PAPER P-1135

A TARGETING MODEL THAT MINIMIZES COLLATERAL DAMAGE

Jeffrey H. Grotte

December 1975

INSTITUTE FOR DEFENSE ANALYSES
PROGRAM ANALYSIS DIVISION
The work reported in this document was conducted under Contract DAHC15 73 C 0200 for the Weapons Systems Evaluation Group. The publication of this IDA Paper does not indicate endorsement by the Weapons Systems Evaluation Group, nor should the contents be construed as reflecting the official position of that agency.

Approved for public release; distribution unlimited.
A TARGETING MODEL THAT MINIMIZES COLLATERAL DAMAGE

Jeffrey H. Grotte

December 1975

INSTITUTE FOR DEFENSE ANALYSES
PROGRAM ANALYSIS DIVISION
400 Army-Navy Drive, Arlington, Virginia 22202

Contract DAHC15 73 C 0200
Task T-237
A Targeting Model That Minimizes Collateral Damage

Jeffrey H. Grotte

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

Weapons Systems Evaluation Group
400 Army-Navy Drive, Arlington, Virginia 22202

Approved for public release; distribution unlimited.

Collateral Damage, Targeting Model, Lexicographic Enumeration, Nonconvex Programming, Integer Programming, Lawler-Bell Algorithm

This paper considers the problem of allocating weapons to achieve targeting objectives while simultaneously minimizing the aggregate damage to surrounding nonmilitary facilities, each of which has an upper limit to the damage it is permitted to incur. This problem is, in general, nonconvex. A model is formulated that assumes only that damage to individual targets or associated nonmilitary facilities does not decrease as the number of allocated weapons increases. An implicit
20. continued

enumeration algorithm, based on that of Lawler and Bell, is described that yields optimal integer solutions. An example is presented.
CONTENTS

I. MATHEMATICAL FORMULATION .......................... 1
II. AN ALGORITHM ..................................... 3
III. A CLASS OF EXAMPLES ............................... 5
IV. COMPUTER APPLICATIONS ............................. 8
REFERENCES ........................................... 11

APPENDIX

FORTRAN LISTING AND INPUT SPECIFICATIONS
This paper considers the problem of allocating weapons to achieve targeting objectives while simultaneously minimizing aggregate damage to surrounding nonmilitary facilities, each of which has an upper limit to the damage it is permitted to incur. A model is formulated which assumes only that damage to individual targets or associated facilities does not decrease as the number of allocated weapons increases. An implicit enumeration algorithm, based on that of Lawler and Bell (see Reference [3]), is described that yields optimal integer solutions. An example is presented.

This paper differs from IDA paper P-1106 (Reference [2]) in that it presents the full generality of the collateral damage minimizing model, whereas P-1106 describes a model (NDM) tailored to specific design requirements. In addition, the code listed in the Appendix may prove a prototype for a modified NDM with greatly decreased run times.
A TARGETING MODEL THAT MINIMIZES COLLATERAL DAMAGE

One of the assumptions behind the argument to employ counterforce targeting of strategic weapons (the targeting of an enemy's strategic capability), as opposed to countervalue targeting (the objective of which is the destruction of population and economy), is that sufficient destruction of strategic targets can be achieved without causing appreciable damage to the surrounding nonstrategic facilities. This paper presents a model which addresses the following two questions: Given a collection of weapons, potential aimpoints, and a configuration of strategic targets—each being assigned a minimum level of damage; and nonstrategic facilities—each having a maximum level of permissible damage,

(A) Is there an assignment of weapons to aimpoints (an allocation) that satisfies the above two sets of constraints?

(B) Of all allocations satisfying the above two sets of constraints, what is the one (or a one) that minimizes the (perhaps weighted) sum of the damage to the nonstrategic facilities?

I. MATHEMATICAL FORMULATION

The fundamental elements of the model are M strategic targets, henceforth called simply "targets," N nonstrategic facilities, or "nontargets," I different weapon types, and J potential aimpoints to which any weapon can be directed. An allocation is the matrix \( z_{i,j} \) for \( i=1,\ldots,I; j=1,\ldots,J \) where \( z_{i,j} \), an integer, is the number of weapons of type i allocated to aimpoint j.
For each target $m$, we suppose a real-valued response function $f_m(z)$ which represents the damage to target $m$ from allocation $z$. We require that $f_m(z)$ be monotonically non-decreasing in each component of $z$, which is an implicit assumption that, given any allocation, the allocation of additional weapons does not result in less damage to any target. Each target $m$ is assigned a real number, $c_m$, which is the minimum damage requirement (targeting objective), i.e., for an allocation $z$ to be feasible, it must satisfy $f_m(z) > c_m$, $m=1,\ldots,M$.

Similarly, for each nontarget $n$ there is a response function $g_n(z)$, monotonically nondecreasing in each component of $z$, and a real number $d_n$ denoting the maximum damage permitted to this nontarget. Further, each nontarget $n$ is assigned a nonnegative weight, or value, $\lambda_n$.

The nonnegative integer $w_i$ is the number of weapons of type $i$ available for allocation.

