6,943)I,J
CONTINUE
C BUILD A MATRIX
1900 DO 1910 I=1,NW
1910 WRITE(6,945)I
CONTINUE
DO 1920 I=1,NW
1920 NW
UNCLASSIFIED

1920 B(M)=NUMT(1)
J=0
K=0
DO 1930 I=1,NT
IF(DEAVE(I),EQ,0) GO TO 1930
IF(J=NUM(I),J=1)
IF(DEAVE(I),LT,0) GO TO 1930
M=M+1
IF(N=J,50) GO TO 8000
XL=0
DO 1925 K=1,1
XL=XL+X(K)*FLAT(NUMT(K))
1925 CONTINUE
B(M)=AL=DEAVE(I)
L=M+M2
IF(L,GT,20) GO TO 8000
JSRT(L)=J
JEND(L)=I
J=0
1930 CONTINUE
IF(J=NUM,0) GO TO 8000
RETURN
8000 WRITE(4,946)
STOP 1
8000 WRITE(9,947)
DO 9100 J=1,NT
WRITE(9,948)(ISUM(I,J),I=1,Z1),(XSUM(I,J),I=1,13)
9100 CONTINUE
STOP
END
SUBROUTINE DPDX(V4,T,K,TG,HP,V4DP,C,DOUT)
INTEGER TV, VN
DIMENSION A(7,2,2,19,4,4,10,2)
DATA A / 
0.211+0.111+6.295E-4+2.195E-5+6.62NE-7, 7.133E-9+3.84E-11;
0.783+0.1365, 2.355E-3+2.59E-4, 9.91E-6+1.07E-7, 1.27E-9;
0.878+0.1033+7.7,93E-4+8.03E-5, 1.09E-6+5.2OPE-7, 8.73E-9;
DATA B / 
1.110+0.659E-5+4.86E-4, 5.03E-5+1.98E-6, 2.94E-9;
DATA C / 
1.099046, 2.17142, 0.260026, -0.752178;
DATA D / 
0.21546, 0.29549, 0.46441/;
1.2491, 1.5901, 0.017162, 3.182161;
3 -1.1972, 5.3593, -0.007088, 2.166695;
4 -2.4046, 7.7429, -0.229998, -3.453694;
5 -3.1168, 17.722, -0.817090, -0.513864;
6 -5.0981, 14.8336, -9.374175, -0.476844;
7 -6.0979, 20.1985, -9.961476, -0.315469;
8 -8.0591, 26.9991, -1.513217, -3.11471;
9 -12.766, 44.9996, -5.05664, 12.216/;
0 0;
1 0.24917, 0.799895, -0.0399345, 0.0011529;
2 0.491923, 1.76786, -0.180471, 0.0010949;
3 0.881143, 2.06087, -0.311596, 0.0014460;
4 1.40821, 4.19412, -0.135490, 0.0018198;
5 1.90863, 5.91546, -0.336323, 0.0024570;
6 2.50967, 8.17013, -0.230725, 0.0041760;
7 3.12376, 11.5372, -0.158126, 1.120297;
8 4.31986, 16.0686, -0.211217, 4.00466;
9 6.01380, 24.6439, 0.338797, 10.6430/;

