UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Integer Prim-Read Solutions to a Class of Target Defense Problems		5. TYPE OF REPORT & PERIOD COVERED Final
		6. PERFORMING ORG. REPORT NUMBER IDA Paper P-1689
7. AUTHOR(s) Stefan A. Burr, James E. Falk, Alan F. Karr		8. CONTRACT OR GRANT NUMBER(s) IDA Central Research Program
9. PERFORMING ORGANIZATION NAME AND ADDRESS Institute for Defense Analyses 1801 N. Beauregard Street Alexandria, VA 22311		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE September 1983
		13. NUMBER OF PAGES 57
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

ballistic missile defense, antiballistic missiles, nationwide defense, strategic defense, Prim-Read, point defense, optimization

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

The problem we address is that of choosing a deployment and firing doctrine for defending separated point targets of (potentially) different values against an attack by an unknown member of sequentially arriving missiles. We minimize the total number of defenders subject to an upper bound on the maximum expected value damage per attacking weapon. We show that the Greedy Algorithm produces an optimal integral solution to this problem.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Copy 49 of 125 copies

IDA PAPER P-1689

INTEGER PRIM-READ SOLUTIONS TO A CLASS OF TARGET DEFENSE PROBLEMS

Stefan A. Burr
James E. Falk
Alan F. Karr

September 1983

IDA

INSTITUTE FOR DEFENSE ANALYSES
PROGRAM ANALYSIS DIVISION

IDA Log No. HQ 82-24955

IDA PAPER P-1689

INTEGER PRIM-READ SOLUTIONS TO A CLASS OF TARGET DEFENSE PROBLEMS

Stefan A. Burr
James E. Falk
Alan F. Karr

September 1983

INSTITUTE FOR DEFENSE ANALYSES
PROGRAM ANALYSIS DIVISION
1801 N. Beauregard Street, Alexandria, Virginia 22311
IDA Central Research Program

CONTENTS

Abstract v

1. INTRODUCTION 1

2. ASSUMPTIONS AND NOTATION 3

3. THE MULTI-TARGET PROBLEM AND SOME OF ITS PROPERTIES 7

   Theorem 1. (Existence of a Solution) 8

   Theorem 2. (Reduction of the Multi-target Problem) 10

4. THE SINGLE-TARGET CASE--MONOTONICITY AND ADMISSIBILITY 12

   Theorem 3. (Existence of a Monotone Solution) 12

   Definition of an Admissible Solution 13

   Theorem 4. (Equivalence of Monotonicity and Admissibility) 13

   Lemma 14

   Example: Multiple Optimal Solutions 15

5. SOLUTION OF THE SINGLE-TARGET PROBLEM 16

   The Algorithm 16

   Theorem 5. (Validation of the Algorithm) 16

6. THE CASE $q = 0$ 20

   Theorem 6. (Solution for $q = 0$) 21

7. EXAMPLES 21

   Solution of a Single Target Case 22

   Solution of a Nationwide Case 23

ACKNOWLEDGMENTS 32

REFERENCES 33

APPENDICES

A. PROOF OF THE LEMMA A-1

B. ANOTHER ALGORITHM FOR THE ALL-INTEGER VERSION OF THE PRIM-READ MODEL B-1

FIGURES

1 Probabilities of Target Destruction vs Attack Size . . . 25
2 Total Optimal Deployment 28

TABLES

1 Probability of Target Destruction 24
2 Solution of the Single Target Case 26
3 Solution of the Single Target Case Without Integer Restrictions 27
4 Solution of a Nationwide Case 29

ABSTRACT

The problem we address is that of choosing a deployment and firing doctrine for defending separated point targets of (potentially) different values against an attack by an unknown number of sequentially arriving missiles. We minimize the total number of defenders subject to an upper bound on the maximum expected value damage per attacking weapon. We show that the Greedy Algorithm produces an optimal integral solution to this problem.

1. INTRODUCTION

In the late 1950s and early 1960s, R.C. Prim and W.T. Read [5] at the Bell Telephone Laboratories and the Office of the Secretary of Defense developed a method to deploy and fire interceptor missiles defending a collection of separated point targets against an attack by an unknown number of sequentially arriving ballistic missiles. Basically, the scheme involved deploying and firing the interceptors in a manner that equalizes the probabilities that each of a prescribed number of attacking weapons destroys the target. These deployments (and associated firing doctrines) have become known as "Prim-Read" deployments.

In January 1964 a major study of limiting damage by civil defense and antiballistic missile defense was completed [1]. This study, performed under the direction of Glenn A. Kent, was highly influential in shaping United States policy with respect to strategic defense (see, for instance, [3] for background discussion). The study used Prim-Read deployments and firing doctrines for interceptors.

Optimality properties of the Prim-Read concept were established by Karr in [4]. Some background discussion of the development of the methodology is also contained in that document. The work to date has been limited to continuous (i.e., not necessarily integral) firing doctrines.

In the present paper, we focus on a particular class of deployment problems whose continuous solutions are shown in [4] to be of the Prim-Read variety, and develop a method that will produce globally optimal solutions of integer versions of these problems. This class will thus be distinguished by two features: (a) the number of defending interceptors allocated to each attacking missile is required to be an integer, and (b) these problems have non-integer versions for which (non-integer) Prim-Read deployments are known to be solutions.

The underlying "physics" of the model, as described in the next section, is that of [1,4] and [2, p. 92]. The optimization problem that we study is taken from [4], with the sole modification that we add integer restrictions on the defense. The authors do not wish to imply that the assumptions here are the only "correct" ones; numerous alternative models can reasonably be addressed.

The problem that we initially address has the general form of minimizing the total number of interceptors required to defend T targets against an attack of k missiles, where <u>neither k nor upper or lower bounds on k are known to the defender in advance</u>, subject to a given upper bound on the maximum target value destroyed per attacking weapon. In the next section, we will introduce the specific problem assumptions and notation used to define the problem.

In Section 3, we note the existence of a solution, and show that the multi-target problem reduces to a collection of single-target problems. Section 4 presents some properties of solutions to the single target problem.

Section 5 contains the main result of this paper, namely that the "Greedy Algorithm" solves the single-target case. Combined with the reduction result of Section 3, this completely solves the specific "Prim-Read" problem that we are addressing.

Section 6 treats the special case of perfect interceptors, whose solution is very easily obtained and is shown also to be a limiting case of the solutions produced by the algorithm of Section 5. Section 7 presents two detailed examples.

2. ASSUMPTIONS AND NOTATION

The general problem we address has the form of minimizing the total number of interceptors required to ensure that the maximum possible damage under any attack is bounded by a linear function of the attack size. The purpose of this section is to introduce specific assumptions and notation so that the problem becomes well-defined.

(1) We assume that the defender must set his strategy first, and in ignorance of both the attacker's resources to be expended (i.e., total number of attacking weapons) and intentions concerning allocation of missiles among targets (i.e., the attacker's strategy). We allow the attacker knowledge of the specific defense strategy adopted by the defender.

(2) We assume that there are T targets, and that each target i has a value $v(i) > 0$. The $v(i)$ need not be integers.

(3) We assume that defending interceptors must be assigned to specific targets, i.e., they cannot be used in an "area defense" mode. We also assume that the targets are separated to the extent that an interceptor assigned to defend a specific target cannot be used to defend a different target. Moreover, an attacking missile directed at a specific target has no effect on the survival or destruction of a different target.

(4) We assume that attacking weapons arrive sequentially (i.e., so that they can be labeled "first", "second", ...) with sufficient time between arrivals that the fate of any particular attacking missile is independent of that of every other attacking missile. The defender does not know either the attack size or any (nontrivial) upper or lower bounds on it. The number of interceptors committed to a given attacking missile must be chosen with no knowledge concerning how many missiles will follow at the same target. In effect the

defender is thereby forced to assign interceptors in advance to potentially arriving missiles at each target.

(5) We assume that neither side can change his strategy during the attack. Thus we are considering only non-adaptive strategies for both sides.

Based on the above assumptions, we define a <u>defense strategy</u> d to be a semi-infinite matrix

$$d = \begin{bmatrix} d(1,1) & d(1,2) & \cdots \\ \vdots & & \\ d(i,1) & d(i,2) & \cdots \\ \vdots & & \\ d(T,1) & d(T,2) & \cdots \end{bmatrix},$$

where d(i,j) = number of interceptors assigned at target i, to be directed at the j^{th} incoming attacking weapon (if there is one) at that target. We will also use the notation d(i) to denote the i^{th} row of d. Throughout this paper, the d(i,j)'s are restricted to be nonnegative <u>integers</u>. In practice, all but a finite number of the d(i,j)'s will be zero.

In a similar fashion, an <u>attack strategy</u> is a vector

$$a = \begin{bmatrix} a(1) \\ \vdots \\ a(i) \\ \vdots \\ a(T) \end{bmatrix},$$

where a(i) denotes the number of attacking weapons to be directed at target i; the a(i)'s are also restricted to be nonnegative integers.

We remind the reader that the defender must choose d first, and the attacker is then free to choose a, based on knowledge of d. Note that we are giving the attacker knowledge of each d(i,j), i.e., he not only knows the number of interceptors stationed at target i, but he also knows the schedule by which these interceptors will be fired.

(6) Let $q \in [0,1)$ denote the probability that an interceptor engaging an attacking missile <u>fails to destroy</u> that missile. We assume that different interceptors deployed against a given attacking missile are independent, and that the same q is valid for every interceptor and attacking missile engagement no matter how many interceptors attack each missile.

(7) We assume that an attacking missile that penetrates the defense will destroy its intended target with probability one.