We can now combine questions (A) and (B) into the following problem $P$:

\[
\min_{z} \sum_{n=1}^{N} \lambda_n g_n(z) \quad \text{subject to} \quad \begin{align*}
    f_m(z) &> c_m \quad m=1,\ldots,M ; \\
    g_n(z) &< d_n \quad n=1,\ldots,N ; \\
    \sum_{j=1}^{J} z_{i,j} &\leq w_i \quad i=1,\ldots,I ; \\
    z_{i,j} &\in \mathbb{Z}^+ \quad i=1,\ldots,I; \ j=1,\ldots,J ;
\end{align*}
\]

where $\mathbb{Z}^+$ is the set of nonnegative integers. If problem $P$ is infeasible, then the answer to question (A) is clearly "no," otherwise an answer to question (B) is ensured because the number of allocations which satisfy constraints (4) and (5) is finite.
II. AN ALGORITHM

Problem P admits solution by implicit enumeration. The following algorithm is based upon the lexicographic technique of Lawler and Bell (see Reference [3])—though, unlike the Lawler-Bell approach, this algorithm does not use binary vectors. We first identify the matrix \( \mathbf{z} \) with a vector \( \hat{\mathbf{z}} \). This can be done in a number of ways, one of which is through the following relationship:

\[
\hat{z}_k = z_{i,j}, \quad k = i + (j-1) \cdot I; \ i=1,...,I; \ j=1,...,J. \tag{6}
\]

Note that this can be reversed as follows:

\[
z_{i,j} = \hat{z}_k, \quad i = k - \langle \frac{k-1}{I} \rangle \cdot I, \quad j = \langle \frac{k-1}{I} \rangle + 1; \quad k=1,...,K=I \cdot J.
\]

where \( \langle x \rangle \) is the largest integer less than or equal to \( x \). With this in mind, we will drop the circumflex, and in the discussion that follows, all allocations will be vectors in \( \mathbb{Z}^+_K \), i.e., \( K \)-dimensional vectors with nonnegative integer components.

We require two binary relations between vectors in \( \mathbb{Z}^+_K \):

**Componentwise (partial) Ordering:**

We write \( \hat{x} \succeq \hat{y} \) if \( x_k \geq y_k \) for \( k=1,...,K \)

\( \hat{x} \succeq \hat{y} \) if \( \hat{x} \succeq \hat{y} \) and \( x_k > y_k \) for at least one \( k \).

**Lexicographic Ordering:**

We write \( \hat{x} \prec \hat{y} \) if \( x_{k'} > y_{k'} \) where \( k' = \max \{ k | x_k \neq y_k \} \), \( 1 \leq k < K \)

and \( \hat{x} \succeq \hat{y} \) if \( \hat{x} \prec \hat{y} \) or \( \hat{x} = \hat{y} \).

Let \( \mathcal{F} = \{ \hat{z} \in \mathbb{Z}^+_K | \hat{z} \leq \hat{w} \} \) for \( k=1,...,K, \ h=k-\langle \frac{k-1}{I} \rangle \cdot I \} \)

Thus \( \mathcal{F} \) is a set of allocations that satisfy constraint (5) of problem P, and clearly contains all allocations that satisfy constraint (4), and so must contain all solutions to problem P providing problem P is feasible. Since \( \succeq \) totally orders \( \mathcal{F} \), we could enumerate all the points of \( \mathcal{F} \) and find the solution to P in this manner. However, the monotonicity of the
objective and constraint functions will permit us to skip over many infeasible and/or nonoptimal points. To see this, we need some notation. Consider a vector $z \in \mathcal{F}$. We will denote by $z+1$ the vector $x$, if it exists, satisfying

$$
\begin{align*}
    x &\in \mathcal{F} \\
    z &\geq z \\
    y &> z \
\end{align*}
$$

At most one such vector exists, but may fail to exist because of the boundedness of $\mathcal{F}$. The vector $z-1$ will be that vector $x$, if it exists, satisfying

$$
\begin{align*}
    x &\in \mathcal{F} \\
    z &\geq z \\
    y &> z \\
\end{align*}
$$

This vector will always exist except for $z = 0$. The vector $z^*$ will be that $x$, if it exists, satisfying

$$
\begin{align*}
    x &\in \mathcal{F} \\
    x &\geq z \\
    x &\neq z \\
    (y \geq z) \land (y \neq z) &\Rightarrow y \geq z
\end{align*}
$$

Intuitively, $z^*$ is the first vector in $\mathcal{F}$ following $z$ (in the lexicographic ordering) which is not (componentwise) greater than or equal to $z$. For some $z$, $z^*$ may not exist; however, we will adopt the following convention: For any $z$ for which $z^*$ does not exist, we will set

$$
(z^*-1)_k = w_h \text{ for } h=k-\frac{k-1}{I} \text{, } I, k=1,\ldots,K,
$$

thereby ensuring that $z^*-1$ exists for every $z \in \mathcal{F}$. Crucial to the algorithm is the observation that for any $z \in \mathcal{F}$, any $y$ that satisfies $z \leq y \leq z^*-1$ also satisfies $y \geq z$. 