IF (C<0) GO TO 10
Km=1
VwN=(1+K,T)K*4(32+K,T)K*4(0+K,T)K*4(0+K,T)K
S=EXP(A13*1+T)*VE4(A2+H1+T)*VE4(A3+H2+T)*VE4(A4+H3+T)*VE4(A5+H4+T)*
D=VE4(A2+H1+T)*VE4(34+H2+T)*VE4(35+H3+T)*VE4(35+H4+T)*
 Astros
Uw/C
IF(U,0,1,10,10,10) GO TO 10
IF(U,0,1,110,10) GO TO 20
PaExp=EXP(0)
Dm(W)=PaExp(U) (8+BT) 2+W*(3+1)+W+9B*4+T)+UW)
DpDC=UpDpU/C
DpDpU=UaDpU/C
RETURN
10 Pa=1
Do TO 1A
20 Pa=0
30 DpDC=0
DpDpU=0
RETURN
END
SUBROUTINE BUILDA
COMMON /BLKA/ B(50)*,C(200),N,M,NZ
COMMON /BLKB/ NP,NT,NW,NT2,T10
COMMON /BLKC/ N2,N3,INP,INP1,INP2,IVN,IVW(6),I10
COMMON /BLKD/ YLQ(5,10),PWL(10),INAP(20,10)
COMMON /BLKE/ PSTENT,HRATU,TRAFND,FINISH,SKPCMK
COMMON /BLKI/ V(11),DEM(8),DEM1(8),DELY(8),DELX(8)
COMMON /BLKL/ YLD(21),ENT(21),CHP(21)
COMMON /BLL/ IR,IN,IAN,IN
COMMON /BLK/ IA(6,200),IA(6,10)
COMMON /41.11/ IR(I),IAN(I)
COMMON /41.2/ V(1),DCN,DCV,ISO,ISO1,ISO2,ISO3
COMMON /41.3/ VLD(1),CNP(1),CNTP(1)
COMMON /41.4/ VLD(1),CNP(1),CNTP(1)
COMMON /41.5/ VLD(1),CNP(1),CNTP(1)
COMMON /41.6/ VLD(1),CNP(1),CNTP(1)
COMMON /41.7/ VLD(1),CNP(1),CNTP(1)
COMMON /41.8/ VLD(1),CNP(1),CNTP(1)
COMMON /41.9/ VLD(1),CNP(1),CNTP(1)
COMMON /41.10/ VLD(1),CNP(1),CNTP(1)
COMMON /41.11/ VLD(1),CNP(1),CNTP(1)
COMMON /41.12/ VLD(1),CNP(1),CNTP(1)
COMMON /41.13/ VLD(1),CNP(1),CNTP(1)
COMMON /41.14/ VLD(1),CNP(1),CNTP(1)
COMM}
1450 CONTINUE
C(1) = 0.
1460 CONTINUE
1470 CONTINUE
B(M+1) = 1
RETURN
END
FUNCTION OBJECT((IC:IT:IP))
COMMON /01/ NR,M1, N0.MV,?Of)
COMMON /01/ P1,20.0.P0),FT D12~ .?0)
COMMON ,a1.KI, V(PI),DEAVE(,l).IIEMA ,(2l),UE)1.q(Zl),
• YLDI?1).ENTI,1).CMP(,l)
COMMON /01/ IR ?O ,IAPI 2i)
COMMON /RtD / Nvw,KA ,F

DIMENSION IWP(3)
DIMENSION DSJ(2i.;)

EQUIVALENCE (OBJSI.fl.DEAVE,R)

DATA EPS/5.UsØ5111•~/

ICNT=0

DO 65 IM=+N1
IF(IC=LE,0)IC=IR(I)
IF(IC=V,0) CALL ADD(1,IC,A)
IFIC=Q,0) IM=FIX(*0.5)
PS=IM=PK(1T J)
50 IF(IW=LE,0)GO TO 60
PC=PS
ICNT=ICNT+1
IM=IM+1
GO TO 50
60 CONTINUE
F=1,
IF(1C=EQ,0) GO TO 70
F=0,
SUM
DO 65 J=1,ICNT
SUM=SUM+K(1T J)*PT(1T J)
65 CONTINUE
C=SUM
70 CONTINUE
OBJECT=PC

C CHECK MIN AND MAX DE
75 IF(P=GT,1.+DEMNI(TI)*EPS,AND.P=LT,1.-EPS) GO TO 88
IF(P=LT,1.-DEMAX(TI)-EPS) NO TO 88
GO TO (100,200,300,350,350,109J)
80 OBJECT=11001,
RETURN

