NATIONAL BUREAU OF STANDARDS REPORT

9977

VALUATION OF TELECOMMUNICATIONS

Progress Report for FY 68
with Appendices

to

Defense Communications Agency

NBS

U.S. DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS

THE NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] provides measurement and technical information services essential to the efficiency and effectiveness of the work of the Nation's scientists and engineers. The Bureau serves also as a focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To accomplish this mission, the Bureau is organized into three institutes covering broad program areas of research and services:

THE INSTITUTE FOR BASIC STANDARDS . . . provides the central basis within the United States for a complete and consistent system of physical measurements, coordinates that system with the measurement systems of other nations, and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. This Institute comprises a series of divisions, each serving a classical subject matter area:

—Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic Physics—Physical Chemistry—Radiation Physics—Laboratory Astrophysics[2]—Radio Standards Laboratory,[2] which includes Radio Standards Physics and Radio Standards Engineering—Office of Standard Reference Data.

THE INSTITUTE FOR MATERIALS RESEARCH . . . conducts materials research and provides associated materials services including mainly reference materials and data on the properties of materials. Beyond its direct interest to the Nation's scientists and engineers, this Institute yields services which are essential to the advancement of technology in industry and commerce. This Institute is organized primarily by technical fields:

—Analytical Chemistry—Metallurgy—Reactor Radiations—Polymers—Inorganic Materials—Cryogenics[2]—Office of Standard Reference Materials.

THE INSTITUTE FOR APPLIED TECHNOLOGY . . . provides technical services to promote the use of available technology and to facilitate technological innovation in industry and government. The principal elements of this Institute are:

—Building Research—Electronic Instrumentation—Technical Analysis—Center for Computer Sciences and Technology—Textile and Apparel Technology Center—Office of Weights and Measures—Office of Engineering Standards Services—Office of Invention and Innovation—Office of Vehicle Systems Research—Clearinghouse for Federal Scientific and Technical Information[3]—Materials Evaluation Laboratory—NBS/GSA Testing Laboratory.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D. C., 20234.

[2] Located at Boulder, Colorado, 80302.

[3] Located at 5285 Port Royal Road, Springfield, Virginia 22151.

NATIONAL BUREAU OF STANDARDS REPORT

NBS PROJECT
4556455

NBS REPORT
9977

VALUATION OF TELECOMMUNICATIONS

Progress Report for FY 68
with Appendices

by
S. C. Strom
Arthur M. Hobbs

to
Defense Communications Agency

Systems Research and Development Division
Center for Computer Sciences and Technology

NBS

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

TABLE OF CONTENTS

Project 4556455

ANNUAL PROGRESS REPORT

for FY 1968

PROJECT TITLE: Valuation of Telecommunications

OBJECTIVE: a. To develop effective methods for calculating the attack worth of the various elements.

b. To develop optimal offensive and defensive strategies for destruction and/or preservation of communication capabilities.

HISTORY: The project is a continuation of a previous effort under contract with the Defense Communications Agency. The contract has been in effect since July 1966.

National Bureau of Standards subcontracted The John Hopkins University to conduct research in graph theoretic methods for analyzing large communications networks.

PROGRESS during FY 1968:

Research and development was concentrated in two general areas.

1. Determination of importance numbers for elements of a communications network.

2. Development of offensive and defensive strategies for a telecommunications network serving a multi-command multi-subordinate structure.

A pathfinding and counting method and a technique of assignment of importance numbers to network elements were developed during the previous period. Efficient computer programs capable of handling real size networks had to be written and made operational. These programs had to be provided with program interrupt and restart routines. The difficulties caused by the size of the network and the necessarily excessive running time were satisfactorily resolved.

Considerable effort was devoted to finding an acceptable method for determination of the worth value, or importance number, of individual elements of the network.

Consultative services of Dr. M. J. Krakowski, of Tulane University, New Orleans, were secured during summer 1967. The results of Dr. Krakowski's investigation are contained in a paper "Some Considerations in Valuation of a Telecommunications Network" which is being published by NBS. Briefly summarized, at the present state of art the problem has no satisfactory numeric solution. Integer linear programming seems to be a promising tool for finding at least partial answers.

Our own efforts were concentrated on the study of the relationship of path lengths to path counts and on the relationship between the numbers of paths utilizing a network element and its relative importance value.

An acceptable method was developed to perform such relative evaluation and to produce numbers which allow us to rank network elements in an approximate order of importance. The results obtained through this method were tested against minimum cut sets (disconnecting sets) in the network and a considerable degree of correlation was found. The study is described in the paper "Comparison of Evaluation Techniques," included as Appendix A.

The above work was accompanied by background theoretical investigations in the related topics of graph theory. In the course of these investigations several new theorems on thickness-preserving compositions of graphs and on the size of cut sets in t-minimal graphs were developed. Thus abstract scientific research and its practical implementation were successfully pursued.

During the course of our research, a new mathematical problem was found whose solution would give directly the necessary size and structure of an optimal military telecommunications network which uses a civilian network. The most general form of this problem asks what graphs can be embedded in other graphs and includes several previously known and still unsolved problems. However, the project requires only the the solution of the special case whose exact statement is given in Appendix B, and this version of the problem may include sufficient restrictions to be solvable. The Johns Hopkins University group has already made some progress toward a solution.

Dr. Mandell Bellmore, of The Johns Hopkins University, with a team of graduate students continued this efforts with research concentrated in the area of multi-command problems. Single-command attacker's and defender's problems had been successfully solved previously. However, those methods of solution could not be simply extended to the multi-command situation; therefore, new approaches were sought and some progress was made. The problem of an attacker with unlimited resources was solved and a computer program implementing the solution was developed. The defender's problem was theoretically solved; its translation into a computer program remains to be done.

APPENDIX A

COMPARISON OF EVALUATION TECHNIQUES

by

Arthur M. Hobbs

1.0 INTRODUCTION

Every large communications system must be designed with considerable care, taking into account a large number of factors, such as the needed capacity and the failure rates of components. An important factor in military communications networks is the survivability, or the ability of the network to serve its purpose in the event of an enemy attack or a natural disaster.

In the present study, we have concerned ourselves solely with the survivability of large communications networks, containing 40 to 80 switches and 300 or more inter-switch links. We would like to assign "importance numbers" to the elements (links and switches) of the networks so that the more important to us that an element survive in an attack, the larger the importance number assigned to it.

In any reasonably complicated network there are numerous paths by which each pair of subscribers might communicate, each path differing from the others by one or more network elements. Therefore, it is unlikely that the removal of any one, or even a small number, of elements will adversely affect the existence of communication paths between users of the network (although the effect of the removal on capacity or quality of communications may be serious indeed). For this reason it seems unlikely that we can find any absolute valuation of the network elements, or any importance numbers whose values have an absolute meaning.

We have written a pair of computer programs (PATHFINDER and WORTH, referred to hereafter as PATHFINDER-WORTH) which assign numbers to the network elements. Although we have referred to these numbers in previous documents (for example, in reference [3]) as "importance numbers", we hope merely that these numbers can properly be used to place the network elements in rank order, so that elements highest on the list are the most valuable to us and those lowest on the list the least valuable. To test this hope, we have compared the ordering obtained through PATHFINDER-WORTH with the ordering resulting from arranging the elements randomly, and with a later described ordering dependent upon cut sets. We believe that if the orderings given by the two different methods, PATHFINDER-WORTH and cut sets, are similar and differ markedly from the random ordering, then we have reason to believe that the ordering given by PATHFINDER-WORTH is useful.

The author wishes to thank James Owen for his help in performing many of the calculations reported here and in preparing the graphs included.