Figure 1 outlines the fundamentals of the algorithm. A brief inspection of the flow chart will make clear that the algorithm must terminate after a finite number of steps. If $H = \infty$ upon termination, the problem is infeasible, otherwise an optimum integer allocation will always be found. The order in which the constraints are examined was chosen because, for certain applications, this order was efficient. However, we make no claim that this is, in any sense, an optimal ordering. For other applications, a different sequence of constraint evaluations might well prove to be better.

III. A CLASS OF EXAMPLES

We will now look at a class of examples with point targets and nontargets, where the destruction of any target or nontarget is a binomial random variable with probability of kill dependent on the allocation, but with independent weapons effects. We will use Cartesian coordinates to specify location, in particular, target coordinates are $(x_m, y_m)$, $m=1,...,M$; nontarget coordinates are $(\mu_n, \nu_n)$, $n=1,...,N$; and aimpoint coordinates are $(\xi_j, \eta_j)$, $j=1,...,J$. For response functions we will employ "probability of kill" which is computed as follows: Let $p_{i,j}^m$ be the probability that a single weapon of type $i$, allocated to aimpoint $j$, destroys target $m$, conditioned on the weapon's arrival at its destination. The probability that a type-$i$ weapon arrives at its destination, its "reliability," is given by $\rho_i$. Because we have assumed independence of weapon effects, it is not difficult to compute the total probability that target $m$ is destroyed by allocation $z$, which is

$$f_m(z) = 1 - \prod_{i=1}^{I} \prod_{j=1}^{J} (1 - \rho_i p_{i,j}^m)^{z_{i,j}}.$$

Similarly, we denote by $p_{i,j}^n$ the conditional probability that a single type-$i$ weapon allocated to aimpoint $j$ destroys
Figure 1. AN IMPLICIT ENUMERATION ALGORITHM
nontarget \( n \). Therefore, the probability that allocation \( z \)
destroys nontarget \( n \) is

\[
g_n(z) = 1 - \prod_{i=1}^{I} \prod_{j=1}^{J} (1 - p_{i,j}^{n})^{z_{i,j}}.
\]

Although the values of the parameters \( \{p_{i,j}^{m}\} \) and \( \{p_{i,j}^{n}\} \)
can be entirely arbitrary, within the obvious limits

\[
0 \leq p_{i,j}^{m} \leq 1 \quad m=1,\ldots,M; \quad i=1,\ldots,I; \quad j=1,\ldots,J,
\]

\[
0 \leq p_{i,j}^{n} \leq 1 \quad n=1,\ldots,N; \quad i=1,\ldots,I; \quad j=1,\ldots,J,
\]

we will use, for tutorial purposes, the following formulae,
which are not unreasonable approximations to certain types of
weapon damage curves and have been proposed by other analysts
(see, for example, Eckler, Reference [1], or McNolty, Reference
[4]):

\[
p_{i,j}^{m} = \exp \left\{ -\alpha_{i,m} \left[ (x_{m} - \xi_{j})^{2} + (y_{m} - \zeta_{j})^{2} \right] \right\} \quad m=1,\ldots,M; \quad i=1,\ldots,I; \quad j=1,\ldots,J
\]

\[
p_{i,j}^{n} = \exp \left\{ -\beta_{i,n} \left[ (u_{n} - \xi_{j})^{2} + (v_{n} - \zeta_{j})^{2} \right] \right\} \quad n=1,\ldots,N; \quad i=1,\ldots,I; \quad j=1,\ldots,J
\]

where all \( \alpha_{i,m}, \beta_{i,n} \) are nonnegative real numbers. The param-
eters \( \{\alpha_{i,m}\} \) and \( \{\beta_{i,n}\} \) are measures of the rate at which weapon
effects decrease with distance.

With these conventions, we can now write explicitly the
problem \( P' \) which comprises this class of examples:

\[ P': \text{ Given nonnegative weights } \lambda_{n}, \quad n=1,\ldots,N, \text{ and the values of } \]

\[
c_{m} \in [0,1] \quad m=1,\ldots,M
\]

\[
d_{n} \in [0,1] \quad n=1,\ldots,N
\]

\[
w_{i} \in \mathbb{Z}^{+} \quad i=1,\ldots,I
\]

\[
\rho_{i} \in [0,1] \quad i=1,\ldots,I
\]
\[ \begin{align*}
\alpha_{i,m} & \geq 0 & i=1,\ldots,I; & m=1,\ldots,M \\
\beta_{i,n} & \geq 0 & i=1,\ldots,I; & n=1,\ldots,N \\
x_m, y_m & & m=1,\ldots,M \\
\nu_n, \nu_j & & n=1,\ldots,N \\
\xi_j, \zeta_j & & j=1,\ldots,J
\end{align*} \]

\[
\text{minimize } h(z) = \sum_{n=1}^{N} \left\{ \prod_{i=1}^{I} \prod_{j=1}^{J} \left( 1 - \rho_{i,j} \exp \left\{ -\beta_{i,n} \left[ (v_n - \xi_j)^2 + (\nu_n - \zeta_j)^2 \right] \right\} \right)^{z_{i,j}} \right\}
\]