C MAX DE
100 OBJECT=V(1T)*OBJECT
RETURN

C M14 WEAPONS
200 OBJECT=ICNT
RETURN

C M15 MEGATONS, ENT, OR CMP
300 OBJECT=I,
IF(1C=EQ)RETURN
DO 310 J=ICNT
10=IM(J)
310 OBJECT=OBJECT*OBJ(10,1NDJ)
RETURN
END
SUBLTINE CREATE
COMMON MLK/ M0, MW, MV, NIV (3D)
COMMON MLK/ NT, MV (3D), NIV (2D)
COMMON MLK/ NIV (3D), MIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMMON MLK/ MIV (2D)
COMMON MLK/ NIV (3D)
COMMON MLK/ NIV (2D)
COMM
UNCLASSIFIED

1060 DO 1110 J=1,NSS
   IF(J.EQ.NSS .AND. NSEC.GT.NJ) GO TO 1070
   ISIZE(J)=INUB(J,KA)-INLB(J,KA)+1
   GO TO 1090
1070 ISIZE(NSS)=0
   DO 1080 K=1,NSEC
   ISIZE(NSS)=ISIZE(NSS)*JU(J)-JL(K)+1
   CONTINUE
   DO 1080 K=1,NSEC
   IR(J)=NSW(J)
   IF(ISIZE(J).EQ.1) GO TO 1110
   L=ISIZE(J)-1
   DO 1100 K=1,L
   1100 IR(J)=IR(J)+1
   IFLAG(J)=.TRUE.
   CONTINUE
   C TRANSFER IR: SECTIONS TO IRR IF IN SECTION HAS CHANGED (IFLAG(I)=TRUE)
   C
   1110 IF(NS.EQ.0) GO TO 1220
   DO 1120 J=1,NSEC
      IF(NOT.IFLAG(NSS)) GO TO 1220
      IFLAG(J)=.FALSE.
      K=ISIZE(I)
      L=INLB(I,K)-1
      DO 1110 J=1,L
      L=M+JIK
   1110 IR(J)=M+JIK
   CONTINUE
   C TRANSFER IR(NSS) TO IRR IF IT HAS BEEN CHANGED (IFLAG(NSS)=TRUE)
   C
   1220 IF(.NOT.IFLAG(NSS)) GO TO 1235
      IFLAG(NSS)=.FALSE.
   K=0
   DO 1230 I=1,NSEC
      IB=JU(I)
      IR(J)=IR(J)+IB
   1230 CONTINUE
   C CALL INSERT AND RETURN 1 IF THE MATRIX IS FULL
   C
   1235 CALL INSERT
   C BUMP ONE ROW DOWN ONE PLACE. 1F ALL ROWS ARE AT BOTTOM 30 TO 1260
   C
   J=1
   1240 IF(IR(I,J).EQ.NSEC(I)) GO TO 1260
      IFLAG(I)=.TRUE.
   J=1
   1250 IR(J)=1
   IF(IR(I,J).LT.I) GO TO 1290
      IR(J)=IR(J)-1
      K=IR(I,J)+1
      IR(I,J)=K
   1290 CONTINUE
   GO TO 1240