For a given deployment d and a given attack strategy a, the probability that target i is destroyed is

$$1 - \prod_{j=1}^{a(i)} (1-q^{d(i,j)}) \ ;$$

with the convention that $\prod_{1}^{0} (\cdot) = 1$, this is zero if a(i) = 0. Therefore

$$V(d,a) = \sum_{i=1}^{T} v(i)\left[1 - \prod_{j=1}^{a(i)} (1-q^{d(i,j)})\right] .$$

It is this quantity that an attacker might seek to maximize for a given d, subject to an upper bound on the number of missiles to be expended.

If the number of attacker's missiles to be expended were known to the defender in advance to be, say, k, a reasonable problem to address might be

$$\begin{aligned} &\underset{d(i,j)}{\text{minimize}} && \sum_{i=1}^{T} \sum_{j=1}^{\infty} d(i,j) \\ &\text{subject to} && V(d,a) \leq u, \\ & && \text{for all a such that} \\ & && \sum_{i=1}^{T} a(i) \leq k \ , \end{aligned}$$

where u is some prescribed upper limit on the expected target value damage. However, in the current context, k is not known in advance. The alternative that we address assumes instead a bound on the maximum expected damage incurred <u>per attacking weapon</u>. This alternative is in the same spirit, but makes sense when k is not known in advance.

3. THE MULTI-TARGET PROBLEM AND SOME OF ITS PROPERTIES

In Sections 3-5 we assume that $0 < q < 1$. The case $q = 0$, in which a single interceptor is certain to destroy an attacking missile, <u>is</u> of physical interest and is treated in Section 6 using arguments different from those in Sections 3-5. As will emerge, the case $q = 0$ is much easier to handle because non-integral numbers of interceptors would never arise in optimal solutions even if they were not forbidden, yet still this case can be regarded as a limiting verion of the case $q > 0$.

Let a real number $s > 0$ be given. This number will serve as an upper bound on the maximum expected damage incurred per attacking weapon. Some schemes for selecting reasonable values of s are given in [3]; we note just one here.

If V is the sum of all of the target values, then V is a given upper bound on the total expected damage incurred, and if the total number of attacker's weapons equals K, then s=V/K is the corresponding upper bound on the maximum expected damage incurred per attacking weapon is the attacker commits his entire inventory. The problems presented below ensure that this average is not exceeded even if the attacker holds back some missiles or builds new ones.

The problem we address herein has the form

$$P(s): \begin{cases} \underset{d(i,j)}{\text{minimize}} \quad \sum_{i=1}^{T} \sum_{j=1}^{\infty} d(i,j) \\ \\ \text{subject to} \quad V(d,a) \leq s \sum_{i=1}^{T} a(i), \text{ for all } a, \end{cases}$$

where, we recall, each d(i,j) and each a(i) is further restricted to be a non-negative integer.

We note that problem P(s) has the equivalent formulation

$$\underset{d(i,j)}{\text{minimize}} \quad \sum_{i=1}^{T} \sum_{j=1}^{\infty} d(i,j)$$

$$\text{subject to} \quad \max \left\{ V(d,a): \sum_{i=1}^{T} a(i) \leq k \right\} \leq sk$$

$$\text{for } k = 1,2,3, \ldots ,$$

and we will address this form in several of the proofs.

<u>Theorem</u> 1 (Existence of a Solution): Problem P(s) has a solution for each value of s.

<u>Proof</u>: First note that the number of potentially effective constraints is finite. For[1] if we set $\tilde{k} = \left\lceil \sum_{i=1}^{T} v(i)/s \right\rceil$, we

[1] On those occasions where it is necessary to round non-integer quantities, we use the notation

$$\lfloor x \rfloor = \max\{i:i \text{ is integer, } i \leq x\}$$
$$\lceil x \rceil = \min\{i:i \text{ is integer, } i \geq x\}.$$

note that the constraints for $\tilde{k}$, $\tilde{k}+1$, ... are irrelevant, since $V(d,a)$ is bounded above by $\sum_{i=1}^{T} v(i)$.

Let

$$f_k(d) = \max\left\{V(d,a): \sum_{i=1}^{T} a(i) \leq k\right\}$$

denote the k^{th} constraining function, so that the k^{th} constraint has the form

$$f_k(d) < sk.$$

Now define the function

$$g_k(d) = \bar{v} \sum_{i=1}^{T} (1 - \prod_{j=1}^{k} (1-q^{d(i,j)})),$$

where $\bar{v} = \max\{v(i): i=1,\ldots,T\}$. Since $f_k(d) \leq g_k(d)$, it suffices to establish feasibility for the constraints $g_k(d) \leq sk$ $(k=1,\ldots,\tilde{k}-1)$. From here, it is easy to show the existence of feasible solutions of the form

$$d(i,j) = \begin{cases} D & \text{for } j = 1,\ldots,\tilde{k}-1,\ i = 1,\ldots,T \\ 0 & \text{for } j = \tilde{k},\ldots;\ i = 1,\ldots,T, \end{cases}$$

where D is chosen large enough that $q^D \leq (1-(1-\frac{sk}{T\bar{v}})^{1/k})$ for $k=1,\ldots,\tilde{k}=1$, so that $P(s)$ is feasible. This, together with the constraints that $d(i,j)$ take on integer values, and the form of the objective function, guarantees that $P(s)$ has a solution. $\square$

The next theorem establishes the result that the problem $P(s)$ can be solved by solving a collection of single-target defense problems, one for each target. This reduction permits us to focus attention in Sections 4 and 5 on the single-target problem P(r) to be stated momentarily.

For convenience of notation, we define

$$V_i(d(i),k) = v(i)\left(1 - \prod_{j=1}^{k} (1-q^{d(i,j)})\right),$$

where $d(i) = (d(i,1),d(i,2),\ldots)$ as above. Then

$$V(d,a) = \sum_{i=1}^{T} V_i(d(i),a(i)) .$$

The proof of the following theorem depends on the fact that the expected damage is the sum of the expected damage at each target, and this is bounded above by the function sk, which is linear in k.

<u>Theorem 2</u> (Reduction of the Multi-target Problem): For each $i \in \{1,\ldots,T\}$, let $d^*(i)$ solve the problem

$$P_i(s): \begin{cases} \text{minimize} & \sum_{j=1}^{\infty} d(i,j) \\ \text{subject to} & V_i(d(i),k) \leq sk \quad (k=1,2,\ldots). \end{cases}$$

Then $d^* = (d^*(1),d^*(2),\ldots,d^*(T))$ solves P(s).

<u>Proof</u>: First note that d^* is feasible for problem P(s), since

$$V_i(d^*(i),a(i)) \leq sa(i)$$

for any a(i), so that

$$\max\left\{V(d^*,a): \sum_{i=1}^{T} a(i) \leq k\right\} \leq sk$$

for each k = 1, 2, ...

Now suppose that d^* is not optimal for $P(s)$; then by Theorem 1 there exists a feasible deployment $\bar{d}$ which satisfies

$$\sum_{i=1}^{T} \sum_{j=1}^{\infty} \bar{d}(i,j) < \sum_{i=1}^{T} \sum_{j=1}^{\infty} d^*(i,j) ,$$

so that, at some target I, we have

$$\sum_{j=1}^{\infty} \bar{d}(I,j) < \sum_{j=1}^{\infty} d^*(I,j) .$$

By definition,

$$\begin{aligned} V_I(\bar{d}(I),k) &= \sum_{\substack{i=1 \\ i \neq I}}^{T} V_i(\bar{d}(i),0) + V_I(\bar{d}(I),k), \\ &= V(\bar{d},\underset{\sim}{k}) \end{aligned}$$

where $\underset{\sim}{k}$ is the I^{th} unit vector times k. Since

$$V(\bar{d},\underset{\sim}{k}) \leq \max\left\{V(\bar{d},a): \sum_{i=1}^{T} a(i) \leq k\right\} \leq sk ,$$

we see that $\bar{d}(I)$ is feasible to problem $P_I(s)$, but entails a smaller total deployment than $d^*(I)$, contradicting the assumed optimality of the latter for $P_I(s)$. $\square$

Since we are now able to focus attention on the single-target problem, it is convenient to simplify the notation. To that end, we will now employ the symbol d to denote a <u>vector whose j^{th} component $d(j)$</u> denotes the number of interceptors assigned to attacking weapon j. Also, we introduce the symbol r in place of $s/v(i)$ and suppress reference to target i. Note that the values of interest of r lie between zero and one (the value of a single target is normalized to equal one).

The reader should note that this change of notation (previously d was a <u>matrix</u>, and d(i) was a <u>vector</u>) will remain in place throughout the rest of the paper, except in Section 6 and Example 2 of Section 7, where a multi-target problem is solved.

The single-target problem thus becomes

$$P(r): \quad \begin{array}{l} \text{minimize } \sum_{j=1}^{\infty} d(j) \\ \text{subject to } V(d,k) \le rk \quad (k=1,2,\ldots) , \end{array}$$

where $V(d,k) = \left(1 - \prod_{j=1}^{k} (1-q^{d(j)})\right)$.

4. THE SINGLE-TARGET CASE--MONOTONICITY AND ADMISSIBILITY

<u>**Definition**</u>: Let d denote any deployment vector (d(1),d(2),...). We say that d is <u>monotone</u> if

$$d(1) \ge d(2) \ge \ldots \quad .$$

Under an attack by missiles arriving sequentially in time, a monotone deployment defends more heavily against the earlier attacking weapons, an action which seems reasonable.

During the proof of Theorem 3 and for the remainder of the paper we suppress dependence on r and write "Problem P(r)" as simply "Problem P".