subject to

\[
f_m(z) = 1 - \prod_{i=1}^{I} \prod_{j=1}^{J} \left( 1 - \rho_{i,j} \exp \left\{ -\beta_{i,n} \left[ (v_n - \xi_j)^2 + (\nu_n - \zeta_j)^2 \right] \right\} \right)^{z_{i,j}} \geq c_m
\]

\[
g_n(z) = 1 - \prod_{i=1}^{I} \prod_{j=1}^{J} \left( 1 - \rho_{i,j} \exp \left\{ -\beta_{i,n} \left[ (v_n - \xi_j)^2 + (\nu_n - \zeta_j)^2 \right] \right\} \right)^{z_{i,j}} \leq d_n
\]

\[
\sum_{j=1}^{J} z_{i,j} \leq w_i & \quad i=1,\ldots,I \\
z_{i,j} \in \mathbb{Z}^+ & \quad i=1,\ldots,I; j=1,\ldots,J
\]

IV. COMPUTER APPLICATIONS

A FORTRAN routine to solve problems of the type given by \( P \) was written for the CDC 6400 computer, and was used to solve the numerical example of this section. (A listing of this program, along with input formats are given in the Appendix.) The values of the parameters are listed in Tables 1-6. The configuration of the targets, nontargets and aimpoints is depicted in Figure 2.

The routine ran for five seconds to compute the optimal solution, \( \hat{z} \), given in Table 7.
### Table 1. TARGET PARAMETERS

<table>
<thead>
<tr>
<th>m</th>
<th>x_m</th>
<th>y_m</th>
<th>c_m</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>.8</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>.8</td>
</tr>
</tbody>
</table>

### Table 2. NONTARGET PARAMETERS

<table>
<thead>
<tr>
<th>n</th>
<th>u_n</th>
<th>v_n</th>
<th>l_n</th>
<th>d_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>.3</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>4</td>
<td>.3</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>.3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0</td>
<td>8</td>
<td>.3</td>
</tr>
</tbody>
</table>

### Table 3. AIMPOINT PARAMETERS

<table>
<thead>
<tr>
<th>J</th>
<th>ξ_j</th>
<th>η_j</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

### Table 4. WEAPON PARAMETERS

<table>
<thead>
<tr>
<th>ρ</th>
<th>w_ρ</th>
<th>P_ρ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>.6</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>.9</td>
</tr>
</tbody>
</table>

### Table 5. COMPONENTS OF μ

<table>
<thead>
<tr>
<th>m</th>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.1</td>
<td>.8</td>
</tr>
<tr>
<td>2</td>
<td>.5</td>
<td>.3</td>
</tr>
</tbody>
</table>

### Table 6. COMPONENTS OF β

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.05</td>
<td>.1</td>
<td>.1</td>
<td>.09</td>
</tr>
<tr>
<td>2</td>
<td>.8</td>
<td>.8</td>
<td>.8</td>
<td>.8</td>
</tr>
</tbody>
</table>

### Table 7. OPTIMAL ALLOCATION \( \hat{z} \)

<table>
<thead>
<tr>
<th>J</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ h(\hat{z}) = 5.2 \]
\[ g_1(\hat{z}) = .28 \]
\[ f_1(\hat{z}) = .83 \]
\[ g_2(\hat{z}) = .24 \]
\[ f_2(\hat{z}) = .83 \]
\[ g_3(\hat{z}) = .24 \]
\[ f_3(\hat{z}) = .83 \]
\[ g_4(\hat{z}) = .28 \]

### Table 8. OPTIMAL ALLOCATION \( \hat{z}' \)

<table>
<thead>
<tr>
<th>J</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ h(\hat{z}') = 4.5 \]
\[ g_1(\hat{z}') = .08 \]
\[ f_1(\hat{z}') = .81 \]
\[ g_2(\hat{z}') = .37 \]
\[ f_2(\hat{z}') = .82 \]
\[ g_3(\hat{z}') = .37 \]
\[ g_4(\hat{z}') = .08 \]
It is interesting to note that if all the $d_i$ are changed to 1.0, which is equivalent to removing the individual non-target damage constraints, then the optimal allocation is $\mathbf{z}^*$, given in Table 8. In this latter case, we have reduced total collateral damage over that given in Table 7, but only at the expense of considerably greater damage to two of the nontargets.
REFERENCES