1.1 DEFINITIONS AND NETWORK CONDITIONS

A communications network is a collection of switching stations, or switches, interconnected by inter-switch communications links, each containing one or more channels for communications. The switches are joined to subscribers, or users of the network, by access lines. The network links and switches are called network elements. We have assumed for this study that no subscriber is also a switch; this assumption is usually valid, there being presently at most one exception in the networks we are examining. Such exceptions can usually be treated by an artificial separation of the subscriber-switch into two parts, one a switch joined to all other elements to which the subscriber-switch is joined, and the other a subscriber joined to that switch by an ideal link.

In this study, we have assumed that a given pair of subscribers can communicate after an attack as long as there is a path still existing between them which does not have too many links in tandem, regardless of the remaining capacity of the network as a whole and regardless of the number of other subscribers also attempting to use the network at the same time. This assumption is reasonable in the special conditions of military networks having priority levels for important subscribers.

We are interested in analyzing a large network which is shared by many independent users, each with his own time-varying importance to national defense. For the purposes of the remainder of this paper, we will call each pair of subscribers who are communicating through our network a command-subordinate pair, whatever their real-life functions. Further, we will assume that, at least for short periods of time T, it is possible to assign a weight, or worth number, to each pair, with increasing worth numbers corresponding to some increasing importance values (e.g., national defense).

A collection of network elements is a cut set of the network if the removal of these elements from the network results in the inability of a set S of one or more command-subordinate pairs to communicate and no subcollection of the elements separates all of the pairs S. A cut set C may properly contain another cut set C' which separates some, but not all, of the command-subordinate pairs separated by C. For example, in Figure 1, the access links (S11, 1), (S11, 4), and (S12, 3) are a cut set separating C1 from both S11 and S12. But (S11, 1) and (S11, 4) also form a cut set, properly contained in the previous one, which cuts C1 from just S11.

Our networks have one additional property which has not been much considered in the past: for voice communications, there is a limited number of links in tandem which may be used by a communication path.

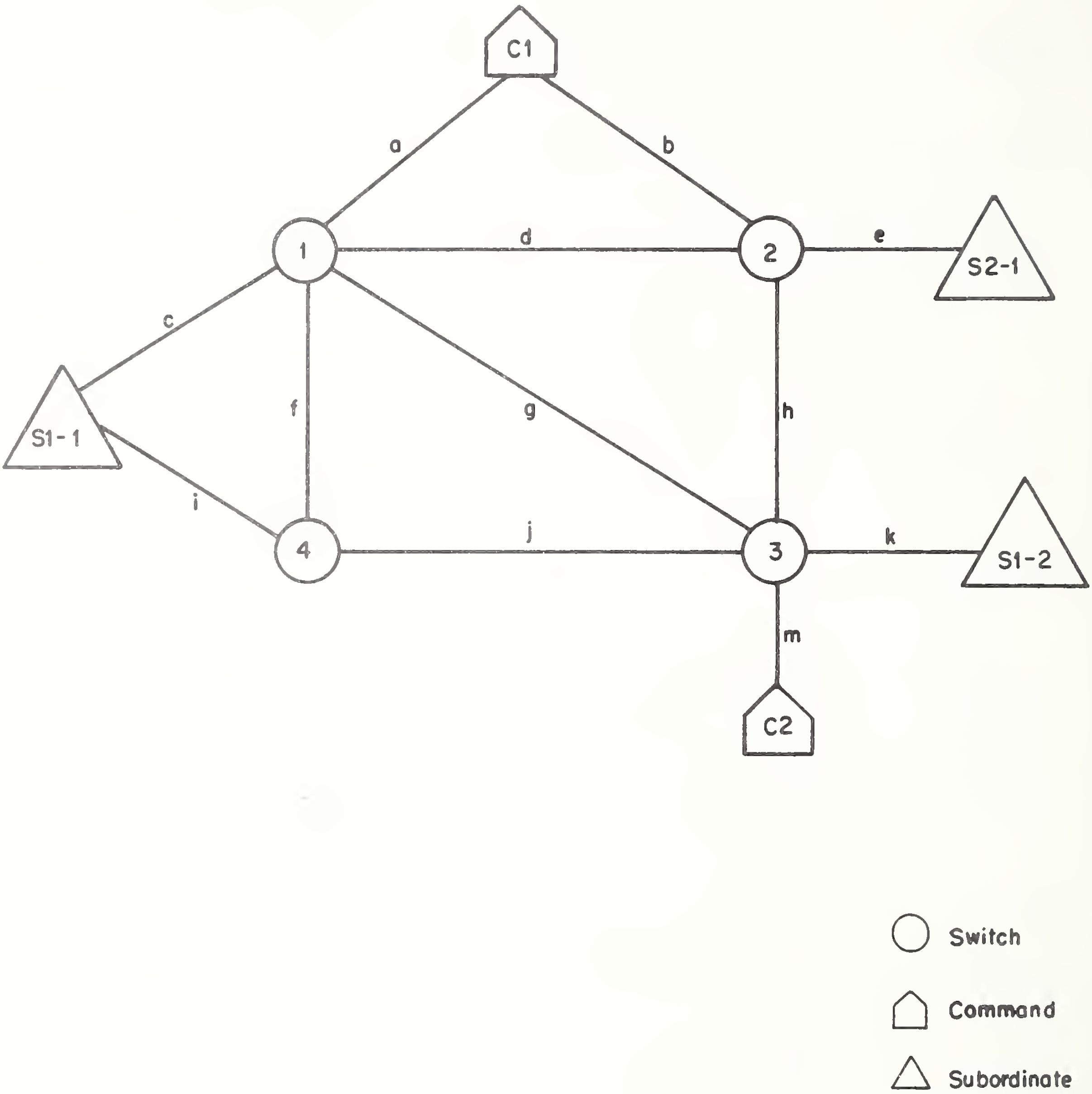


FIGURE 1. FOUR SWITCH NETWORK I

Quality of communications deteriorates very rapidly beyond this number and it becomes impossible to talk over paths with more links. This maximum number is much smaller than the total number of links in the network, and thus only a small fraction of the total theoretical number of paths in the network are usable for communications. This property has implications for cut sets; there will be many more cut sets in the restricted path length network than in a network without the restriction, since a set of elements whose removal does not cut all paths may cut all short paths. On the other hand, the property drastically reduces the number of paths requiring individual analysis (if such is to be done).

Very few of the theoretical results which have been proven for general networks hold true when we are restricted to just some of the paths. For this reason, we have been forced into experimental analysis of examples which we can investigate by hand. In each of the examples used, we obtain weights for the elements of the network and use the weights to give us an ordering of the elements.

2.0 EVALUATION METHODS

Several methods of weighting the network elements have been proposed. For example, Professor Martin Krakowski [2] has proposed that the elements be weighted by the amount of insurance which should be bought for them in an analogous commercial system, with the amount of insurance being obtained by means of a linear program. This method is sufficiently documented in the report cited; we will not repeat the description here.

Two additional methods for evaluating the network elements have been proposed. Both of these methods make the assumption that the command-subordinate worths for different pairs are independent, and that the numbers obtained for an element of the network using different command-subordinate pairs can be simply added to obtain an importance number for the element. This second assumption has been thoroughly discussed in the report by Professor Krakowski cited as reference [1]; suffice it to say here that the assumption is a dangerous one. However, no adequate number-assigning scheme has been proposed which does not make this assumption, other than the insurance scheme. Further, if the worths assigned to different pairs are actually independent, the additivity assumption need not be fatal to our weighting schemes.