C-17

UNCLASSIFIED
C
C REFINTIALIZE SECTION I. IF I\^WSS RETUOJ PLRE INCREMENT I

1d6N IFLAG(1)=TKE.
IR(I)=10
I=I+1
IR(1)=N(I)
I=I+1
IF(I.GE.MS)GO TO 1240
RETURN
END
SUBROUTINE INSERT
COMMON ABLKA/ B(5), C(200) /M, N,F
COMMON ABLKB/ N,F, NT /M, NW, HJ
COMMON ABLKF/ N,F, NT /M, NW, HJ
COMMON ABLK/ N,F /M, N,F
* INLK, INLJK /M, N,F, A, B, C, D, E, N, F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMMON ABLK/ N,F /M, N,F * 
COMM
J=J+1
CALL ANAT(J+1, A)
IF(A.EQ.0.) GOTO 115
IN(J+1)=0
120 CONTINUE
220 DO 250 I=1,NT
IF(IN(I).EQ.0) GO TO 250
CNP(I)=OBJECT(I)+I
IF(CNP(I) .GT. 10000.) GO TO 250
NV=MV+1
IA(I,MV)=1
DA(MV)=NV*(1+(Y-I)*(Y-I))
TA(MV)=F
RA(I)
DO 240 J=1,NW
IF(IR(J),EQ.0) GO TO 240
IF(IR(J),GT.1) GO TO 230
K=K+1
IA(K,NV)=J
GO TO 240
230 CONTINUE
IF(IR(J),EQ.2) IA(4,NV)=J
IF(IR(J),EQ.3) IA(3,NV)=J
240 CONTINUE
250 CONTINUE
IF(NV+MV+LT.2986) RETURN
WRITE(6,1001)
STOP 2
1001 FORMAT(21,WEAPON VECTOR LIMIT EXCEEDED*)
END
SUBROUTINE SIMPLE
COMMON /ALRA/ B(51), C(2001), N, N
COMMON /ALRA/ NP, NT, NN, NV, TUG
COMMON /ALRA/ JA(51), IA(51)
COMMON /ALRA/ Q(51), S(51)
COMMON /ALRA/ J(51), I(51)
DIMENSION Q(51), S(51)
DATA E2 = 5.0, 0.000000001

INITIALIZE
IFLAG = 0
NS = NN
NS = NN
NS = NN
Do 10 NS = 1, NS
Do 10 JS = 1, JS
10 Q(J) = 0
DI(J) = 0
30 CONTINUE
IF(NS.EQ.0) GO TO 40
DI(J) = 0
40 CONTINUE
Do 50 JS = 1, JS
Di(J) = 0
50 JS = NS
Do 50 JS = 1, JS
50 JS = NS
10 JS = 0
DO JS = 1, JS
CALL AI(IT, J, TI)
IF(T1.gt.T2) GO TO 120
120 CONTINUE
IT = 1
JS = 0
DO JS = 1, JS
CALL AI(IT, J, TI)
IF(T1.gt.T2) GO TO 120
120 CONTINUE
IT = 1
JS = 0
DO JS = 1, JS
Call AI(IT, J, TI)
IF(T1.gt.T2) GO TO 120
120 CONTINUE
IT = 1
JS = 0
DO JS = 1, JS
Call AI(IT, J, TI)
IF(T1.gt.T2) GO TO 120
120 CONTINUE
IT = 1
JS = 0
DO JS = 1, JS
Call AI(IT, J, TI)
IF(T1.gt.T2) GO TO 120
120 CONTINUE
IT = 1
JS = 0
DO JS = 1, JS
Call AI(IT, J, TI)
IF(T1.gt.T2) GO TO 120
120 CONTINUE
IT = 1
JS = 0
DO JS = 1, JS
Call AI(IT, J, TI)
IF(T1.gt.T2) GO TO 120
120 CONTINUE
IT = 1
JS = 0
DO JS = 1, JS
Call AI(IT, J, TI)
IF(T1.gt.T2) GO TO 120
120 CONTINUE
IT = 1
JS = 0
DO JS = 1, JS
Call AI(IT, J, TI)
IF(T1.gt.T2) GO TO 120
120 CONTINUE
IT = 1
JS = 0
DO JS = 1, JS
Call AI(IT, J, TI)
UNCLASSIFIED

```
YTED.
DO 140 IS=1,MM
CALL A*(E+S,TT)
E=TT
IF(TT.LT.EPS) GO TO 14A
IF(D(1),LT,0,) GO TO 140
YT1E(T1+1)/TT
IF(T2E,VT,TT,AND,VT,ST,=0.), GO TO 143
YT2E
14A CONTINUE
E(I)=LQ,0) .SWITY
CALL A*(E+J,TT)
E(N)=0
GO TO 300
200 CONTINUE
XEM.
GO ZIEEJM,NN
CALL A*(E+J,TT)
IF(11,LT,1*EPS) GO TO 210
XT=TT
XJM.
210 CONTINUE
I(I)=LQ.0) GO TO 400
E(N)=0
IEM.
XTH,0)
CALL A*(E+J,TT)
E(I)=11
IF(TT.LT.EPS) GO TO 220
XTN(E+1,,TT)
IF(2E,VT,TT,AND,VT,ST,=0.), GO TO 220
YT2E
15M.
220 CONTINUE
I(I)=LQ.0) GO TO 600
300 CONTINUE
J(A(I,0,0)) .SWITY
X(I=1,j)
DO 510 IA1,MM
E(I)=E(I)/TT
510 CONTINUE
E(I)=LQ,0) .SWITY
DO 330 IA1,MM
IF(E(I),LT,1) GO TO 330
EI(I)=E(I)+EI(I)+EI(I)
DO 320 01,MM
EI(I)=E(I)+EI(I)+EI(I)
320 CONTINUE
330 CONTINUE
E(I)=E(I)+E(I)
DO 340 IA1,MM
EI(I)=EI(I)+EI(I)
340 CONTINUE
DO 350 E=IA1,MM
EI(I)=EI(I)
```