APPENDIX

FORTRAN LISTING AND INPUT SPECIFICATIONS
FORTRAN LISTING

PROGRAM MDLTWO(INPUT,OUTPUT)
COMMON/LIMITS/XTAR,NONTAR,XNONTAR,YNONTAR,XAIM,YAIM,NWEAP
COMMON/TARGETS/XTAR(10),YSTAR(10),DEST(10)
COMMON/NONTAR/XNON(10),YNON(10),FACTOR(10),UPNOND(10)
COMMON/AIMPNTS/XAIM(100),YAIM(100)
COMMON/EP/EWNUM(10),RELB(10),EFFSTAR(10,10),EFFNON(10,10)
COMMON/SCRATCH/PKT(10,10,100),PKN(10,10,100),IPRESAL(1000)
COMMON/AMT/ITARSURV(10),IBTNAL(1000),BTNTS(10),NTV(10),SV,IFLAG,NFLAG,BTNNTS
CALL READIT
CALL CALCPRB
CALL LEADEX
CALL OUT
END
SUBROUTINE READIT
COMMON/LIMITS/NOTAR,NONON,NOAIM,NOWEAP
COMMON/TARGETS/XTAR(10),YTAR(10),DEST(10)
COMMON/NONTAR/XNON(10),YNON(10),FACTOR(10),UPNON(10)
COMMON/IMPNTS/XAIM(100),YAIM(100)
COMMON/EP/EPLNUM(10),REBL(10),EFFTAR(10),EFFNON(10),PRESAL(1000),
TARSURV(10),IBTNTV(1000),BINT(10),NTV(10),SV,IFLAG,NFLAG,BTNNTV
2,BTNNTV(10),NONFLAG
READ1,NOTAR,NONON,NOAIM,NOWEAP
1 FORMAT(4/10)
DO 5 I=1,NOTAR
READ10+XAIM(I),YAIM(I),DEST(I)
5 CONTINUE
DO 15 I=1,NONON
READ20+XNON(I),YNON(I),FACTOR(I),UPNON(I)
15 CONTINUE
DO 25 I=1,NOAIM
READ30+XAIM(I),YAIM(I)
25 CONTINUE
DO 40 I=1,NOWEAP
READ40+XNUM(I),RELBL(I)
40 CONTINUE
READ200,(EFFTAR(I),M),NOTAR
READ300,(EFFNON(I),N),NONON
200 CONTINUE
FORMAT(8F10.6/2F10.6)
300 FORMAT(8F10.6/2F10.6)
100 CONTINUE
RETURN
END
SUBROUTINE CALCPRB
COMMON/LIMITS/NOTAR, NONON, NOAIM, NOWEAP
COMMON/TARGETS/XTAR(10), YTAR(10), DEST(10)
COMMON/NONTAR/XNON(10), YNON(10), FACTOR(10), UPNOND(10)
COMMON/AIMPNTS/XAIM(100), YAIM(100)
COMMON/E*EP/IWNUM(10), RELBL(10), EFFTAR(10), EFFN0N(10), 10
COMMON/SCRATCH/PKT(10, 10, 100), PKN(10, 10, 100), IPRESAL(1000),
ITARSURV(10), IBTNAL(1000), BTNMTS(10), NTV(10), SV, IFLAG, NFLAG, BTNNTS
COMMON/SLRATCM/PKT(10, 10, 100), PKN(10, 10, 100),
BTNNTS(10), IBTNAL(1000), BTNMTS(10), NTV(10), SV, IFLAG, NFLAG, BTNNTS
2*BTNNTS(10), NONFLAG
D010 M = 1, NOTAR
D010 I = 1, NOWEAP
D010 J = 1, NOAIM
WW = EFFTAR(M) + ((XAIM(J) - XTAR(M)) + (YAIM(J) - YTAR(M)) + 2)
PKT(M + J) = RELBL(I) * EXP(-WW)
10 CONTINUE
D020 M = 1, NONON
D020 I = 1, NOWEAP
D020 J = 1, NOAIM
WW = EFFN0N(M) + ((XAIM(J) - YNON(N)) + (YAIM(J) - YNON(N)) + 2)
PKN(N + J) = RELBL(I) * EXP(-WW)
20 CONTINUE
RETURN
END
SUBROUTINE LEXO

THIS SUBROUTINE FINDS AND STORES THE OPTIMAL ALLOCATION USING
LEXICOGRAPHIC ENUMERATION AFTER LAWLER-BELL. OPTIMAL VALUES ARE
STORED AS FOLLOWS—

BTNNTS = TOTAL NONTARGET SURVIVAL LEVEL (FROM NTS)
IBTNAL = OPTIMAL ALLOCATION
BTNNTS() = RESULTING TARGET SURVIVAL LEVEL (FROM TAR)
BTNNTV() = INDIVIDUAL NONTARGET SURVIVAL LEVELS (FROM NTS)

COMMON/LIMITS/NOTAR, NONON, NOAIM, NOEAP
COMMON/TARGETS/TAR(10), YTR(10), DEST(10)
COMMON/NONTAR/XNON(10), YNON(10), FACTOR(10), UPNOND(10)
COMMON/AIMPTS/XAIM(100), YAIM(100)
COMMON/EWEP/INUM(10), RELN(10), EFFTR(10), EFFNON(10), EST(10)
COMMON/SCRATCH/PK(10), PKN(10,100), IPRESAL(1000)
1TARSURV(10), IBTN(1000), BTNNTS(10), NTV(10), SV, IFLAG, NFLAG, BTNNTS
2*BTNNTV(10), NTV