<u>**Theorem 3**</u>: For every r there exists a monotone optimal solution of problem P.

Proof: Let d* denote an optimal solution of problem P. Then

$$V(d^*,k) = 1 - \prod_{j=1}^{k} (1-q^{d^*(j)}) \leq rk, \quad k=1,2,\ldots \quad .$$

If $d^*(\bar{j}) < d^*(\bar{j}+1)$ for some $\bar{j}$, define the vector $\tilde{d}$ from d* by interchanging the components $d^*(\bar{j})$ and $d^*(\bar{j}+1)$. Both $\tilde{d}$ and d* have the same component sum, and we note that

$$V(\tilde{d},k) = V(d^*,k) \text{ for } k \neq \bar{j}$$

while

$$V(\tilde{d},\bar{j}) < V(d^*,\bar{j})$$

so that $\tilde{d}$ is also feasible to problem P. Since it is easily seen that one can repeatedly interchange adjacent components of d* that increase, until one obtains a monotone solution of P, the theorem is proven. $\square$

Note that the above interchange strictly decreased the value of $V(d^*,\bar{j})$, while keeping the other quantities $V(d^*,k)$ unchanged, i.e., with the same total number of interceptors, the defender can limit his expected damage under an attack by $k \neq \bar{j}$ missiles to the same amount, and decrease this damage if attacked by $\bar{j}$ missiles. This is obviously a desirable property from the defender's point of view, and motivates us to make the following definition.

Definition: A deployment d that solves problem P is admissible if there is no other deployment $\tilde{d}$ such that

(a) $\tilde{d}$ also solves problem P

(b) $V(\tilde{d},k) \leq V(d,k)$ for all k

and $V(\tilde{d},k) < V(d,k)$ for some k .

The proof of the previous theorem establishes that admissible deployments are monotone (or else they could be "monotonized", yielding deployments $\tilde{d}$ having at least one strictly better value $V(\tilde{d},k)$). The next theorem establishes that optimal monotone deployments are, in fact, admissible.

<u>Theorem 4</u>: An optimal deployment d^* of problem P is monotone if and only if it is admissible.

<u>Proof</u>: Let d^* be an optimal monotone deployment, and assume that it is not admissible. Thus there must be another optimal deployment d that also solves P, and for which

$$V(\tilde{d},k) \leq V(d^*,k)$$

for all k, with strict inequality holding for some k. We can assume that $\tilde{d}$ is also monotone (perhaps "monotonized" by the procedure established above).

Since $V(\tilde{d},1) \leqq V(d^*,1)$, it follows that $\tilde{d}(1) \geq d^*(1)$. Let k_1 denote the smallest value of k such that $\tilde{d}(k_1) > d^*(k_1)$. We note, then, that

$$V(\tilde{d},k_1) < V(d^*,k_1).$$

Because both $\tilde{d}$ and d^* are optimal, we know that $\sum_{k=1}^{\infty} \tilde{d}(k) = \sum_{k=1}^{\infty} d^*(k)$. Now let k_2 denote the smallest value of $k > k_1$ such that

$$\sum_{k=1}^{k_2} \tilde{d}(k) = \sum_{k=1}^{k_2} d^*(k).$$

We claim that $V(\tilde{d},k_2) > V(d^*,k_2)$, which violates the assumption that d^* was not admissible. It suffices to show that

$$\prod_{k=1}^{k_2}(1-q^{\tilde{d}(k)}) < \prod_{k=1}^{k_2}(1-q^{d^*(k)})$$

and this is a direct result of the following Lemma, if we set $t = k_2 - k_1 + 1$, $a_i = \tilde{d}(i+k_1-1)$, and $b_i = d^*(i+k_1-1)$. $\square$

<u>Lemma</u>: Let $a = (a_i)$ and $b = (b_i)$ be a pair of positive monotonically nonincreasing vectors of size t such that

$$\text{(a)} \quad \sum_{i=1}^{s} a_i > \sum_{i=1}^{s} b_i \qquad \text{for } s = 1,2,\ldots, t-1 ;$$

$$\text{(b)} \quad \sum_{i=1}^{t} a_i = \sum_{i=1}^{t} (1-q^{b_i}) .$$

Then for $q \epsilon (0,1)$,

$$\prod_{i=1}^{t} (1-q^{a_i}) < \prod_{i=1}^{t} (1-q^{b_i}) .$$

<u>Proof</u>: See Appendix A.

There may be more than one optimal admissible solution, as the following example exhibits.

<u>Example</u>: With $r = 1/3$ and $q = 0.6$, the deployments $d^* = (3,2)$ and $\tilde{d} = (4,1)$ are both optimal, admissible solutions. We have

$$\begin{aligned} V(d^*,1) &= .2160 & \qquad V(\tilde{d},1) &= .1296 \\ V(d^*,2) &= .4982 & \qquad V(\tilde{d},2) &= .6518 \\ V(d^*,3) &= 1.000 & \qquad V(\tilde{d},3) &= 1.000 , \end{aligned}$$

so that the solution (3,2) yields a higher expected damage if there is but a single attacker, but a lower expected damage if there are two attackers. The average expected damage per attacker values are

$$\frac{V(d^*,1)}{1} = 0.2160 \qquad \frac{V(\tilde{d},1)}{1} = 0.1296$$

$$\frac{V(d^*,2)}{2} = 0.2491 \qquad \frac{V(\tilde{d},2)}{2} = 0.3259$$

$$\frac{V(d^*,3)}{3} = 0.3333 \qquad \frac{V(\tilde{d},3)}{3} = 0.3333 \ ,$$

which are all, as required, bounded above by $r = 1/3$.

5. SOLUTION OF THE SINGLE-TARGET PROBLEM

In this section, we present a proof that the "Greedy Algorithm" solves the single-target problem. The method begins by computing the minimum number of defending interceptors necessary to ensure that the expected value of the damage from one attacker is at most r. The iteration step assumes that k-1 attacking weapons have been assigned interceptors, and the minimum number of interceptors to be assigned to attacking weapon k is then computed, subject to the usual bound.

<u>The Algorithm</u>:

Let $\bar{d}(1) = \left\lceil\!\left\lceil \frac{\ell n\ r}{\ell n\ q} \right\rceil\!\right\rceil$ and, given $\bar{d}(1), \ldots, \bar{d}(k-1)$, let

$$\bar{d}(k) = \left\lceil\!\!\left\lceil \frac{\ell n \left(1 - \dfrac{1-rk}{\prod_{i=1}^{k-1} (1-q^{\bar{d}(i)})}\right)}{\ell n\ q} \right\rceil\!\!\right\rceil$$

for $2 \leq k < 1/r$. For $k \geq 1/r$, let $\overline{d}(k) = 0$. (Note that the general form for $k \geq 2$ is actually valid for $k = 1$, with the convention that $\prod_{i=1}^{0} (\cdot) = 1$.)

Theorem 5: The above recursion yields an optimal solution of $\overline{d}$ of Problem P.

Proof: The first component $\overline{d}(1)$ solves

$$\begin{aligned} &\text{minimize} \quad d(1) \\ &\text{subject to } 1 - (1-q^{d(1)}) \leq r \end{aligned}$$

and the k^{th} component $\overline{d}(k)$ solves the problem

$$\begin{aligned} &\text{minimize } d(k) \\ &\text{subject to} \\ &1 - \left(\prod_{\ell=1}^{k-1} (1-q^{\overline{d}(\ell)}) \right) (1-q^{d(k)}) \leq rk \end{aligned}$$

for $2 \leq k < 1/r$, as is easily verified.

Let $n = \lfloor\lfloor 1/r \rfloor\rfloor$. Clearly, if $d(1), \ldots, d(n)$ is _any_ sequence feasible for the first n constraints, then $d(1), \ldots, d(n), 0, 0, \ldots$ is feasible for _all_ constraints, as the $(n+1)^{st}$, $(n+2)^{nd}$, ... are redundant.

Let

$$C = \{x \in R^n : x \geq 0,\ V(x,k) \leq rk;\quad k=1,\ldots,n\}$$

(the members of C are _not_ restricted to have integer components), and let L denote the lattice of points with integer coordinates in $(R^n)^+$. We note that

$$\overline{d} \in C \cap L ,$$

and we must show that $d \in C \cap L$ implies that

$$\sum_{i=1}^{n} d(i) \geq \sum_{i=1}^{n} \bar{d}(i) .$$

We define, for each $\ell = 1, \ldots, n$, a deployment d^{ℓ} by setting

$$d^{\ell}(i) = \begin{cases} \bar{d}(i) & \text{if } i \neq \ell \\ \bar{d}(\ell) - 1 & \text{if } i = \ell \end{cases} .$$

None of these deployments can be feasible, because $\bar{d}(\ell)$ was the <u>smallest</u> $d(\ell)$ to satisfy constraint ℓ, given $\bar{d}(1), \ldots, \bar{d}(\ell-1)$.

Let

$$K = \{d: \sum_{i=1}^{n} d(i) = \sum_{i=1}^{n} \bar{d}(i) - 1\}$$

so that we need to show that $C \cap (K \cap L) = \emptyset$. The points d^{ℓ} are those elements in $K \cap L$ that are "nearest" to $\bar{d}$. In what follows, we will show that other points in $K \cap L$ (which are farther away from $\bar{d}$) would have more difficulty in being feasible than any of the d^{ℓ}'s, so that all of $K \cap L$ lies outside of C.