2.1 CUT SET METHOD

One method proposed for evaluating the network consists of computing the following number for each of the network elements (switches and links) $\underline{e}$:

$$v(\underline{e}) = \sum_{C} \left(\frac{v(C)}{n(C)} \right) \Big/ N(\underline{e}) \qquad (1)$$

where C ranges over all of the cut sets in the network which include network element $\underline{e}$;

v(C) is the sum of the worth numbers of the command-subordinate pairs which can no longer communicate after the elements of the cut set C are removed from the network;

n(C) is the number of elements in the cut set C; and

N($\underline{e}$) is the number of cut sets in which $\underline{e}$ appears.

The rationale for this formula is as follows: We have no way of determining in advance what cut set might be used by forces attacking the network. It is therefore plausible that we should assign to each network element the average of the values assigned to it by each of the possible attacks C in which it appears.

If the removal of an element causes the total separation of all communications, that element can appear in only one cut set. But we would like the value of such an element to be the total of the worths assigned to all command-subordinate pairs. Since in this case $N(\underline{e})$ would be 1, and $n(C)$ would be 1, the formula given would yield this result. It is not reasonable to average over all cut sets for each element whether or not it appears in the cut sets, simply because we have restricted ourselves to cut sets rather than allowing all separating sets of network elements to be used.

Further, an element which appears in a cut set with only a small number of other elements in the set would intuitively seem to deserve a value larger than an element which appears in a cut set with the same value but which contains many other network elements. After all, the element in the smaller cut set is doing more of the work of separating the network. We take this consideration into account by dividing each $v(C)$ by $n(C)$, thus normalizing for the size of cut sets.

To apply this evaluation scheme to a communications network requires that all proper cut sets for one or more of the command-subordinate pairs be found and evaluated, and that the indicated computation be performed. Unfortunately, there is not yet any method by which the required searching and computing can be carried out in any reasonable amount of time. Most methods now require that numerous candidates for cut sets be found, many rejected, and the rest evaluated. This process is at best enormously time consuming. Thus, although this cut set scheme for evaluating the network elements seems in many ways to be a reasonable method, we have not tried to implement it in the form of a computer program.

2.2 PATHFINDER-WORTH

A second evaluation scheme has been programmed for use on the networks we are concerned with. This method takes advantage of the maximum path length for usable communications in a real voice communications network to reduce computational labor (in contrast to the cut set method above, which becomes even more laborious when this restriction is added). The method used here is to evaluate the following formula for each network element $\underline{e}$:

$$v(\underline{e}) = \sum_{(c,\ s)} \left(\frac{w(c,\ s)}{t(c,\ s)}\ n(c,\ s,\ \underline{e}) \right), \qquad (2)$$

where the summation is carried out over all command-subordinate pairs (c, s), and in which

w(c, s) is the command-subordinate worth for the pair (c, s);

t(c, s) is the total number of <u>usable</u> paths joining command c with subordinate s; and

n(c, s, <u>e</u>) is the number of those paths which pass through element <u>e</u>.

The concept used in obtaining this formula is that each <u>usable path</u> through a network element contributes a value to the element dependent solely on what that path connects and how many other paths perform the same connecting function. If only one path connects command c with subordinate s, that path must be very important, and any network element on it should be important as well. We therefore assign the path, and thus every element on the path, the full command-subordinate worth of the pair that the path joins. If, on the other hand, the path is only one of many paths joining the command-subordinate pair, cutting this one path will matter little to the pair. Therefore, we should assign the path only a small value. We have chosen to assign each path joining the command-subordinate pair (c, s) simply the worth w(c, s) divided by the number of paths t(c, s) joining the pair. We then multiply this value of one path by the total number of the paths joining the command-subordinate pair and passing through a network element to find the contribution of the command-subordinate pair to the element. Thus, if an element happens to be included in all or most of the paths between a command-subordinate pair, it will receive all or most of the worth of that pair, since removing the one element would destroy all or most of the paths joining the pair. Finally, carrying out our assumption of additivity, we simply add all of the values obtained for a network element for the different command-subordinate pairs together, thus obtaining a final value for each of the elements.

The above computation requires the finding and counting of every usable path between every command-subordinate pair. Although this requires a great deal of work, here the path length limitation helps, rather than hinders, the counting process. In addition, several techniques have been developed [3] to render the counting process even faster. It is for this reason, that we have chosen to implement this method.

2.3 PATH WEIGHTING

Since none of the importance number assignment methods can be shown to always yield perfect orderings of the elements of the network, a relatively easy change was made in the pathfinding computer program PATHFINDER which allowed each path as it was found to be weighted by a value based on the length of the path. The path lengths and values to be assigned to paths of those lengths are included in the data of the program, and can be modified from run to run.

One reason we chose to do this was that a communicator will usually prefer a short path over a longer one, given a choice. His reasons include lower noise on shorter paths, lower vulnerability on shorter paths (there being fewer network elements to fail or be destroyed), and smaller strain placed on the network's overall capacity by the use of a short path instead of a long one. It therefore seems reasonable that a short path should be worth more than a long one.

If we change t(c, s) to be the sum of the weights assigned to the full set of paths between command c and subordinate s, and n(c, s, e) to be the sum of the weights assigned to the set of paths between command c and subordinate s which pass through element e, then the above formula (2) gives a value of the element e which depends strongly on whether most of the paths passing through it are short or long.

Since n(c, s, e) and t(c, s) are both found in the pathfinding program simply by adding a "1" to the appropriate numbers whenever a new path is found, the substitution of weighted paths can be made simply by adding the path weight instead of the "1". This is precisely the change made in the program. Of course, it is perfectly feasible to impose the weight "1" on every path, as well as any other weight, thus retaining the original formulation of the formula (2) as a special case.

3.0 TEST CASES

Since only the pathfinding evaluation scheme has been programmed on the computer, in order to compare its results with those of the cut set scheme we were forced to limit ourselves to very small networks. This limitation imposes a restriction on the validity of our results for the very large networks in which we are interested; we discuss the validity question in more detail in Section 4 of this paper.

We chose one four switch network (shown in figures 1 and 2) and one five switch network (figure 3) for our study. Each of these networks was provided with two independent commands, one command with two subordinates and one with one subordinate. We connected the commands and subordinates to the four switch network in two different ways. Because of the enormous number of cut sets (424) for even so small a network as one with five switches, we joined the commands and subordinates to the five switch network in only one way. To free the results of any bias introduced by making one command-subordinate pair overwhelm the others, and to make the intuitive analysis of the networks easier, we assigned a command-subordinate worth of 10 to each of the pairs in each of the examples. We wanted to include the effect of a maximum usable path length which was less than the length of the longest path in the networks. Since the networks were so small, this desire forced us to use a path length of 3 switches, which is the shortest interesting path length less than the longest path length (4) in the four switch networks.

3.1 CUT SET EVALUATION

To evaluate the network elements using the cut set method, it was first necessary for us to find all of the cut sets in those networks. It should be remembered that a cut set is a set of network elements (links and switches) which separate one or more of the command-subordinate pairs (c, s). Furthermore, no proper part of the cut set (a sub-set with fewer elements) successfully separates the <u>same</u> command-subordinate pairs (c, s) (although it <u>might</u> still separate some of them). We have therefore classified the cut sets found by the list of command-subordinate pairs which each separates. None of these cut sets appears more than once. These lists appear in Sections 6.1, 6.2, and 6.3, corresponding respectively to the first and second four switch networks and the five switch network.