REAL NTV
INTEGER ITEMPAL(1000)
INTEGER ITEMST(1000)
DO 9000 LL=1, M
ITEMST(LL)=0
9000 CONTINUE
BTNNTS=9999999999.
H=NOEAP, NOAIM
DO 1 K=1, M
IPRESAL(K)=0
1 CONTINUE
C BEGIN ENUMERATION
C CHECK ZERO VECTOR FOR FEASIBILITY
CALL TAR
IF(IFLAG NE 0) GO TO 100
C IF HERE, NO ADDITIONAL WEAPONS ARE NEEDED
8002 CALL NTS
IF(NONFLAG NE 0) RETURN
8003 BTNNTS=SV
DO 10 K=1, NONON
BTNNTV(K)=NTV(K)
10 CONTINUE
IBTN(1)=IPRESAL(K)
11 CONTINUE
DO 12 K=1, NOTAR
BTNNTS(K)=TARSURV(K)
12 CONTINUE
RETURN
C THIS SECTION COMPUTES NEXT ALLOCATION
310 DO 315 J=1, NOAIM
DO 315 I=1, NOEAP
KKK=J*(J-1)/2+I
IF(IPRESAL(KKK) LT INUM(I)) GO TO 320
IPRESAL(KKK) = 0
315 CONTINUE
C HERE IF IPRESAL WAS LAST ALLOCATION
RETURN
320 IPRESAL(KKK) = IPRESAL(KKK) + 1
369 CALL NUMS
IF(NFLG.EQ.1) GO TO 600
400 CALL TARS
   IF(NFLG.NE.1) GO TO 500
C MERE IF ALLOCATION INFEASIBLE
C STORE IPRESAL
100 DO 405 K=1,M
   ITEMPAL(K)=IPRESAL(K)
405 CONTINUE
C NOW TO COMPUTE IPRESALSTAR=1
DO 410 K=1,M
   IF(IPRESAL(K).EQ.0) GO TO 410
   IPRESAL(K)=0
410 GO TO 415
415 IF(K.GE.M) GO TO 420
   L=K+1
   DO 425 J=1,L-1
      IPRESAL(K)=0
      CONTINUE
425 CONTINUE
   GO TO 420
430 IPRESAL(K)=IPRESAL(K)+1
   GO TO 435
420 DO 440 I=1,N
   DO 445 J=1,N
      KK=I+(J-1)*N
      IF(IPRESAL(KK).LT.INUM(I)) GO TO 430
   440 CONTINUE
   GO TO 480
435 DO 445 J=1,N
   DO 445 I=1,N
      KK=I+(J-1)*N
      IF(IPRESAL(KK).NE.0) GO TO 450
   445 CONTINUE
   IPRESAL(KK)=IPRESAL(KK)-1
C NOW WE HAVE IPRESALSTAR = 1
480 DO 9005 LL=1,M
   IF(ITEMST(LL).NE.IPRESAL(LL)) GO TO 9010
9005 CONTINUE
   GO TO 9020
9010 DO 9015 LL=1,M
   ITEMST(LL)=IPRESAL(LL)
9015 CONTINUE
   CALL TARS
   IF(NFLG.NE.0) GO TO 310
C MERE IF IPRESALSTAR = 1 IS FEASIBLE
9020 CONTINUE
   DO 945 K=1,M
      IPRESAL(K)=ITEMPAL(K)
945 CONTINUE
   GO TO 310
500 CALL NT$NS
   IF(NONFLAG.NE.0) GO TO 600
8010 IF(SV.GE.BTNNTS) GO TO 600
C HAVE FOUND A NEW OPTIMUM
DO 510 K=1,M
ITNAl(K)=IPRESAL(K)
510 CONTINUE
DO 515 I=1,NOTAR
BTNNTS(I)=TARSURV(I)
515 CONTINUE
BTNNTS=5V
DO 520 I=1,NONON
BTNNTV(I)=NTV(I)
520 CONTINUE
C
SKIP TO IPRESALSTAR
600 DO 610 K=1,M
IF(IPRESAL(K) *EQ.0)GO TO 610
IPRESAL(K)=0
GO TO 620
610 CONTINUE
620 IF(K *GE. M) RETURN
L=K+1
DO 625 K=L,M
J=(K-1)/NOWEAP
K=NOWEAP*J
IF(IPRESAL(K) *LT.*INUM(I))GO TO 630
IPRESAL(K)=0
625 CONTINUE
RETURN
630 IPRESAL(K)=IPRESAL(K)+1
GO TO 399
END
SUBROUTINE TARS
COMMON/LIMITS/NOTAR, NONON, NOAIM, NOWEAP
COMMON/TARGETS/XTAR(10), YTAR(10), DEST(10)
COMMON/NONTAR/XNON(10), YNON(10), FACTOR(10), UPNON(10)
COMMON/AIMPTS/XAIM(100), YAIM(100)
COMMON/EWEP/NUM(10), RELBL(10), EFFTAR(10,10), EFFNON(10, 10)
COMMON/SRATCH/PKT(10, 10, 100), PKN(10, 10, 100), IPRESAL(1000),
ITARSURV(10), IBTNAL(1000), BTMTS(10), NTV(10), SV, IFLAG, MFLAG, BTNNTS
2*BTNTS(10), NONFLAG
DO 10 M=1, NOTAR
   TSURV=1.
   DO 50 I=1, NOWEAP
      DO 50 J=1, NOAIM
         KKK=I+(J-1)*NOWEAP
         IF (IPRESAL(KKK) .LE. 0) GO TO 50
         IF (PKT(M+I+J) .GE. 1) 8*9
   TSURV=0.
   GO TO 50
9    PS=(1.*PKT(M+I,J))**IPRESAL(KKK)
   TSURV=TSURV+PS
50   CONTINUE
C     ARE CONSTRAINTS SATISFIED
   WW=1.-DEST(M)
   IF (TSURV .LT. WW) 6*7
6   IFLAG=1
   RETURN
7   TSURV(M)=1.-TSURV
10  CONTINUE
   IFLAG=0
   RETURN
END
SUBROUTINE NTS
COMMON/LIMITS/NOTAR, NONON, NOAIM, NOWEAP
COMMON/TARGETS/XTAR(10), YTAR(10), DEST(10)
COMMON/NONTAR/XNON(10), YNON(10), FACTOR(10), UPNOND(10)
COMMON/AIMPTS/XAIM(100), YAIM(100)
COMMON/EPF/IWNUM(10), RELBL(10), EFFTAR(10*10), EFFNON(10*10)
COMMON/SCRATCH/PKT(10*10*100), PKN(10*10*100), IPRESAL(1000),
1TARSURV(10*10), IBTNA(1000), RMTS(10), NTV(10), SV, IFLAG, NFLAG, BTNNTS
2, BTNNTV(10), NNONFLAG
REAL NTV,
SV=0.,
D01000 N=1, NONON
TEMPSV=1.
D050 I=1, NOWEAP
D050 J=1, NOAIM
KKK=I+J-1, NOWEAP
IF(IPRESAL(KKK) .LE. N) GOTO 50
IF(PKN(N+I+J) .GE. 1) GOTO 9
8 TEMPSV=0.
GO TO 50
9 PS=(1.0-PKN(N+I+J)) * IPRESAL(KKK)
TEMPSV=TEMPSV+PS
50 CONTINUE
WW=1.0-TEMPSV
IF(WW .GT. UPNOND(N))7, 10
7 NONFLAG=1
RETURN
10 SV=SV+FACTOR(N) * WW
NTV(N)=FACTOR(N) * WW
1000 CONTINUE
NONFLAG=0
RETURN
END
SUBROUTINE NUMBS
COMMON/LIMITS/NOTAR,NONON,NOAIM,NOWEAP
COMMON/TARGETS/XTARGET(10),YTARGET(10),DEST(10)
COMMON/NOTAR/XNONON(10),YNONON(10),FACTOR(10),UPNONON(10)
COMMON/AMPNTS/XAMPNTS(100),YAMPNTS(100)
COMMON/EP/IWNUM(10),RELB(10),EFFTARGET(10),EFFNONON(10)
COMMON/SLATCH/PKT(10,10,100),PKN(10,10,100),IPRESAL(1000),