UNCLASSIFIED
350 CONTINUE
  IF(FLG.EQ.1) GO TO 800
  GO TO 100

C  SOLUTION

400 CONTINUE
  IF(D(1) .GE. 0.) RETURN
  900 WRITE(*,1991)
    GO TO 160
  900 WRITE(*,1992)
  700 CONTINUE
  1001 FORMAT(*) PROBLEM INFEASIBLE!
  1002 FORMAT(*) PROBLEM UNBOUNDED!
  END
UNCLASSIFIED

```
SUBROUTINE OUTMAT
COMMON /MLAA/ B(5), C(20), H(4), I(4), J(4), T(4), Z(4)
COMMON /MLAB/ N(4), X(4), Y(4), Z(4)

DIMENSION IN(30), IM(30), I1, I1

LCM
DO 110 I1=1,N
IF(MOD(IN(I1),EQ,8)) WRITE(0,YDB)(J,J=1,NO)(J,J=1,NT)

LCM=C
DO 120 J1=1,N
CALL AMAT(J1,J1)

CONTINUE
110 CONTINUE
DO 120 J1=1,N

CONTINUE
DO 160 L1=1,N

IF(IN(I1)=0) WRITE(0,YDB)(J,J=1,NO)(J,J=1,NT)

CONTINUE
DO 160 L1=1,N

I1=.10*(IN(I1)+1)

RETURN
900 FORMAT (15X,6X,15X,2X,6X,6X,6X)
910 FORMAT (15X,6X,15X,2X,6X,6X,6X)
920 FORMAT (15X,6X,15X,2X,6X,6X,6X)
930 FORMAT (15X,6X,15X,2X,6X,6X,6X)
END
```
UNCLASSIFIED

SUMMOIT NE OUTPJT  
CAMMO N #RL*A/ S"-l ) ,C5 ,00i'

COMMON /RL1/ *(72),,AIR((10),4),E(56001),X=W(120001),T(200001)
DIMENSION HEAD(5,9)
DIMENSION ADE(50),AXD(50),AYU(50),XFM(50),XCHP(50)
DIMENSION JGQT(10),LYNK(51+1+1+1+1+10+100)
INTEGR HOGS/NAD
LOGICAL PURUSOCNV,INAT,VAR+5,N40
DATA MIAOIZHAX .4HVALU,4P(( nA,I$M101,INO,241N,ANWCAP,la 4V 45 - 4HJSO,
1 1$ •2NI aI,SH MEtIA. SMTSIN S,4W u;f.IWO.1NIN .AH CMr '4NJ5E0 ,
2 114 .114 ?WIN .,WL.I' .,Hlj%rO,TH •)N /
900 FORMAT(I14,PROOFlN..U,1 n,iT14*L OLUT$Uq FOR N.,Aj,.ZNUII
901 FORMAA(.I2,ZX,P4,211,21A4,,12.2,j(r)2.2.F5.1)))
902 FORMAT(.5I.'TOTALS.,F20.2.I(F1..0.2.I$.PUi9k,FI1.i.. ')
903 FORMAT(5,,/bOX, '1EAPON 0LSU(/)
907 FORMAT(71(.J2.ZX.pA4,211,A4.~ (FlL).l,FS.1),FI0.4)
921 FORMIT IRF *0.5 )
922 FORMAT( t2 , FS.2,F1*.*, 4~ IZ)
930 FORMAT*) THE RELATIVE
959 FORMAII.1',-3911,SSENSITI VJI,
C