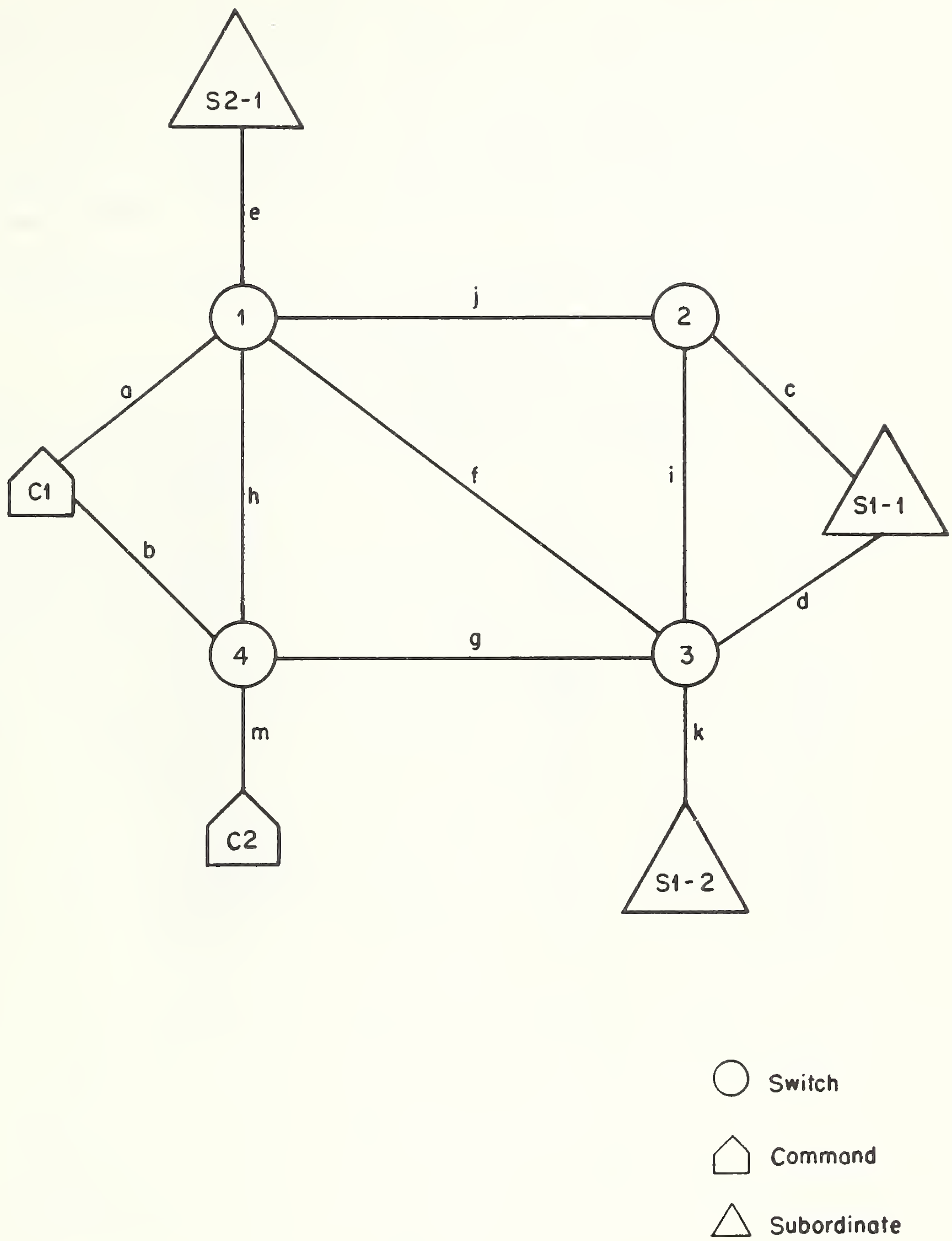


FIGURE 2. FOUR SWITCH NETWORK II

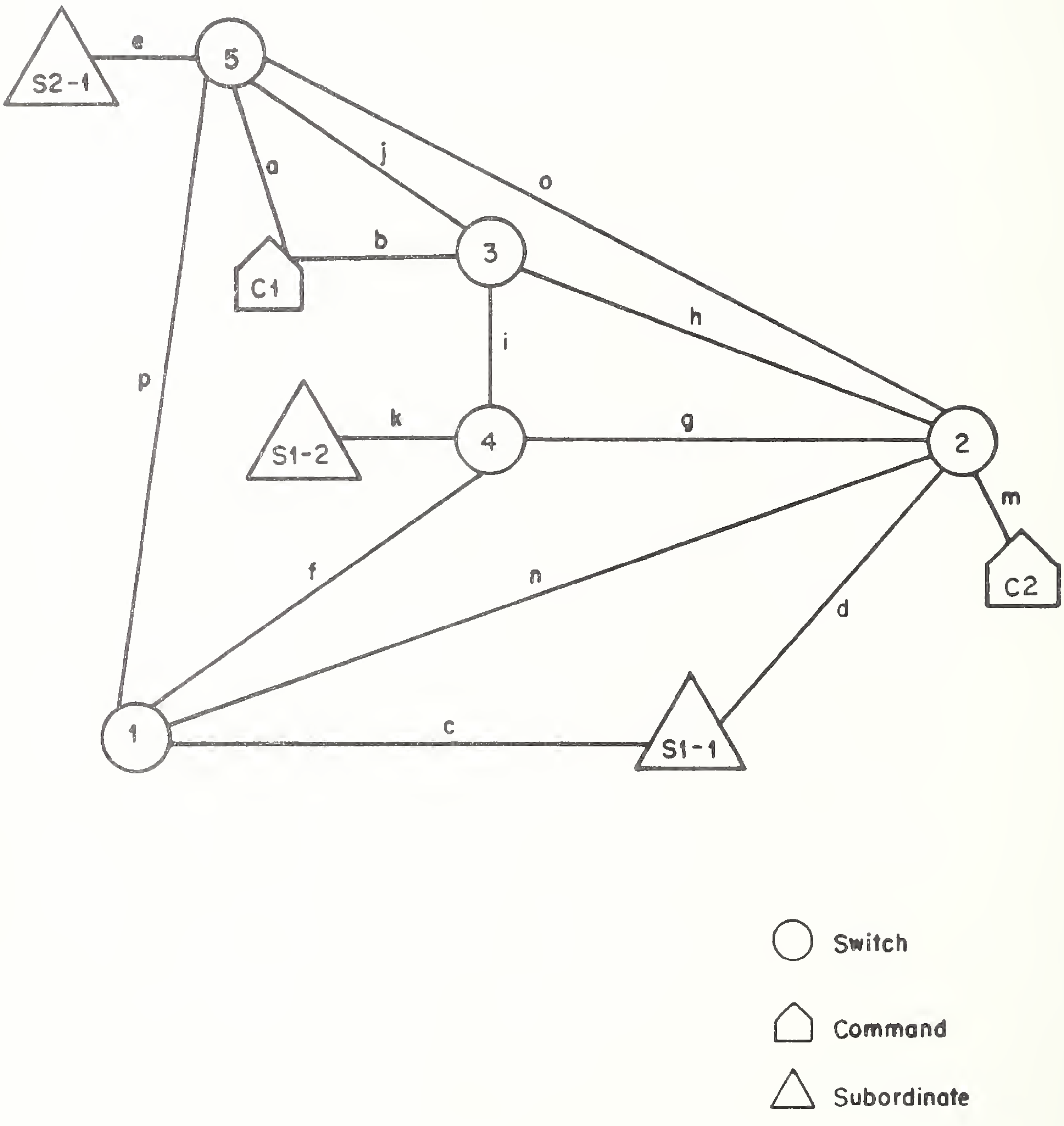


FIGURE 3. FIVE SWITCH NETWORK

The cut sets were used to evaluate the network elements in exact accord with the formula shown in Section 2.1. Each cut set was assigned the sum of the worths of the command-subordinate pairs which it cut, and each element of that cut set was assigned the value of the cut set divided by the number of elements in the cut set. The values assigned to each element were added together and divided by the number of cut sets in which that element appeared. This process gave the importance number as indicated by the formula.