Any element $\tilde{d}$ in $K \cap L$ can be obtained from an element in $\{d^1, \ldots, d^n\}$ by a sequence of transformations that simultaneously increase a component of d^{ℓ} by 1, and decrease another component of d^{ℓ} by 1.

Now fix some ℓ, and let i_1 and i_2 be distinct integers between 1 and n. Define

$$\tilde{d}(i) = \begin{cases} d^{\ell}(i_1) + 1 & i = i_1 \\ d^{\ell}(i_2) - 1 & i = i_2 \\ d^{\ell}(i) & i \neq i_1, i_2 , \end{cases}$$

so that $\tilde{d}$ is in $K \cap L$. We want to show that $\tilde{d} \notin C$; for otherwise, $\tilde{d}$ would be a better feasible solution than $\bar{d}$.

The first component of $\tilde{d}$ that differs from $\bar{d}$ must offer an increase in that component because of the way $\bar{d}$ was constructed. But we can assume that $\tilde{d}$ differs from $\bar{d}$ in components i_1, ℓ, and i_2 (if $i_1 = \ell$, then $\tilde{d} = d^{i_2}$, which is known to lie outside of C). Thus, we can assume that

$$i_1 < \min\{i_2, \ell\} .$$

We claim that

$$V(\tilde{d}, m) \geq V(d^{\ell}, m) , \tag{1}$$

where $m = \max\{i_2, \ell\}$. Since we know that $V(d^{\ell}, m) > rm$, it will follow that $\tilde{d} \notin C$.

Rewriting (1), we see that it is equivalent to

$$(1-q^{d^{\ell}(i_1)+1})(1-q^{d^{\ell}(i_2)-1}) \leq (1-q^{d^{\ell}(i_1)})(1-q^{d^{\ell}(i_2)}) ,$$

which further is equivalent to

$$q^{d^{\ell}(i_1)}(1-q) \leq q^{d^{\ell}(i_2)}(q^{-1}-1) ,$$

i.e.,

$$q^{d^{\ell}(i_1) - d^{\ell}(i_2)+1} \leq 1 \ ,$$

which is certainly true if $d^{\ell}(i_1) - d^{\ell}(i_2) + 1 \geq 0$, i.e., if

$$\overline{d}(i_1) + 1 \geq \overline{d}(i_2) \qquad (2)$$

(d^{ℓ} is the same as $\overline{d}$, except in component ℓ). Now, while the $\overline{d}$ produced by the Greedy Algorithm may not be monotone, at least we see by construction that we always have $\overline{d}(i) + 1 \geq \overline{d}(i+1)$, so that (2) is satisfied.

The procedure extends inductively to <u>any</u> element in $K \cap L$ and the theorem is proved. ▯

Before proceeding to the next section, we point out that Burr has derived a different algorithm based on an equivalent formulation of the problem. It is included in Appendix B of this paper, and has a similar "greedy" nature, but (unlike this method) always produces monotone deployments.

6. THE CASE q = 0

For this section we revert to the multiple-target formulation and notation of Sections 2 and 3. When $q = 0$, interceptors are perfectly effective and the defender ensures destruction of an attacking missile by committing just one interceptor against it. Hence we may and do restrict attention to deployments $d = (d(i,j))$ for which $d(i,j)$ is either zero or one for each i and j. Furthermore, it is evident that no optimal solution d^* to the basic problem $P(s)$ can have (for some i) $d^*(i,j) = 0$ but $d^*(i,j') = 1$ for some $j < j'$ (since the deployment obtained by changing $d^*(i,j')$ to zero has the same payoff function but strictly fewer interceptors). Thus we consider only deployments d of the form

$$d(i,j) = \begin{cases} 1, & j=1,\ldots,\delta(i) \\ 0, & j>\delta(i)\ , \end{cases}$$

where the $\delta(i)$ are nonnegative integers. (If $\delta(i) = 0$, then $d(i,j) = 0$ for all j.) The deployment d is completely specified by the $\delta(i)$, so we take the $\delta(i)$ as the choice variables of the problem, which now becomes

$$P'(s): \begin{cases} \underset{\delta(i)}{\text{minimize}} \quad \sum_{i=1}^{T} \delta(i) \\ \text{subject to} \quad V(d,a) \leq s \sum_{i=1}^{T} a(i), \qquad \text{for all } a\ . \end{cases}$$

The interpretation of $\delta(i)$ is the number of attacking missiles against which target i is defended and, because $q = 0$, to which it is invulnerable. Against the $(\delta(i)+1)^{st}$ attacking missile it is completely defenseless and would be destroyed. It follows that the payoff function V assumes the simplified form

$$V(d,a) = \sum_{i=1}^{T} v(i)I(a(i)>\delta(i))\ ,$$

where for an event A, $I(A)$ is the indicator function of A. (If A occurs, then $I(A) = 1$ and otherwise $I(A) = 0$.)

Problem $P'(s)$ is virtually trivial to solve.

__Theorem 6__: The unique solution to Problem $P'(s)$ is given by

$$\delta^*(i) = \left\lceil\!\left\lceil \frac{v(i)}{s} - 1 \right\rceil\!\right\rceil\ , \qquad i=1,\ldots,T\ .$$

__Proof__: If the constraints of $P'(s)$ are satisfied, then in particular

$$v(i) \le s(\delta(i)+1) \tag{3}$$

for every i. Conversely, if (3) holds for every i, then for each attack allocation a

$$V(d,a) \le \sum_{i=1}^{T} s(\delta(i)+1) I(a(i) \ge \delta(i)+1)$$

$$\le s \sum_{i=1}^{T} a(i) \ .$$

Thus Problem P´(s) reduces to the problem

$$\underset{\delta(i)}{\text{minimize}} \quad \sum_{i=1}^{T} \delta(i)$$

$$\text{subject to} \quad v(i) \le s(\delta(i)+1) \ , \qquad i=1,\ldots,T \ ,$$

and of course, subject also to the constraint that each $\delta(i)$ be a nonnegative integer. The unique optimal solution to this latter problem is as given above. ▯

The optimal deployment δ^* is monotone at each target, which is consistent with results in Section 4. Also, taking limits as $q \to 0$ in the algorithm of Section 5 yields the deployment

$$\bar{d}(k) = \lim_{q \to 0} \left\lceil\!\left\lceil \frac{\log(rk)}{\log q} \right\rceil\!\right\rceil$$

$$= \begin{cases} 1, & \text{if } k < v/s \\ 0, & \text{if } k \ge v/s \ , \end{cases}$$

which is the same deployment as δ^* from Theorem 6.

7. EXAMPLES

In this section, we present two examples in order to illustrate the method and to compare the results with those obtained by ignoring the integer restrictions. The first example addresses a single target case and is solved for a wide range of r values, while the second example addresses a nationwide case for one value of the parameter s, and the target values set to be the populations of U.S. cities.

Solution of a Single-Target Case

The defender wishes to defend a single target against an attack of unknown size. He decides that he will surrender the target if 8 warheads are directed against it, but wishes to structure a defense against attack sizes of 0,1,...,7. The defender decides to set his defense so that the maximum probability of target destruction (as a function of the attack size) is bounded above by a linear function of the attack size k. Thus he decides to set an upper bound of min {k/8,1} on the probability of target destruction.

In this example, the defender's interceptors have a kill probability of (1-q) = 0.4.

According to the algorithm, we compute

$$\bar{d}(1) = \left\lceil \frac{\ell n\ 0.125}{\ell n\ 0.6} \right\rceil = 5$$

$$\bar{d}(2) = \left\lceil \frac{\ell n\left(1 - \frac{1 - 2/8}{1-(0.6)^5}\right)}{\ell n\ 0.6} \right\rceil = 4$$

$$\bar{d}(3) = 3$$

$$\bar{d}(4) = 4$$

$$\bar{d}(5) = 3$$

$$\bar{d}(6) = 2$$

$$\bar{d}(7) = 2,$$

and $\bar{d}(8) = \bar{d}(9) = \ldots = 0$. With this deployment, the defender will assign five interceptors to the first attacking missile, four to the second, and so on.

Note that this deployment is not monotone. An equivalent monotone solution is $\bar{\bar{d}} = (5,4,4,3,3,2,2,0,\ldots)$. Both $\bar{d}$ and $\bar{\bar{d}}$ use 23 interceptors.

The attacker's optimal probability of target destruction as a function of the attack size is listed in Table 1, for both $\bar{d}$ and $\bar{\bar{d}}$, and plotted in Figure 1. Note that $\bar{\bar{d}}$ yields a better defense against an attack of size three than does $\bar{d}$.

We go on to compute the optimal defense for all values of r in the range [1/10, 1). Table 2 contains the results. In [4], Karr derives the optimal total

$$D^* = \frac{-\ell n\left(\left(\frac{1}{r}\right)!\right)}{\ell n\ q}$$

when 1/r is an integer and the integer restrictions on the components of d are dropped. These totals are included in Table 3. Figure 2 displays information from Tables 2 and 3.

Solution of a Nationwide Case

In this example, we take cities in the U.S. as targets, with their populations in thousands as values. Population data are from the 1980 Census as reported in the 1983 U.S. Almanac. The value of s is 200 (the defense is to be set so that no single attacking weapon can kill more than 200,000 people) and $q = 0.5$. Table 4 presents the continuous solution computed from the method of [4], the continuous solution rounded up for each attacking weapon at each target, and the integer solution from applying the algorithm of Section 4. The required interceptor stockpiles are 349, 446 and 414, respectively, as shown in Table 4.