UNCLASSIFIED
UNCLASSIFIED

WRITE(6,930)

INITIALIZE ROUTINE

999 ZWP=0.
ZIL=0.
ZEMT=0.
ZCHP=0.
ZKV = 0.
ZYK = 0.
ICNT=0
DO 19Q I=1,IM
H(I)=0.
1990 CONTINUE
DO 1320 J=1,M
JJA=J(A(J))
IF(JJA.GT.NTV) GO TO 1326
DO 1310 I=1,IM
CALL MAT(I,JJA,A)
IF(A.EQ.0.) GO TO 1310
H(I)=H(I)+A(A(J))
1310 CONTINUE
CONTINUE
1320 CONTINUE

INTERPRET SOLUTION JA+KA

DO 1007 I=1,M
IF SLACK VARIABLE GO TO 1005
IF(JA(JA(J),ST,NTV)) GO TO 1005
IF(KA(JA(J),ST,GA(I),ICNT+1)
JQ(JA(J))
CONTINUE
1001 CONTINUE

FIND TARGET NUMBER

J=0
1003 J=J+1
Q(NW)=J
CALL MAT(JA(J),JQ,G)
IF(JA(J).EQ.0.) GO TO 1001
J=J(T(J))

CALCULATE DE

Y=OBJECT(JA(J),J,PS)
XDE(I)=(I+PS)*Y(T(J))

INITIALIZE VECTOR PARAMETERS

KWP(I)=0.
KYL(I)=0.
REM(I)=0.
XCHP(I)=0.

CALCULATE VALUE OF VECTOR PARAMETERS

DO 1002 J=1,IM

C-26

UNCLASSIFIED
CALL ANAT(J,JQ,A)
IF(A_EW.0) GO TO 1007
XYP(J) =XYP(J)+A
XYL(J) =XYL(J)+AYYL(J)
XENT(J) =XENT(J)+AXENT(J)
XCP(J) =XCP(J)+AXCP(J)

1006 CONTINUE

** SIN SOLUTION VALUES **

XYP(J) =XYP(J)+A
XYL(J) =XYL(J)+AYYL(J)
XENT(J) =XENT(J)+AXENT(J)
XCP(J) =XCP(J)+AXCP(J)
XK(J) =XK(J)+AXK(J)

1007 IF XLYL(J) > 0 THEN GO TO 1007

1008 JTGY(J)= JTYA(J)
1009 IF JTGY(J) LT 0 THEN GO TO 1010
1010 IF JTGY(J) LT 0 THEN JTGY(J) = 0
1011 CONTINUE

** LINK TARGETS **

DO 1540 J=1,NT
1540 YXPN(J) = YXPN(J) + YP(J)
1541 YXL(J) = YXL(J) + YL(J)
1542 YENT(J) = YENT(J) + YT(J)
1543 YCP(J) = YCP(J) + YC(J)
1544 JL=JL+1
1545 JTGY(J) = JTGY(J) + 1
1546 IF (JTGY(J) EQ 0) THEN GO TO 1007