Using the method described above, the following importance numbers were obtained for the elements of the three networks:

FOUR SWITCH NETWORK I

ELEMENT	SUM OF VALUES	NUMBER OF CUTS	v(e)
a	218.667	40	5.446
b	173.405	32	5.418
c	391.074	69	5.667
d	233.072	52	4.482
e	207.121	34	6.091
f	164.167	33	4.974
g	302.241	61	4.954
h	182.002	29	6.275
i	187.574	31	6.050
j	192.073	38	5.054
k	313.669	49	6.401
m	207.121	34	6.091
1	204.003	24	8.500
2	135.335	16	8.458
3	85.000	7	12.142
4	201.503	33	6.106

FOUR SWITCH NETWORK II

ELEMENT	SUM OF VALUES	NUMBER OF CUTS	v(e)
a	160.833	27	5.956
b	103.	15	6.866
c	193.835	28	6.922
d	336.670	62	5.430
e	173.334	25	6.933
f	273.834	47	5.826
g	222.167	38	5.846
h	238.334	39	6.111
i	173.500	34	5.102
j	205.	35	5.857
k	248.168	38	6.530
m	173.334	25	6.933
1	122.501	12	10.208
2	187.002	25	7.480
3	225.	22	10.227
4	163.501	18	9.083

FIVE SWITCH NETWORK

ELEMENT	SUM OF VALUES	NUMBER OF CUTS	v(e)
a	554.248	119	4.657
b	429.514	89	4.836
c	536.851	108	4.970
d	947.334	201	4.713
e	431.271	85	5.073
f	465.977	108	4.314
g	539.263	125	4.314
h	579.089	127	4.559
i	573.448	121	4.739
j	561.278	132	4.252
k	508.279	94	5.407
m	431.271	85	5.073
n	521.341	118	4.418
o	743.125	161	4.615
p	563.345	120	4.694
1	421.231	75	5.616
2	316.996	38	8.342
3	308.831	53	5.827
4	351.332	54	6.506
5	255.664	34	7.519

In all of the following lists, the element names given for the networks are those shown in figures 1, 2, and 3. The letters refer to links while the numbers are the names of switches.

The orders of network elements implied by the important numbers given above are as follows, given in descending order:

FOUR SWITCH NETWORK I

3, 1, 2, k, h, 4, e = m, i, c, a, b, j, f, g, d

FOUR SWITCH NETWORK II

3, 1, 4, 2, e = m, c, b, k, h, a, j, g, f, d, i

FIVE SWITCH NETWORK

2, 5, 4, 3, 1, k, e = m, c, b, i, d, p, a, o, h, n, f = g, j

3.2 PATHFINDER-WORTH EVALUATION

All three of these networks were tested using PATHFINDER-WORTH. Since the networks are quite small, we were able to use several different path weights with each of the networks and compare the results of each of these runs with the ordering for the same network using cut sets. The weights used were as follows:

RUN	1	2	3	4	5	6	7
PATH LENGTH							
1	1.0	1.0	1.0	.7	1.0	1.0	1.0
2	5.0	1.1	1.0	.6	.7	.4	.1
3	10.0	1.3	1.0	.4	.05	.05	.01

The orderings in ascending order of importance numbers for the three networks in these seven runs follows:

FOUR SWITCH NETWORK I

RUN	ORDER
1	c, f, j, i, 4, a, e = m = k, b, h, g, d, 2, 1, 3
2	c, f, j, i, 4, b, e = m = k, a, h, g, d, 2, 1, 3
3	c, f = j, i, 4, b, e = m = k, a, d = g = h, 1, 2, 3
4	j, f, c, i, 4, b, e = m = k, d, g, a, h, 1, 2, 3
5	j, f, i, 4, d, g, c, b, e = m = k, a, h, 1, 2, 3
6	j, i, f, 4, d, g, c, b, e = m = k, a, h, 1, 2, 3
7	j, i, f, 4, d, b, g, c, e = m = k, h, a, 2, 1, 3

FOUR SWITCH NETWORK II

RUN	ORDER
1	c, d, j, i, b, 2, k = e = m, a, h, f = g, 4, 3, 1
2	c, j, d, i, 2, b, k = e = m, a, h, f = g, 4, 3, 1
3	c, i, j, d, 2, b, k = e = m, a, f = g = h, 4, 3, 1
4	c, i, j, d, 2, b, k = e = m, f = g, a, h, 4, 3, 1
5	i, c, j, 2, d, b, f = g, k = e = m, h, a, 3, 4, 1
6	i, c, j, 2, d, b, f = g, k = e = m, h, a, 3, 4, 1
7	i, c, j, 2, d, b, f = g, k = e = m, h, a, 3, 4, 1

FIVE SWITCH NETWORK

RUN	ORDER
1	f, c, d, i, g, o, n, h = p, b, j, e = m = k, a, l, 4, 3, 2, 5
2	f, c, g, d, i, n, o, h = p, j, b, e = m = k, a, l, 4, 3, 2, 5
3	f, c, g, d, i, n, j, h = o = p, b, e = m = k, a, l, 4, 3, 2, 5
4	f, c, g, n, d, i, j, h = p, o, b, l, e = m = k, a, 4, 3, 2, 5
5	f, n, g, j, c, h = p, l, d, a, i, e = m = k, 4, b, o, 3, 2, 5
6	f, n, g, j, c, h = p, l, d, i, a, e = m = k, 4, b, o, 3, 2, 5
7	f, n, g, j, c, h = p, l, d, a, i, e = m = k, 4, b, o, 3, 2, 5

It is clear that those orderings are not identical to, and in some cases not even very close to, the orderings obtained using cut sets. However, since we are not sure the cut set orderings are optimal, it is reasonable for us to carry on with the numerical comparison of these results.

3.3 COMPARISON METHODS

There are three comparison methods which we have used to test the above results against each other and against a random ordering of the network elements.

The first method we used is the Rank Difference Method, found on page 394 of reference [4]. In this method, "given $\underline{n}$ corresponding pairs of measured items $(X_{\underline{i}}, Y_{\underline{i}})$, $(\underline{i} = 1, \ldots, \underline{n})$. Let $(\underline{u}_{\underline{i}}, \underline{v}_{\underline{i}})$ be the corresponding rank numbers. Here $\underline{u}_{\underline{i}} = 1$ for the largest $X_{\underline{i}}$, 2 for the next largest $X_{\underline{i}}$, etc., and similarly $\underline{v}_{\underline{i}} = 1$ for the largest $Y_{\underline{i}}$, 2 for the next largest $Y_{\underline{i}}$, etc.

$$\rho = 1 - \left(\frac{6 \sum_{\underline{i}} (u_{\underline{i}} - v_{\underline{i}})^2}{\underline{n}\,(\underline{n}^2 - 1)} \right)$$, (rank difference) coefficient

of correlation. In every case, $-1 \leq \rho \leq 1$. Check $\sum_{i} (\underline{u}_{\underline{i}} - \underline{v}_{\underline{i}}) = 0$."
Briefly, the larger the value of ρ, the closer the two lists are to being identical.

In the second method, we simply counted the smallest number P of interchanges of elements in one ordering necessary in order to bring it into the other ordering. In this method, we first arrange all elements with equal values into their correct relative ordering in one of the two lists L and L' with respect to the other list. We do this simply because no ordering of these elements is preferred (we must remember this fact when considering the quality of the ordering). Next, beginning with the lowest element in list L, we count the number of elements with which that element in list L' must be interchanged to bring it into the lowest position in list L'. This number goes into the value P. We delete this lowest element from list L' and perform the same operation with the next element of list L. This process will continue until all elements of list L' have been deleted. At this time, P has the desired value. Note that using this method, no interchange is counted for any element after it is deleted from list L'.