ITARSURV(10),IBTIVAL(1000),BTNNTS(10),NTV(10),SV,IFLAG,NFLAG,BTNNTS
2*BTNNTS(10),NONFLAG
DO1=1,*NOWEAP
ISUM=0
DO2=1,*NOWEAP
KKK=I-J-1,*NOWEAP
ISUM=IPRESAL(KKK)+ISUM
CONTINUE
IF(ISUM.LE.IWNUM(I))GOTO1
NFLAG=1
RETURN
CONTINUE
NFLAG=0
RETURN
END
SUBROUTINE OUT

COMMON/LIMITS/NOTAR, NONON, NOAIM, NOWEAP
COMMON/TARGETS/XTAR(10), YTAR(10), DEST(10)
COMMON/NONTAR/XNON(10), YNON(10), FACTOR(10), UPNOND(10)
COMMON/AIMPTS/XAIM(100), YAIM(100)
COMMON/EMEP/IWNUM(10), RELBL(10), EFFTAR(10), EFFNON(10, 10)
COMMON/SCRATCH/PKN(10, 10, 100), PKN(10, 10, 100), IPTESAL(1000),
1 TAVERS(10), IBTMTA(1000), BNTMS(10), NTV(10), SV, IFNLAG, NFLAG, BTVNFS
2, BTVNFS(10), NONFLAG