Table 1. PROBABILITY OF TARGET DESTRUCTION

k	$V(\bar{d},k)$	$V(\bar{\bar{d}},k)$
1	0.07776	0.07776
2	0.19728	0.19728
3	0.37067	0.30121
4	0.45223	0.45223
5	0.57055	0.57055
6	0.72515	0.72515
7	0.82410	0.82410
8	1.0	1.0
9	1.0	1.0
↓	↓	↓

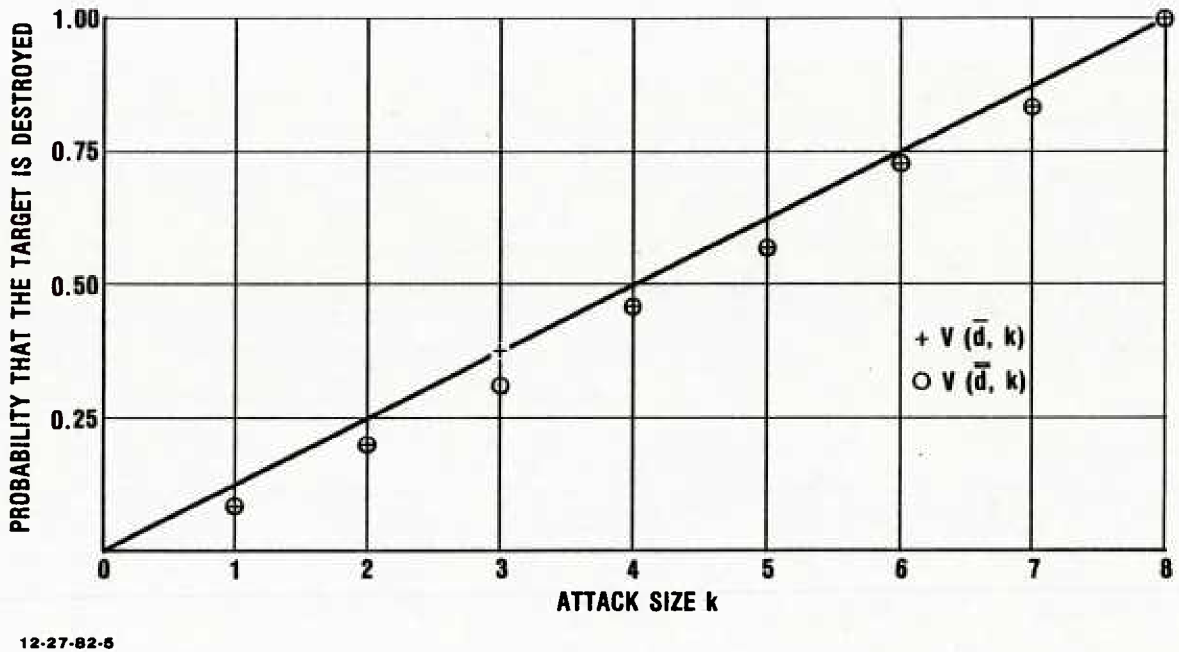

Figure 1. PROBABILITIES OF TARGET DESTRUCTION VS ATTACK SIZE

Table 2. SOLUTION OF THE SINGLE TARGET CASE

r Range	$\bar{d}$ via the Greedy Algorithm									Optimal Sum
[1/10, 0.1004)	5	4	5	4	3	4	3	2	2	32
[0.1004, 0.1019)	5	4	4	4	4	3	3	2	2	31
[0.1019, 0.1046)	5	4	4	4	4	3	3	2	1	30
[0.1046, 0.1087)	5	4	4	4	3	3	3	2	1	29
[0.1087, 1/9)	5	4	4	4	3	3	2	2	1	28
[1/9 , 0.1130)	5	4	4	4	3	3	2	2		27
[0.1130, 0.1141)	5	4	4	3	4	3	2	1		26
[0.1141, 0.1209)	5	4	4	3	3	3	2	1		25
[0.1209, 1/8)	5	4	4	3	3	2	2	1		24
[1/8 , 0.1267)	5	4	3	4	3	2	2			23
[0.1267, 0.1287)	5	4	3	3	3	2	2			22
[0.1287, 0.1296)	5	4	3	3	3	2	1			21
[0.1296, 0.1338)	4	4	4	3	3	2	1			
[0.1338, 0.1353)	4	4	4	3	2	2	1			20
[0.1353, 0.1404)	4	4	3	3	3	2	1			
[0.1404, 1/7)	4	4	3	3	2	2	1			19
[1/7 , 0.1468)	4	4	3	3	2	2				18
[0.1468, 0.1550)	4	4	3	3	2	1				17
[0.1550, 0.1588)	4	4	3	2	2	1				16
[0.1588, 0.1644)	4	3	3	3	2	1				
[0.1644, 1/6)	4	3	3	2	2	1				15
[1/6 , 0.1726)	4	3	3	2	2					14
[0.1726, 0.1878)	4	3	3	2	1					13
[0.1878, 1/5)	4	3	2	2	1					12
[1/5 , 0.2063)	4	3	2	2						11
[0.2063, 0.2160)	4	3	2	1						10
[0.2160, 0.2491)	3	3	2	1						9
[0.2491, 1/4)	3	2	2	1						8
[1/4 , 0.2664)	3	2	2							7
[0.2664, 1/3)	3	2	1							6
[1/3 , 0.3432)	3	2								5
[0.3432, 0.3600)	3	1								4
[0.3600, 0.3720)	2	2								
[0.3720, 1/2)	2	1								3
[1/2 , 0.6000)	2									2
[0.6000, 1)	1									1

Table 3. SOLUTION OF THE SINGLE TARGET CASE WITHOUT INTEGER RESTRICTIONS

r Range	$\bar{d}$ via [4]	Optimal Sum	Optimal Integer Sum
1/10	(4.51, 4.30, 4.07, 3.81, 3.51, 3.15, 2.71, 2.15, 1.36, 0,...)	29.57	32
(1/10,1/9)	not allowed	-	28 to 32
1/9	(4.30, 4.07, 3.81, 3.51, 3.15, 2.71, 2.15, 1.36, 0,...)	25.06	27
(1/9,1/8)	not allowed	-	24 to 27
1/8	(4.07, 3.81, 3.51, 3.15, 2.71, 2.15, 1.36, 0,...)	20.76	23
(1/8,1/7)	not allowed	-	19 to 23
1/7	(3.81, 3.51, 3.15, 2.71, 2.15 1.36, 0,...)	16.69	18
(1/7,1/6)	not allowed	-	15 to 18
1/6	(3.51, 3.15, 2.71, 2.15, 1.36, 0,...)	12.88	14
(1/6,1/5)	not allowed	-	12 to 14
1/5	(3.15, 2.71, 2.15, 1.36, 0,...)	9.37	11
(1/5,1/4)	not allowed	-	8 to 11
1/4	(2.71, 2.15, 1.36, 0,...)	6.22	7
(1/4,1/3)	not allowed	-	6 or 7
1/3	(2.15, 1.36, 0,...)	3.51	5
(1/3,1/2)	not allowed	-	3,4, or 5
1/2	(1.36, 0,...)	1.36	2
(1/2,1)	not allowed	-	1 or 2