As an example, consider the two lists L and L' below:

L:	$\underline{a}, \underline{b}, \underline{c}, \underline{d}, \underline{e}$	(3)
L':	$\underline{b}, \underline{e}, \underline{c}, \underline{a}, \underline{d}$	

To bring list L' into the same order as L, we note that $\underline{a}$ in L' must be interchanged with three elements in order to bring it into its position in list L. Deleting $\underline{a}$ from L', we obtain

new L': $\underline{b}, \underline{e}, \underline{c}, \underline{d}$

Now we notice that no interchanges are needed to bring $\underline{b}$ into its correct position as the lowest in the remaining list L'. Therefore, P remains 3. Deleting $\underline{b}$ from L', we get

new L': $\underline{e}, \underline{c}, \underline{d}$.

Now c is the lowest among the remaining elements of L' in L. Therefore, we add 1 to P, getting 4, and delete $\underline{c}$ from L', getting

new L': $\underline{e}, \underline{d}$.

Only 1 interchanged more is needed to bring d and e into their correct relative positions. Therefore, for the L and L' shown in (3), P = 5.

The last method of comparing lists is simply a count of the number of different orderings implied by the equalities present in the ordering numbers generated by the evaluation techniques. We obtain this count in the following manner: For each set of equal valued elements, find the factorial of the number of equal valued elements. For one list, multiply together all factorials found. The larger the result, the more equalities present, and hence the less useful the ordering.

The following tables give the results of all three of these tests for each of the seven runs of PATHFINDER, and for a random ordering, compared against the order obtained using cut sets:

FOUR SWITCH NETWORK I

RUN	ρ	P	EN
1	.4353	35	6
2	.4471	37	6
3	.5088	36	72
4	.5530	32	6
5	.7260	24	6
6	.7382	26	6
7	.7206	25	6
RANDOM	.3235	48	0

FOUR SWITCH NETWORK II

RUN	ρ	P	EN
1	.4500	42	12
2	.4294	44	12
3	.4647	40	72
4	.4794	38	12
5	.5853	34	12
6	.5853	34	12
7	.5853	34	12
RANDOM	.2647	50	0

FIVE SWITCH NETWORK

RUN	ρ	P	EN
1	.7233	40	12
2	.7609	36	12
3	.7865	32	144
4	.7775	35	12
5	.7744	36	12
6	.7699	37	12
7	.7744	36	12
RANDOM	.1910	82	0

It will be noticed immediately that:

1) Both values ρ and P are much worse for the random ordering than for any of the PATHFINDER-WORTH runs.

2) None of the runs produced an ordering identical to that obtained using cut sets.

The ordering of the runs given by the two tests ρ and P are, with the best fit with the cut set ordering first:

FOUR SWITCH NETWORK I

ρ 6, 5, 7, 4, 3, 2, 1, RANDOM

P 5, 7, 6, 4, 1, 3, 2, RANDOM

FOUR SWITCH NETWORK II

ρ 5 = 6 = 7, 4, 3, 1, 2, RANDOM

P 5 = 6 = 7, 4, 3, 1, 2, RANDOM

FIVE SWITCH NETWORK

ρ 3, 4, 5 = 7, 6, 2, 1, RANDOM

P 3, 4, 2 = 5 = 7, 6, 1, RANDOM

Using the equality count test, we notice that run 3, where all path weights were equal, has a much higher equality value than any of the other runs for all three networks. This suggests strongly that we should exclude run 3 from further consideration.

Among the other runs, runs 4 through 7 give the best fits with the cut set ordering, although with varying orders among the runs. For both four switch networks, run 5, with path weights 1., .7, and .05 for path lengths 1, 2, and 3 switches, respectively, appears to be the best with respect to cut sets. In the five switch network, this weighting scheme is bettered only by run 4, with weights .7, .6, and .4. It would appear, then, that some reduction in value with increasing path length, so long as the reduction is not extreme, improves the ordering with respect to the cut set ordering.

In retrospect, this result is not surprising, since elements in large cut sets are penalized by having the worth of the cut set divided by the number of elements of the cut set. We would expect a long path to contain elements of more large cut sets than short paths, simply because long paths have more elements. But since the greatest increase in penalty to cut sets comes with the increase from one element to two elements (element values change from the whole worth of the cut set to ½ the worth of the cut set), and falls off more and more slowly after that, the reduction in weight for longer path lengths in PATHFINDER should be mild.

Comparing the orderings of the PATHFINDER-WORTH runs with a random ordering, we notice that even with an extremely bad weighting scheme such as that of run 1 (1, 5, 10 with increasing path length), PATHFINDER results are closer to cut set results than the random ordering is. This lends considerable weight to our belief that the PATHFINDER ordering does give an approximation of the ideal "true" ordering of the network elements. We conclude that elements found to be high in the PATHFINDER ordering have a good chance of being important, or worthy of further study, and those low in the ordering may well be quite unimportant. Notice, however, that there are elements, such as element 4 in FOUR SWITCH NETWORK I, which are low in the PATHFINDER-WORTH ordering and high in the cut set ordering, or *vice versa*. Such elements serve as a warning that we *cannot accept the PATHFINDER-WORTH ordering of the network elements as the final arbiter of network element importance*; the ordering can only serve as a guide.

4.0 VALIDITY OF PASSING FROM SMALL NETWORKS TO LARGE ONES

As mentioned previously, we were forced to carry out the present study using very small networks simply because of the large number of cut sets even in a five switch network (424 of all kinds). But our real concern is with networks containing 40 to 75 switches and a proportionate number of links. How valid are our conclusions for such networks?

Of course, we cannot give a definitive answer to this question without actually carrying out the study for such networks, a presently impossible task. However, we can note the effects of going from the four switch networks to the five switch network and extrapolate from those effects. First, the largest value of ρ for the five switch network is noticably larger (the fit for that run is noticably better) than the largest values for either of the two four switch networks, and the smallest value of ρ (for the random ordering) is noticably smaller than the smallest value for either of the two four switch networks. While the smallest value for P is found on one of the four switch networks, this is caused partly by the fact that there are fewer elements to permute on those two networks. The largest value of P by far is that for the random ordering on the five switch network.

Although the number of elements in the five switch network is 20 as compared with 16 in each of the four switch networks, the number of equalities generated in the five switch network orderings is the <u>same</u> as the number in one of the four switch networks and only one more than in the other network. On so small a sample it is dangerous to draw any conclusions, but it seems possible that the equality number may actually decrease or at least stay the same with increasing network size. Such a result would seem reasonable since, with increasing complexity, the chances of different numbers of different length paths passing through different links ought to increase.

For the above reasons we can resonably hope that whatever validity PATHFINDER-WORTH orderings have in small networks will not be destroyed by increasing the size of the network, other things being equal.

But what other things might not be equal? First the character (density and other properties) of the larger network is almost certainly different from that of the smaller networks. This situation is unavoidable; however, we selected our examples to have a density (number of links divided by total possible number of links, without parallel links) as close to that of the larger networks as possible. In addition, we attempted to use proportionate numbers of commands, subordinates, and connections of those elements to the switches. Finally, we scaled down the maximum path length from 6 to 8 switches to 3 switches (the smallest number which would give values for every link and switch of the networks).

A second problem is that the path weighting scheme in the larger network cannot easily be inferred from the path weighting schemes in the small networks. Fortunately, we need only extend the weighting scheme from a maximum of 3 switches to 6 or 8 switches, not to 40 or 80, since the cut sets and paths found and used both depend on the maximum path length chosen. In addition, we notice one additional property of the values of ρ and P in Chart II - the values of these measures are quite close over wide ranges of choices of path weighting schemes. Thus the orderings are rather insensitive to path weight variations and we need merely specify a declining weight with increasing path length, rather than a very specific set of values for those path weights. Such a simple specification should yield a usable ordering of the network elements.