** UNCLASSIFIED **

1547 CONTINUE

C-27
1060 \( w(j) = 0 \)
\( YDE = 0 \)
\( K = \text{IP}() \)
1070 \( IFIX^2(K), EQ, 0) \) GO TO 1670
\( J = \text{JA}(X) \)
DO 1670 \( J = 1, NW \)
CALL AMAT(J, JQA)
\( MA(J) = \text{IFIX}(A, 0, 5) \)
1070 CONTINUE
WRITE(*, 910) \( XA(K), \psi DE(K) \times (MA(J) \times J + 1 \times NW) \)
\( YDE = VU_{X}X(K) \times \psi DE(K) \)
DO \( 1280 \) \( J = 1, NW \)
CALL AMAT(J, JQA)
\( H(J) = H(J) + \phi XA(K) \)
1080 CONTINUE
1090 \( K = \text{LINK}(K) \)
IF \((K \neq 81, 0) \) GO TO 1670
IF \((\text{NUM}^2(I) \times \text{EQ} = 0) \) \( YDE = 1 \)
IF \((\text{NUM}^2(I) \times \text{GT} = 0) \) \( YDE = \psi DE/\text{NUM}(I) \)
DO \( 1180 \) \( J = 1, NW \)
\( \text{IM}^2(J) = \text{IFIX}(\text{IM}(J) + 0, 5) \)
1100 CONTINUE
\( \text{NUM}^2(I) \)
1110 WRITE(*, 911) \( \phi X YDE \times (\text{IM}(J) \times J + 1 \times NW) \)
RETURN
END
SUBROUTINE DUAL
COMMON /BLKA/ B(I), C(I)
COMMON /BLKC/ JA(S1), KA(S1)
WRITE*(9,900)
10 M=M+1
DO 110 T1=1, M1
110 WRITE*(9,900)I,J1(J1),X1(J1)
RETURN
900 FORMAT(1l1)
904 FORMAT(1x,216,F15.6)
END

UNCLASSIFIED
SUBROUTINE AMAT(I,J,A)
COMMON /BLRA/ B(1:1), C(2001:2001)
COMMON /BLKB/ N,N, N,N
COMMON /BLKT/ JSRT(20), JSRT(20)
ANB.
IF(I.LE.NK+ANDJ.LE.NV) GO TO 100
IF(J.HI.NI) GO TO 20
IF(J.GT.NI) GO TO 10
IF(I.GT.NI) GO TO 40
IF(I.GT.NK) RETURN
IF(J.EQ.JV) RETURN
10 CONTINUE
ANC(I,J)
RETURN
20 CONTINUE
IF(J.NE.NV) GO TO 30
IF(J.EQ.NJ) ANJ(J)
RETURN
30 CONTINUE
IF(J.GT.NJ) ANJ(J)
IF(J.EQ.NJ) ANJ(J)
RETURN
40 CONTINUE
IF(I.GT.IJ+JSRT(IJ+1)+J+J) AND (J+J+J.LE.EXT(IJ+1)+J) AND (J+J.LE.EXT(IJ+1)+J) AND (J+J.LE.EXT(IJ+1)+J) AND (J+J.LE.EXT(IJ+1)+J)
RETURN
50 CONTINUE
IF(J.GT.JV) RETURN
RETURN
100 CONTINUE
IF(J.EQ.NK+I) GO TO 200
IF(J.EQ.NI+I) ANA(I,J)
RETURN
200 CONTINUE
IF(J.EQ.IA(I,J)) GO TO 300
IF(J.EQ.IA(IA(I,J)+I)) ANA(I,J)
IF(J.EQ.IA(IA(I,J)+I)+I) ANA(I,J)
IF(J.EQ.IA(I,J)+I) ANA(I,J)
RETURN
300 CONTINUE
ANJ(J)
RETURN
END
UNCLASSIFIED

SURROUNTE AX(I,J,TT)
COMMON A(51,51,1),M1
TTM4
DO 10 K=1,N1
CALL AMAT(K,SA)
TTM4=M1(KA)
10 CONTINUE
RETURN
END