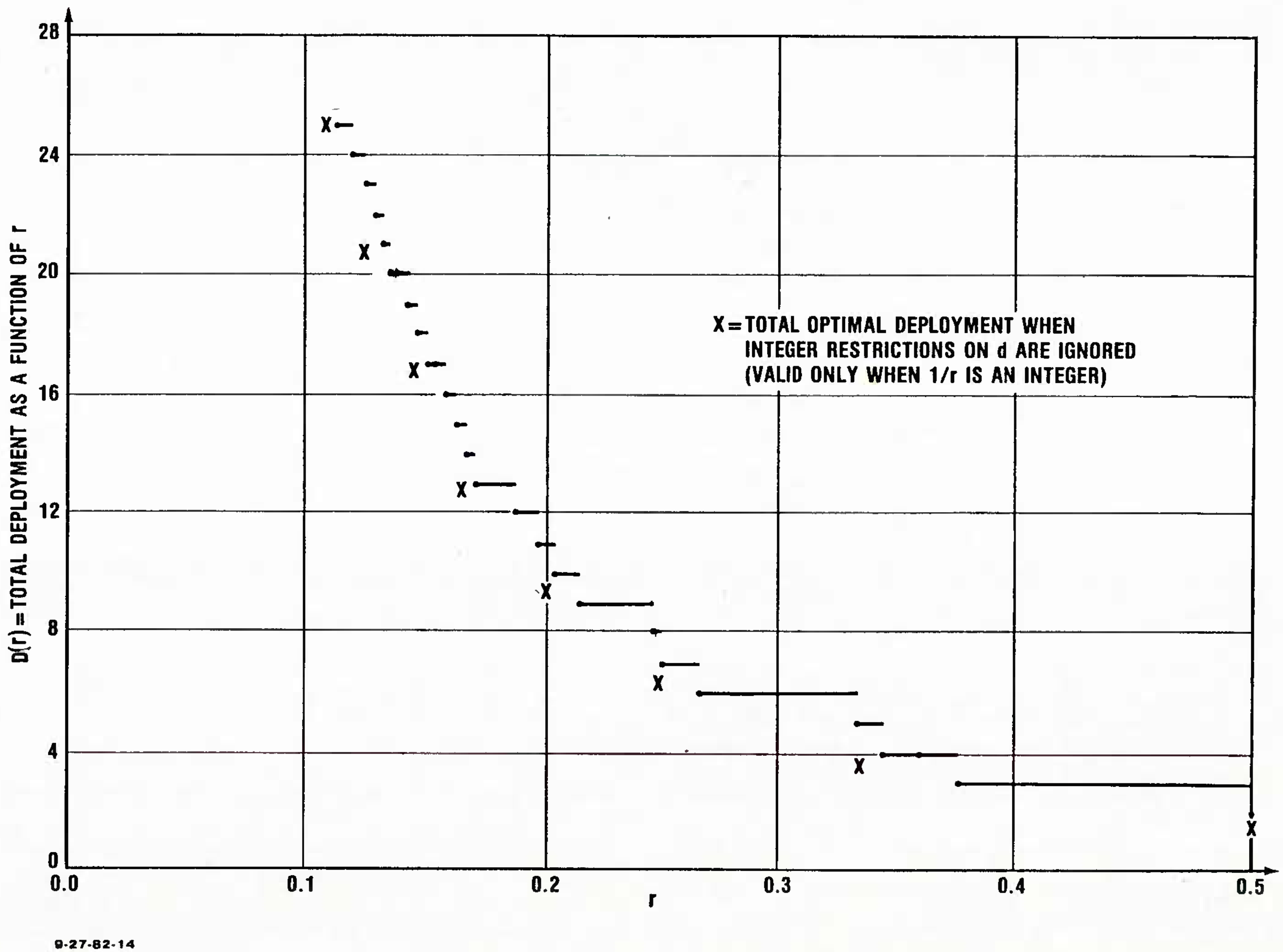

Figure 2. TOTAL OPTIMAL DEPLOYMENT

Table 4. SOLUTION OF A NATIONWIDE CASE

Target	Value	Prim-Read Stockpile	Rounded	Integer Stockpile
1	7071	134.92	153	139
2	3005	40.37	49	43
3	2997	40.20	45	42
4	1688	16.85	21	19
5	1594	15.22	17	16
6	1203	9.54	14	12
7	904	5.93	8	8
8	876	5.60	8	7
9	787	4.48	5	5
10	785	4.47	5	5
11	765	4.30	5	5
12	763	4.28	5	5
13	701	3.72	5	5
14	679	3.50	5	5
15	646	3.15	5	5
16	638	3.06	5	4
17	637	3.04	5	4
18	636	3.03	5	4
19	574	2.42	3	3
20	565	2.37	3	3
21	563	2.35	3	3
22	557	2.31	3	3
23	541	2.21	3	3
24	494	1.86	3	3
25	491	1.84	3	3
26	456	1.55	3	3
27	453	1.52	3	3
28	448	1.47	3	3
29	425	1.26	3	3
30	425	1.26	3	3
31	424	1.25	3	3
32	403	1.03	3	3
33	385	0.94	1	1
34	385	0.94	1	1
35	371	0.89	1	1
36	366	0.87	1	1
37	361	0.85	1	1
38	361	0.85	1	1
39	358	0.84	1	1
40	355	0.83	1	1
41	347	0.79	1	1
42	345	0.79	1	1
43	339	0.76	1	1
44	332	0.73	1	1
45	331	0.73	1	1

Table 4. SOLUTION OF A NATIONWIDE CASE (Continued)

Target	Value	Prim-Read Stockpile	Rounded	Integer Stockpile
46	329	0.72	1	1
47	314	0.65	1	1
48	312	0.64	1	1
49	298	0.58	1	1
50	284	0.51	1	1
51	279	0.48	1	1
52	276	0.46	1	1
53	272	0.44	1	1
54	270	0.43	1	1
55	267	0.42	1	1
56	262	0.39	1	1
57	242	0.28	1	1
58	237	0.24	1	1
59	237	0.24	1	1
60	232	0.21	1	1
61	224	0.16	1	1
62	222	0.15	1	1
63	219	0.13	1	1
64	219	0.13	1	1
65	218	0.12	1	1
66	215	0.10	1	1
67	206	0.04	1	1
68	204	0.03	1	1
69	204	0.03	1	1
70	204	0.03	1	1
71	203	0.02	1	1
72	200	0.00	0	0
	TOTALS	349	446	414

Note that the attacker's optimal strategy is not determined here. For a given number k of attacking missiles, his problem is

$$\text{maximize} \quad V(\overline{d},a)$$
$$\text{subject to}$$
$$\sum_{i=1}^{T} a(i) \leq k,$$

where

$$V(\overline{d},a) = \sum_{i=1}^{T} v(i) \left(1 - \prod_{j=1}^{a(i)} \left(1 - (0.8)^{\overline{d}(i,j)}\right) \right),$$

with $\overline{d}$ as in Table 4. This is a non-linear "knapsack" type problem whose solution could be found by dynamic programming.

ACKNOWLEDGMENTS

The authors wish to thank Dr. Lowell Bruce Anderson for his careful reading of earlier drafts of the paper and for his numerous and insightful suggestions throughout the study. Dr. Jerome Bracken was extremely influential in directing the authors' efforts and providing intuitive guidance and encouragement. Finally, Dr. Richard Soland contributed a number of helpful comments that have been incorporated into the final version.

REFERENCES

1. Director of Defense Research and Engineering, Damage Limiting: A Rationale for the Allocation of Resources by the US and the USSR, 21 January 1964 (recently declassified; study directed by Glenn A. Kent).

2. Eckler, A. Ross, and Stefan A. Burr, Mathematical Models of Target Coverage and Missile Allocation, Military Operations Research Society, 1972.

3. Kaplan, Fred, *Strategic Thinkers*, The Bulletin of the Atomic Scientists, Volume 38, Number 10, December 1982, pp. 51-56.

4. Karr, Alan F., Nationwide Defense Against Nuclear Weapons: Properties of Prim-Read Deployments, P-1395, Institute for Defense Analyses, April 1981.

5. [1]Read, W.T., Jr., Tactics and Deployment for Anti-Missile Defense (U), Bell Telephone Laboratories, Whippany, NJ, 1958. CONFIDENTIAL.

[1] No classified material from the text of this classified paper is used herein.

APPENDIX A

PROOF OF THE LEMMA

PROOF OF THE LEMMA

Lemma: Let $a = (a_i)$ and $b = (b_i)$ be a pair of positive, monotonically nonincreasing vectors of size t such that

$$\text{(a)} \quad \sum_{i=1}^{s} a_i > \sum_{i=1}^{s} b_i \qquad \text{for } s = 1,2,\ldots, t-1 \ ;$$

$$\text{(b)} \quad \sum_{i=1}^{t} a_i = \sum_{i=1}^{t} b_i .$$

Then for $q \in (0,1)$,

$$\prod_{i=1}^{t} \left(1-q^{a_i}\right) < \prod_{i=1}^{t} \left(1-q^{b_i}\right) .$$

Proof: We first show that there exists a $t \times t$ doubly stochastic matrix M such that

$$b = Ma.$$

Let

$$M_1 = \left(\frac{1}{a_1-a_2}\right) \begin{pmatrix} b_1-a_2 & a_1-b_1 & 0 & \cdots & 0 \\ a_1-b_1 & b_1-a_2 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} .$$

Note that M_1 is doubly stochastic. Let $a^1 = M_1 a$ and note that

$$a_1^1 = b_1$$

$$a_2^1 = a_1 + a_2 - b_1 > b_2$$

$$a_3^1 = a_3$$

$$\vdots$$

$$a_t^1 = a_t \ .$$

In general, we will have a vector a^{k-1} such that

$$a_i^{k-1} = b_i \qquad \text{for } i = 1,\ldots,k-1$$

$$\sum_{i=1}^{s} a_i = \sum_{i=1}^{s} a_i^{k-1} > \sum_{i=1}^{s} b_i \qquad \text{for } s = k,\ldots,t-1$$

$$\sum_{i=1}^{t} a_i^{k-1} = \sum_{i=1}^{t} b_i$$

and if we pre-multiply such a vector by the doubly stochastic matrix

$$M_k = \left(\frac{1}{a_k^{k-1} - a_{k+1}^{k-1}}\right) \left(\begin{array}{c|cc|c} I_{k-1} & 0 & 0 & 0 \\ \hline 0 & b_k - a_{k+1}^{k-1} & a_k^{k-1} - b_k & 0 \\ 0 & a_k^{k-1} - b_k & b_k - a_{k+1}^{k-1} & 0 \\ \hline 0 & 0 & 0 & I_{t-k-1} \end{array}\right)$$

we obtain a new vector a^k which has the above properties, until $k = t - 1$, when $a^{t-1} = b$.

Since the product of doubly stochastic matrices remains doubly stochastic, the matrix

$$M = M_{t-1} \cdots M_1$$

has this property, and also

$$b = Ma .$$

Let $M = (\lambda_{ij})$ so that

$$b_i = \sum_{j=1}^{t} \lambda_{ij} a_j ,$$

where

$$\sum_{j=1}^{t} \lambda_{ij} = 1, \quad \lambda_{ij} \geq 0 .$$

Since the function $\ell n(1-q^z)$ is strictly concave as a function of z, we have

$$\ell n(1-q^{b_i}) > \sum_{j=1}^{t} \lambda_{ij} \; \ell n\left(1-q^{a_j}\right)$$

$$= \ell n \prod_{j=1}^{t} \left(1-q^{a_j}\right)^{\lambda_{ij}} ,$$

so that

$$\sum_{i=1}^{t} \ell n(1-q^{b_i}) > \sum_{i=1}^{t} \ell n \prod_{j=1}^{t} \left(1-q^{a_j}\right)^{\lambda_{ij}}$$

$$= \ell n \prod_{i=1}^{t} \prod_{j=1}^{t} \left(1-q^{a_j}\right)^{\lambda_{ij}}$$

$$= \ell n \prod_{j=1}^{t} \left(1-q^{a_j}\right)^{\sum_{i=1}^{t} \lambda_{ij}} .$$

But since M is doubly stochastic, $\sum_i \lambda_{ij} = 1$, so that

$$\prod_{i=1}^{t}\left(1-q^{b_i}\right) > \prod_{j=1}^{t}\left(1-q^{a_j}\right) ,$$

which completes the proof. $\square$

We note the dependency on log-concavity in this proof. This property arises in several instances, and seems to be an essential feature of the problem. See also Appendix B.