5.0 CONCLUSIONS

At the very best, this study could add weight to the belief that the PATHFINDER-WORTH values give an ordering of the network elements which guides us to important elements (ones high on the list) and unimportant elements (ones low on the list). In our opinion, the study is successful in this respect. Our conclusions are as follows:

1) A moderate amount of reduction in weight for increasing path length with the PATHFINDER program appears to improve the resulting ordering of the network elements when compared to the ordering obtained using cut sets.

2) Network elements found to be high in the PATHFINDER-WORTH ordering have a good chance of being important, or worthy of receiving further analysis and possibly even a larger proportion of our resources, while those low in the ordering may well be quite unimportant and hence candidates for deletion from the network.

NOTE: As demonstrated in Section 3.3, however, we cannot accept the PATHFINDER-WORTH ordering of the network elements as the final arbiter of network element importance; the ordering can only serve as a guide for further analysis.

3) For reasons which include insensitivity to variations in path length weights for PATHFINDER-WORTH values, we believe the results described above will still hold true for large networks (containing 40 to 80 switches).

6.0 CUT SET LISTS

We found the cut sets listed in Sections 6.1, 6.2, and 6.3 by a method involving the following fact: We know that any cut set which separates more than one command-subordinate pair must contain, for each of the pairs cut, a cut set which separates that pair and which cannot be reduced and still cut that pair. Therefore, for each network we proceeded as follows: We found a list of all cut sets for each command-subordinate pair (even in the five switch network there were only 25 at most for a single pair). We then made all combinations of one cut set from each of two such lists and one from each of the three lists. The resulting collections of elements were classified into classification groups by the command-subordinate pairs they were known to separate (because of the lists from which their elements were chosen). Within each classification group, all sets of elements which properly contained another set of elements in the same group were deleted, and all but one copy of each remaining set was deleted. Finally, all cut sets which appeared in the group classified as cutting all command-subordinate pairs were deleted wherever they appeared in groups classified as cutting just two pairs or just one command-subordinate pair, and all cut sets which appeared in the groups classified as cutting two command-subordinate pairs were deleted from the groups classified as cutting only one command-subordinate pair. The resulting lists are those which are given in Sections 6.1, 6.2, and 6.3.

The process described in the previous paragraph could be readily programmed for the computer, and in fact the portion which involved combining cut sets was programmed. If the whole process were computerized, it could be applied to large systems; however, the running time for such applications would be extremely large (many hours for even moderately large networks).

FOUR SWITCH NETWORK I

CUT SETS WHICH CUT ALL PAIRS

a, b, e
a, b, m
a, d, h
c, e, i, k
c, f, g, h
c, i, k, m
c, d, h, i, k
c, e, f, j, k
c, f, j, k, m
c, g, h, i, j
a, d, e, g, j, k
a, d, g, j, k, m
b, c, d, e, f, g
b, c, d, f, g, m
c, d, f, h, j, k
b, c, d, e, g, i, j
b, c, d, g, i, j, m

1, h
1, b, m
1, b, e
1, j, k, m
1, e, j, k
1, i, k, m
1, e, i, k
2, a
2, c, f, g
2, c, i, k
2, c, f, j, k
2, c, g, i, j
3, a, b
3, a, d
3, c, f
3, c, i
4, c, k, m
4, b, c, d, g, m
4, c, e, k
4, c, g, h
4, a, b, g, h
4, c, d, h, k
4, b, c, d, e, g
4, a, d, e, g, k
4, a, d, g, k, m

1, 2
1, 3
1, 4, k, m
1, 4, e, k
2, 4, c, g
2, 4, c, k
3, 4, c

CUT SETS WHICH CUT OFF COMMAND 1 BUT LEAVE COMMAND 2 CONNECTED

a, b	1, b	1, 4, k
c, i, k	1, i, k	
c, f, j, k	1, j, k	
b, c, d, f, g	4, c, k	
a, d, g, j, k	4, a, d, g, k	
b, c, d, g, i, j	4, b, c, d, g	

CUT SETS WHICH CUT OFF COMMAND 2 AND SUBORDINATE 1 OF COMMAND 1

c, e, i	1, e, j	1, 4, m
c, i, m	1, j, m	1, 4, e
c, e, f, j	1, i, m	2, 4, c
c, f, j, m	1, e, i	
c, d, h, i	2, c, i	
a, d, g, j, m	2, c, f, j	
a, d, e, g, j	4, c, e	
c, d, f, h, j	4, c, m	
	4, a, d, g, m	
	4, a, d, e, g	

CUT SETS WHICH CUT OFF COMMAND 2 AND SUBORDINATE 2 OF COMMAND 1

e, k	3	2, 4, g
k, m	2, k	
d, h, k	2, g, f	
f, g, h	2, g, j	
g, h, j	4, g, h	
b, d, e, f, g	4, b, d, e, g	
b, d, e, g, j	4, b, d, g, m	
b, d, f, g, m	4, c, d, h	
b, d, g, j, m		

CUT SETS WHICH CUT COMMAND 1 FROM SUBORDINATE 1

c, i	1, i	1, 4
c, f, j	1, j	
a, d, g, j	4, c	
a, d, f, g, i	4, a, d, g	

CUT SETS WHICH CUT COMMAND 1 FROM SUBORDINATE 2

k	4, b, d, g
b, d, f, g	
b, d, g, j	

CUT SETS WHICH CUT COMMAND 2 FROM HIS SUBORDINATE

m	2
e	
d, h	
g, h	

FOUR SWITCH NETWORK II

CUT SETS WHICH CUT ALL PAIRS

a, b, e
a, b, m
a, g, h
a, b, f, h
b, f, h, j
c, d, e, k
c, d, k, m
e, f, g, j
f, g, h, j
f, g, j, m
a, c, e, f, g

a, c, f, g, m
b, c, f, h, i
c, d, f, h, k
c, d, g, h, k
c, e, f, g, i
c, f, g, h, i
c, f, g, i, m
d, e, i, j, k
d, i, j, k, m
a, d, e, g, j, k
a, d, e, h, i, k
a, d, f, h, i, k
a, d, g, j, k, m
a, d, h, i, k, m
b, d, e, f, j, k
b, d, f, j, k, m
d, f, h, i, j, k
d, g, h, i, j, k

1, b
1, g
1, c, d, k
1, d, i, k
2, b, f, h
2, d, e, k
2, d, k, m
2, e, f, g
2, f, g, h
2, f, g, m
2, d, f, h, k
2, d, g, h, k
3, a, h
3, c, e
3, c, h
3, c, m
3, e, j
3, h, j
3, j, m
4, a
4, f, j
4, c, d, k
4, c, f, i
4, d, i, j, k

1, 2, d, k
1, 3
1, 4
2, 3, e
2, 3, h
2, 3, m
2, 4, f
2, 4, d, k
3, 4, c
3, 4, j

2, 3, 4

CUT SETS WHICH CUT OFF COMMAND 1 BUT LEAVE COMMAND 2 CONNECTED

a, b
c, d, k
f, g, j
a, c, f, g
c, f, g, i
d, i, j, k
a, d, g, j, k
a, d, h, i, k
b, d, f, j, k

2, d, k
2, f, g
3, c
3, j

2, 3

CUT SETS WHICH CUT OFF COMMAND 2 AND SUBORDINATE 1 OF COMMAND 1

c, d, e
c, d, m
c, d, f, h
c, d, g, h
d, e, i, j
d, i, j, m
a, d, e, g, j
a, d, e, h, i
a, d, f, h, i
a, d, g, j, m
a, d, h, i, m
b, d, e, f, j
b, d, f, j, m
d, f, h, i, j
d, g, h, i, j