REAL NTV

PRINT 5 FORMAT(1H1, "PROGRAM MDL")
PRINT 10 FORMAT(1H1, "INPUT DATA")
PRINT 15 FORMAT(1H1, "TARGETS (*I4, *)")
PRINT 20 FORMAT(1H1, "TARGET NUMBER*, 6X*, X COORD.*, 7X*, Y COORD.*, 6X,
1*PROB. OF DEST.*, */)
DO 25 I = 1, NONTAR
PRINT 25 FORMAT(1I1, XTAR(I), YTAR(I), DEST(I))
CONTINUE
PRINT 111
PRINT 30 FORMAT(1H1, "NONTARGETS (*I4, *)")
PRINT 35 FORMAT(1H1, "NONTARGET NUMBER*, 6X*, X COORD.*, 7X*, Y COORD.*, 12X,
1*VALUE*, 7X*, DAMAGE LIMIT*/)
DO 41 I = 1, NONTAR
PRINT 41 FORMAT(1I1, XNON(I), YNON(I), FACTOR(I), UPNOND(I))
CONTINUE
PRINT 111
PRINT 45 FORMAT(1H1, "AIMPOINTS (*I4, *)")
PRINT 50 FORMAT(1H1, "AIMPOINT NUMBER*, 6X*, X COORD.*, 7X*, Y COORD.*/
1*/)
DO 56 I = 1, NOAIM
PRINT 56 FORMAT(1I1, XAIM(I), YAIM(I))
CONTINUE
PRINT 111
PRINT 60 FORMAT(1H1, "WEAPON CLASSES (*I4, *)")
PRINT 65 FORMAT(1H1, "CLASS NUMBER*, 5X*, TOTAL AVAILABLE*, 5X*, RELIABILITY*
1*/)
DO 71 I = 1, NOWEAP
PRINT 71 FORMAT(1I1, INUM(I), RELBI(I))
CONTINUE
PRINT 100
PRINT 100 FORMAT(1H1, "WEAPON-TARGET EFFECTIVENESS TABLE")
PRINT 101 FORMAT(1H1, "TARGET")
102  FORMAT(1H1*WEAPON*/)
PRINT102
103  FORMAT(1H0*14X;4X;9(8X;12)/)
00105I*NOWEAP
PRINT110*I,(EFTAR(I;J);J=1;NOTAR)
110  FORMAT(1M0*4X;I2;4X;10F10.3)
105  CONTINUE
PRINT111
PRINT200
200  FORMAT(1H0*WEAPON=NONTARGET EFFECTIVENESS TABLE*)
PRINT201
201  FORMAT(1H1*/NONTARGET*)
PRINT 102
PRINT203* (I;1;=1;NONON)
203  FORMAT(1M0*14X;12.9(8X;12)/)
00205I*NOWEAP
PRINT210* I*(EFFNON(I;J);J=1;NONON)
210  FORMAT(1M0*4X;12;4X;10F10.3)
205  CONTINUE
PRINT111
PRINT111
PRINT1005
1005  FORMAT(1H1*ALLOCATION RESULTS*)
PRINT1100
1100  GOTO1100
PRINT1010
1010  IF (BTNNTS.LT.9999999999.)6UT0H00
1010
1010  FORMAT(1M0*IT IS IMPOSSIBLE TO MEET THE TARGET DAMAGE CONSTRAINTS*
1010  RETURN
1100 PRINT 1125
1125  PRINT1130
1130  FORMAT(1M0*FOLLOWING IS THE OPTIMAL ALLOCATION*)
PRINT1130
1130  CONTINUE
PRINT1136* 
1136  CONTINUE
PRINT1205
1205  PRINT1210
1210  FORMAT(1M0*TARGET DAMAGE*)
PRINT1210
1210  CONTINUE
PRINT1215
1215  FORMAT(1M0*DEST(I)
1215  CONTINUE
PRINT1220
1220  CONTINUE
PRINT1225,TOT
1225  FORMAT(1H0*ORIGINAL TOTAL NONTARGET VALUE WAS *F10.3)
Pc=BTNNTS/TOT*100.
PRINT1230,BTNNTS,PC
1230 FORMAT(1H-,*TOTAL EXPECTED NONTARGET VALUE DESTROYED IS * F10.3,* OR * F10.3,* PERCENT,*))
PRINT111
PRINT1235
1235 FORMAT(1H-,*INDIVIDUAL NONTARGET EXPECTED VALUE DESTROYED LISTED BELOW *)
PRINT1240
1240 FORMAT(1H-,*EXPECTED VALUE DESTROYED* F10.3,*PERCENT* F10.3,*SPECIFIED MAXIMUM (PERCENT)*)
DO 1246 I=1,NONON
WWW=100*UPNOND(I)
PCC=BTNNTV(I)/FACTOR(I)*100*
PRINT1245,I,FACTOR(I),BTNNTV(I),PCC,WWW
1245 FORMAT(1H-,* F10.3,* F10.3,* F10.3,* F10.3,* F10.3)
1246 CONTINUE
RETURN
END
**INPUT SPECIFICATIONS**

Refer to problem P' for notation.

Core requirements impose the following limits:

\[
M \leq 10 \\
N \leq 10 \\
J \leq 100 \\
I \leq 10 
\]

<table>
<thead>
<tr>
<th>Card Name</th>
<th>Input Parameters</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIMITS (1 card each)</td>
<td>M, N, J, I</td>
<td>4I10</td>
</tr>
<tr>
<td>TARGET</td>
<td>(x_m, y_m, c_m)</td>
<td>3F10.6</td>
</tr>
<tr>
<td>NONTARGET (1 card each)</td>
<td>(\mu_n, \nu_n, \lambda_n, d_n)</td>
<td>4F10.6</td>
</tr>
<tr>
<td>AIMPOINT (1 card each)</td>
<td>(\xi_j, \zeta_j)</td>
<td>2F10.6</td>
</tr>
<tr>
<td>WEAPON (1 deck each)</td>
<td>(w_1, p_1, a_1, l_1, a_1, 2, b_1, l_1, b_1, 2, \ldots)</td>
<td>8F10.6/2F10.6</td>
</tr>
</tbody>
</table>