APPENDIX B

ANOTHER ALGORITHM FOR THE ALL-INTEGER VERSION OF THE PRIM-READ MODEL

ANOTHER ALGORITHM FOR THE ALL-INTEGER VERSION OF THE PRIM-READ MODEL

1. Introduction

Falk [3] has given an algorithm for finding an optimal all-integer solution of the Prim-Read allocation model. We present here another such algorithm. This algorithm requires time proportional to log ρ, where ρ is the price; the algorithm of [3] requires time proportional to ρ. Thus, the present algorithm is probably superior for hand computation at least; for machine computation, both are so fast that there is presumably no great advantage. However, our main purpose in presenting another algorithm is not computation but insight about the validity of the original (non-integral) Prim-Read model. This model has many very attractive properties (see [4]), but of course the non-integral assignments produced are not feasible. The integral assignments produced by our algorithm can be readily compared with the semi-integral model of [1]. The latter model can in turn be (somewhat less readily) compared with the original Prim-Read model. Our ultimate goal (among others) is to provide easily computed, guaranteed bounds for the optimal integer solution in terms of the Prim-Read solution. This note is the third (after [1,2]) of a series aimed at that goal.

2. Characterizing an Optimal Solution

Before giving the algorithm we must establish some results which will characterize the particular solution it will find. As usual, the optimization problem to be solved is as follows:

(2.1) $$\text{minimize } D = \sum_i d(i) ,$$

subject to the constraints

$$(2.2) \qquad V(d,a) = 1 - \prod_{i=1}^{a} (1-q^{d(i)}) \leq a/\rho ,$$

for all positive integers a, where the d(i) are required to be non-negative integers. As discussed in [1] and [2], ρ need not be an integer; of course, $\rho > 0$. By Proposition 3.1 of [1], there always exists a solution for which $d(j) \geq d(j+1)$ for all j, that is, a monotone solution.

We may describe such a solution, as in [1], by a sequence of positive integers $x_1 \geq x_2 \geq \ldots \geq x_\ell$:

$$(2.3) \qquad \begin{aligned} d(i) &= \ell , & 0 < i \leq x_\ell , \\ d(i) &= \ell-1, & x_\ell < i \leq x_{\ell-1} , \\ &\;\;\vdots \\ d(i) &= 2 , & x_3 < i \leq x_2 , \\ d(i) &= 1 , & x_2 < i \leq x_1 , \\ d(i) &= 0 , & x_1 < i . \end{aligned}$$

Of course, it is clear that at the optimum, $x_1 = \{\rho\} - 1$, where $\{\rho\}$ denotes the least integer $\geq \rho$. We also note that D is given by $x_1 + \ldots + x_\ell$.

Clearly, we could now restate the optimization problem (2.1-2) in terms of the x_k, and indeed this was done, for non-integral x_k, as formulas (4.1-4) of [1]. We do not do this now; but we do need the notion of an x_k not being tightly constrained. Call x_k in a feasible set of x_j _free_ if the set $\{x'_j\}$, defined by $x'_j = x_j$ for $\ell \geq j > k$, $x'_k = x_k - 1$, and $x'_j = x_1$ for $k > j \geq 1$, is also feasible. (Note that this entails $x_k > x_{k+1}$.) That is, the set of d's starts out the same, but the value k - 1 starts one step earlier, and then continues to the end. If $x_\ell = 1$, the effect of setting $x'_\ell = x_\ell - 1$ is the same as that of replacing ℓ by $\ell - 1$.

We will show that we can always find an optimal solution in which no x_k is free; we begin with a simple algebraic proposition. We note in passing that it is equivalent to the fact that $1 - q^t$ is log-concave in t.

<u>Proposition 2.1</u>: If $d \geq e + 1$ and $e \geq 0$, then

$$(1-q^d)(1-q^e) \leq (1-q^{d-1})(1-q^{e+1}) .$$

<u>Proof</u>:

$$\begin{aligned}
&(1-q^{d-1})(1-q^{e+1}) - (1-q^d)(1-q^e) \\
&\quad = - q^{d-1} - q^{e+1} + q^d + q^e \\
&\quad = q^e(1-q+q^{d-e}-q^{d-e-1}) \\
&\quad = q^e(1-q)(1-q^{d-e-1}) \geq 0 . \qquad \square
\end{aligned}$$

Armed with this, we can prove a crucial fact. We have to state the result a bit more technically than one would like because of the possibility that $x_{k-1} = x_{k-2}$ (in which case, no $d(i)$ is equal to $k - 2$).

<u>Proposition 2.2</u>: Let $x_1, \ldots, x_\ell$ define an optimal (feasible) solution to (2.1-2), as in (2.3), and suppose that x_k is free, where $k > 1$. Let k' be the largest value for which $x_{k'} = x_{k-1}$. (Usually, $k' = k - 1$). Then if x_k is replaced by $x_k - 1$ and $x_{k'}$ is replaced by $x_{k'}+1$, the resulting sequence $\{x'_j\}$ is again an optimal solution.

Proof: Since D is unchanged, we need only check that the resulting set of x_j is feasible. Consider first the (non-monotone) sequence of $d(i)$ that comes from (2.3) by setting $d(x_k-1) = k - 1$ and $d(x_{k-1}) = k$. Clearly D is unchanged. Moreover, this new solution obviously satisfies (2.2) for $a \geq x_{k-1}$, and it satisfies (2.2) for $a < x_{k-1}$ because x_k was free. Now we make this allocation monotone. Set $d(x_{k-1}) = k - 1$, $d(x_{k-1}+1) = k'$. By the definition of k', $x_{k-1} + 1 = x_{k'} + 1$ and k' is one greater than the old value of $d(x_{k-1}+1) = d(x_{k'}+1)$, so that the new assignment has the same stockpile D as before. But this assignment is also feasible by Proposition 2.1. Since this is just the assignment specified by the sequence $\{x_j'\}$, the proof is complete. $\square$

The above proposition leads immediately to the following theorem, which completely characterizes one solution to the all-integer problem.

Theorem 2.1: There is a unique feasible allocation of type (2.3) for which no x_k is free, and this allocation is optimal.

Proof: Clearly, there is only one choice of x_ℓ that is (feasible, but) not free. Once x_ℓ is specified, there is only one choice of $x_{\ell-1}$ that is not free, and so on. Therefore, the x_j are completely specified in turn by the conditions on them. (We will shortly consider how to calculate the x_j.) To see that this solution is optimal, we need only show that an optimal solution can be converted to it while remaining optimal.

Consider any such optimal solution. If no x_j is free, we are done. Let k be the largest index for which x_k is free. We cannot have $k = 1$, for then the solution would not have been optimal. By Proposition 2.2, it is possible to reduce x_k by one and still have an optimal solution. If the resulting x_k is still free, we may continue the process until it is no longer free. (If $k = \ell$, this could involve reducing x_ℓ to zero and hence replacing ℓ by $\ell - 1$.) In any case, this process leads to an optimal solution in which the largest j for which x_j is free is < k. If we continue to do this, we eventually arrive at an optimal assignment for which no x_j is free. This completes the proof. ∎

3. The Algorithm

Theorem 2.1 characterizes an optimal solution, and one may clearly derive an algorithm from it. Like Falk's algorithm [3], it is a greedy algorithm; but it is greedy in the x_j of (2.3), not in the d(i) of (2.1-2). Its solution will in general be different from Falk's, since it produces monotone d(i), but of course the resulting stockpile D will always be the same.

Of course, Theorem 2.1 does not give an algorithm in itself, since we must make explicit how to choose the x_k so that they are not free. We now state a complete algorithm. Although it looks tedious, the algorithm is actually rather easy to apply. We use the notations [t] and {t} to denote the greatest integer $\leq t$ and the least integer $\geq t$, respectively.

<u>Algorithm</u>: Set $\ell = \{-\log \rho/\log q\}$ and set $\delta_j = -\log(1-q^j)$, $j = 1, 2, \ldots, \ell$. Also, set $x_{\ell+1} = 0$, $\theta_\ell = 0$. Then, for $j = \ell$, $\ell - 1, \ldots, 2$ set in turn

$$(3.1)\qquad a_{j-1} = \quad - 1/ \;_{j-1} ,$$

$$(3.2)\qquad \phi'_{j-1} = \delta_{j-1}[a_{j-1}] + \log(1-[a_{j-1}]/\rho) ,$$

$$(3.3)\qquad \phi''_{j-1} = \begin{cases} \delta_{j-1}\{a_{j-1}\} + \log(1- \{a_{j-1}\}/\rho) & \text{if } \{a_{j-1}\}/\rho < 1 , \\ 0 & \text{otherwise.} \end{cases}$$

$$(3.4) \qquad x_j = \left\{ \frac{\max(\phi'_{j-1}, \phi''_{j-1}) - \theta_j}{\delta_{j-1} - \delta_j} \right\} ,$$

$$(3.5) \qquad \theta_{j-1} = \theta_j + (\delta_{j-1} - \delta_j) x_j \; .$$

Finally, set $x_1 = \{\rho\} - 1$. Note that the calculation of θ_1 may be omitted.

We must show that this works; we state that as a theorem.

Theorem 3.1: The above algorithm produces a feasible sequence of x_j such that none are free; they therefore represent an optimal allocation.

Proof: Of course, ℓ must be great enough so that the solution is feasible for the first attacker, so that we must have, from (2.2),

$$1/\rho \geq 1 - (1 - q^{d(1)}) = q^{\ell} ,$$

so that

$$- \log \rho \geq \ell \log q,$$

and hence

$$\ell \geq \{-\log \rho / \log q\} \; .$$

On the other hand, if ℓ is strictly greater than this, x_ℓ must be free since even $x_\ell = 0$, $x_{\ell-1} = \{\rho\} - 1$ is feasible. Hence, $\ell = \{-\log \rho / \log q\}$.

The remainder of our proof is closely related to ideas developed in Section 4 of [1]. In particular, $V(a,d)$ for any a in the range $x_j < a \leq x_{j-1}$ is given, for some θ, by

$$V(a,d) = 1 - \exp(\theta - \delta_{j-1}a) \ ,$$

since

$$V(a,d) = 1 - \prod_{i=1}^{x_j} (1-q^{d(i)}) \cdot (1-q^{j-1})^{a-x_j}$$

$$= 1 - \prod_{i=1}^{x_j} (1-q^{d(i)}) \cdot (1-q^{j-1})^{-x_j} \cdot \exp(-\delta_{j-1}a) \ ,$$

where we take θ to be the natural logarithm of the product and the following factor. We will set $\theta = \theta_{j-1}$ in any such range; we must show that θ_{j-1} is as defined in the algorithm. Of course, $\theta_\ell = 0$ in the first range, as it should be. For any other range, x_j is the value of a at the intersection of the curves $v = 1 - \exp(\theta_{j-1} - \delta_{j-1}a)$ and $v = 1 - \exp(\theta_j - \delta_j a)$; this easily leads to

$$a = x_j = \frac{\theta_{j-1} - \theta_j}{\delta_{j-1} - \delta_j} \ , \tag{3.6}$$

which is equivalent to (3.5). (Formula (3.6) already occurs in the proof of Theorem 4.1 of [1], except that there it was stated in terms of $\gamma_j = -\theta_j$.)

Of course, before we can determine θ_{j-1}, we must determine x_j. We must choose x_j as small as possible consistent with the requirement that the curve $v = 1 - \exp(\theta_{j-1} - \delta_{j-1}a)$ lies on or below the line $v = a/\rho$ at all integer a. If they were tangent, the point of tangency would be at $a_{j-1} = \rho - 1/\delta_{j-1}$; this is formula (4.5) of [1]. However, since only integer values of a are to be considered, we need that the curve lies on or below the line at $a = [a_{j-1}]$ and $a = \{a_{j-1}\}$, unless $\{a_{j-1}\} \geq \rho$, in which case this point should be ignored. If the curve $v = 1 - \exp(\theta - \delta_{j-1}a)$ goes through the point $(a,v) = (a, a/\rho)$, then we have

$$\exp(\theta - \delta_{j-1}a) = 1 - a/\rho \ ,$$

so that

(3.7) $$\theta = \delta_{j-1}a + \log(1-a/\rho) \ .$$

From this we see that ϕ'_{j-1} and ϕ''_{j-1} in (3.2-3) are the θ of (3.7) corresponding to $a = [a_{j-1}]$ and $a = \{a_{j-1}\}$, respectively, provided $\{a_{j-1}\} < \rho$. Consequently $\theta = \max(\phi'_{j-1},\phi''_{j-1})$ is the smallest θ for which $v = 1 - \exp(\theta-\delta_{j-1}a)$ does not go above $v = a/\rho$ at any integer point. (Note that, in the case $\{a_{j-1}\} \geqq \rho$, $\theta = \phi'_{j-1}$ since the latter is positive, which is as it should be.) Hence, by (3.6), we need to have, and need only have,

$$x_j \quad \frac{\max(\phi'_{j-1},\phi''_{j-1}) - \theta_j}{\delta_{j-1} - \delta_j}$$

in order for x_j to produce a feasible solution. Therefore, if we take x_j to be as small as possible, but still an integer, x_j will not be free. This gives us (3.4). Finally, it is clear that $x_1 = \{\rho\} - 1$. This completes the proof. $\square$

The form of (3.1-5) permits rather ready comparison with the results of the algorithm for the semi-integral model presented in Theorem 4.2 of [1]. This fact is a primary motivation for developing the above algorithm; however, such a comparison will be deferred to a later note.

4. Numerical Examples

We discuss here two numerical examples, hand-calculated. In a later note we will give more extensive numerical results produced with the aid of a computer. First, consider the case $q = .25$, $\rho = 50$, considered in [1] and [3]. We have $\ell = 3$, and $\delta_1 = .2877$, $\delta_2 = .06454$, $\delta_3 = .01575$, $x_4 = 0$, $\theta_3 = 0$. Then,

$a_2 = 34.51$,

$\phi'_2 = 1.0549$, $\phi''_2 = 1.0549$,

$x_3 = \{21.62\} = 22$,

$\theta_2 = 1.0734$,

$a_1 = 46.52$,

$\phi_1' = 10.708$, $\phi_1'' = 10.708$,

$x_2 = \{43.17\} = 44$,

and, finally, $x_1 = 49$. The total stockpile is $D = 22 + 44 + 49 = 115$.

Some remarks are in order here. First, we have left off the unnecessary calculation of θ_1. Second, the stockpile agrees with the corresponding one in [3], as it must. Moreover, the numbers of 3's, 2's, and 1's tally exactly with those in [3], but of course the sequence is different. It seems likely that such an exact tally should always occur, but this has not been proved. Finally, note the surprising fact that in each case, $\phi_j' = \phi_j''$. This is not a coincidence; in fact we will prove in the last section that it always happens when $\rho - q^{-j}$ is an integer. Thus, when $q = .5, .25, .2, .1, \ldots$ and ρ is an integer, this applies, a useful fact for hand calculation.

Now we let $q = .2$, $\rho = 50$, again considered in [1] and [3]. We have $\ell = 3$, $\delta_1 = .2231$, $\delta_2 = .04082$, $\delta_3 = .008032$, $x_4 = 0$, $\theta_3 = 0$. Then,

$a_2 = 25.50$,

$\phi_2' = \phi_2'' = .3274$,

$x_3 = \{9.985\} = 10$,

$\theta_2 = .3279$,

$a_1 = 45.51$,

$\phi_1' = \phi_2'' = 7.737$,

$x_2 = \{40.65\} = 41$,

and $x_1 = 49$. This time, $D = 10 + 41 + 49 = 100$.

As before, even the individual x_j agree exactly with the corresponding numbers of 3's, 2's, and 1's in [3].

5. Conditions for a Simplified Calculation

It has been already said that $\phi_j' = \phi_j''$ whenever $\rho - q^{-j}$ is an integer. Although this is a rare event for ρ and q completely arbitrary, it will often occur for the values typically chosen, simplifying hand calculation. For this reason, it is worthwhile to prove this, and a little more, explicitly.

Proposition 5.1: For $1 \leq j \leq \ell$, it will happen that $\phi_j' = \phi_j''$ in the algorithm of Section 3 if and only if either $\rho - 1/\delta_j$ or $\rho - q^{-j}$ is an integer.

Proof: If $\rho - 1/\delta_j = a_j$ is an integer, then $[a_j] = \{a_j\}$ and the result is trivial. Otherwise, we may set $[a_j] = a$, $\{a_j\} = a + 1$. In this case, $\phi_j' = \phi_j''$ yields

$$0 = \phi_j'' - \phi_j' = \delta_j(a+1) + \log(1 - (a+1)/\rho) - \delta_j a \log (1-a/\rho)$$

$$= \delta_j + \log \frac{\rho - a - 1}{\rho} - \log \frac{\rho - a}{\rho}$$

$$= \delta_j + \log \frac{\rho - a - 1}{\rho - a} ,$$

so that

$$-\delta_j = \log \frac{\rho - a - 1}{\rho - a} .$$

But $\delta_j = -\log(1-q^j)$, so

$$1 - q^j = \frac{\rho - a - 1}{\rho - a} = 1 - \frac{1}{\rho - a} ,$$

which is equivalent to

$$a = \rho - q^{-j} .$$

Since a is an integer it is necessary that $\rho - q^{-j}$ be an integer in this case.

To prove that this condition is also sufficient, we need only show that

$$\rho - q^{-j} \le \rho - 1/\delta_j \le \rho - q^{-j} + 1 ,$$

which is equivalent to

$$q^j \le \delta_j \le 1/(-q^{-j}+1) = q^j/(1-q^j) .$$

But since $0 < q^j < 1$, we may expand $\delta_j = -\log(1-q^j)$ and $q^j/(1-q^j)$ in series; the above is then equivalent to

$$q^j \le q^j + q^{2j}/2 + q^{3j}/3 + \ldots$$

$$\le q^j + q^{2j} + q^{3j} + \ldots ;$$

this is obvious, so the proof is complete. $\square$

REFERENCES

[1] Burr, Stefan A., Feasible Approximations to Prim-Read Deployments, Draft Paper, October 1981.

[2] Burr, Stefan A., Two Notes on the Prim-Read Deployment Model, Draft Paper, December 1981.

[3] Falk, James E., Integer Prim-Read Firing Doctrines Via the Greedy Algorithm, Draft Paper, January 1982.

[4] Karr, Alan F., Nationwide Defense Against Nuclear Weapons: Properties of Prim-Read Deployments, P-1395, Institute for Defense Analyses, April 1981.