1, c, d
1, d, i
2, d, e
2, d, m
2, d, f, h
2, d, g, h
4, c, d
4, d, i, j

1, 2, d
2, 4, d

CUT SETS WHICH CUT OFF COMMAND 2 AND SUBORDINATE 2 OF COMMAND 1

e, k
k, m
f, h, k
g, h, k
a, e, f, g
a, f, g, m
b, f, h, i
e, f, g, i
f, g, h, i
f, g, i, m

1, k
3, e
3, h
3, m
4, k
4, f, i

3, 4

CUT SETS WHICH CUT COMMAND 1 FROM SUBORDINATE 1

c, d	2, d
d, i, j	
a, d, g, j	
a, d, h, i	
b, d, f, j	

CUT SETS WHICH CUT COMMAND 1 FROM SUBORDINATE 2

k	3
a, f, g	
f, g, i	

CUT SETS WHICH CUT COMMAND 2 FROM HIS SUBORDINATE

e	1
m	4
f, h	
g, h	

6.3 FIVE SWITCH NETWORK

CUT SETS WHICH CUT ALL PAIRS

a, b, e
a, b, m
b, j, o, p
c, d, e, k
c, d, k, m
h, i, o, p
a, b, h, n, o
a, b, h, o, p
a, b, j, n, o
a, e, h, i, j
a, h, i, j, m
b, h, k, o, p
c, d, h, k, n, o
c, d, h, k, o, p
c, d, j, k, n, o
c, d, j, k, o, p
c, d, e, g, i, p
c, d, g, i, m, p
c, d, e, f, g, i
c, d, f, g, i, m
a, c, d, e, h, i
a, c, d, h, i, m
a, c, d, e, g, i
a, c, d, g, i, m
d, e, i, k, n, p
d, i, k, m, n, p
d, e, g, i, n, p
d, g, i, m, n, p
d, e, f, k, n, p
d, f, k, m, n, p
a, h, i, j, n, o
b, d, e, k, o, p
b, d, k, m, o, p
b, d, e, i, o, p
b, d, i, m, o, p
b, d, e, k, n, p
b, d, k, m, n, p
a, d, e, h, i, p
a, d, h, i, m, p
a, c, e, h, i, o
a, c, h, i, m, o
a, c, h, i, n, o
c, g, h, k, o, p
d, f, h, k, o, p

b, c, h, k, n, o
c, h, i, k, n, o
c, f, h, i, n, o
c, g, h, k, n, o
b, c, j, k, n, o
b, c, f, j, n, o
f, g, h, k, o, p
a, c, e, g, h, k, o
a, c, g, h, k, m, o
a, d, e, f, h, j, k
a, d, f, h, j, k, m
a, d, e, f, h, k, p
a, d, f, h, k, m, p
a, d, e, i, j, k, n
a, d, i, j, k, m, n
a, d, i, j, k, n, o
c, d, g, i, j, o, p
b, c, d, e, g, j, p
b, c, d, g, j, m, p
b, c, d, e, f, j, o
b, c, d, f, j, m, o
b, c, d, e, f, g, j
b, c, d, f, g, j, m
b, c, d, e, f, i, o
b, c, d, f, i, m, o
c, d, e, f, h, i, o
c, d, f, h, i, m, o
d, i, j, k, n, o, p
d, g, i, j, n, o, p
d, f, j, k, n, o, p
b, d, e, g, j, n, p
b, d, g, j, m, n, p
a, e, f, g, h, j, k
a, f, g, h, j, k, m
a, d, e, g, i, j, n
a, d, g, i, j, m, n
a, d, g, i, j, n, o
a, d, e, f, j, k, n
a, d, f, j, k, m, n
a, d, f, j, k, n, o
a, c, e, g, h, j, k
a, c, g, h, j, k, m
c, d, f, g, i, j, n, o
a, f, g, h, j, k, n, o

CUT SETS WITH SWITCHES WHICH CUT ALL PAIRS

1, d, e, k
1, d, k, m
1, h, i, o
1, b, j, o
1, d, h, k, o
1, d, j, k, o
1, d, e, g, i
1, d, g, i, m
1, a, b, h, o
1, g, h, k, o
1, b, h, k, o
1, d, g, i, j, o
1, a, d, e, h, i
1, a, d, h, i, m
1, b, d, e, g, j
1, b, d, g, j, m
1, b, d, e, i, o
1, b, d, i, m, o
1, a, e, g, h, j, k
1, a, g, h, j, k, m
2, c, k
2, a, b
2, i, p
2, a, c, i
2, c, f, i
2, b, k, p
2, b, j, p
2, f, k, p
2, a, i, j
2, b, c, f, j
2, a, f, j, k
3, a, e
3, a, m
3, o, p
3, a, n, o
3, c, k, n, o
3, c, f, n, o
3, c, d, e, f, g
3, c, d, f, g, m
3, c, d, e, f, o
3, c, d, f, m, o
3, c, d, e, g, p
3, c, d, g, m, p
3, d, e, k, n, p
3, d, k, m, n, p
3, d, e, g, n, p
3, d, g, m, n, p
4, c, d, e
4, c, d, m
4, h, o, p
4, d, e, n, p
4, d, m, n, p
4, a, e, h, j
4, a, h, j, m
4, c, h, n, o
4, c, d, j, n, o
4, c, d, j, o, p
4, d, j, n, o, p
4, a, h, j, n, o
4, a, d, e, h, p
4, a, d, h, m, p
4, a, c, e, h, o
4, a, c, h, m, o
4, a, d, e, j, n
4, a, d, j, m, n
4, a, d, j, n, o
4, b, d, e, o, p
4, b, d, m, o, p
4, b, c, j, n, o
5, b
5, h, i
5, c, d, k
5, c, d, g, i
5, c, g, h, k
5, d, f, h, k
5, d, f, k, n
5, d, i, k, n
5, d, g, i, n
5, f, g, h, k

CUT SETS WITH SWITCHES WHICH CUT ALL PAIRS

1, 2, i
1, 2, k
1, 2, b, j
1, 3, o
1, 3, d, g, m
1, 3, d, e, g
1, 4, d, e
1, 4, d, m
1, 4, h, o
1, 4, d, j, o
1, 5, d, k
1, 5, d, g, i
1, 5, g, h, k
2, 3, a
2, 3, p
2, 3, c, f
2, 4, c
2, 4, p
2, 4, a, j
2, 5, i
2, 5, f, k
3, 4, c, n, o
3, 5
4, 5, h
4, 5, d, n
4, 5, c, d

1, 2, 3
1, 2, 4
1, 4, 5, d
2, 4, 5

CUT SETS WHICH CUT COMMAND 1 AND LEAVE COMMAND 2 CONNECTED

a, b,
c, d, k
a, h, i, j
c, d, g, i, p
c, d, f, g, i
a, c, d, h, i
a, c, d, g, i
d, i, k, n, p
d, g, i, n, p
d, f, k, n, p
b, d, k, o, p
b, d, i, o, p
b, d, k, n, p
a, d, h, i, p
a, c, h, i, o
b, c, d, g, j, p
b, c, d, f, j, o
b, c, d, f, g, j
b, c, d, f, i, o
c, d, f, h, i, o
b, d, g, j, n, p
a, f, g, h, j, k
a, c, g, h, j, k
a, c, g, h, k, o
a, d, f, h, j, k
a, d, f, h, k, p
a, d, i, j, k, n
a, d, g, i, j, n
a, d, f, j, k, n
b, c, h, k, n, o

1, d, k
1, d, g, i
1, a, d, h, i
1, b, d, i, o
1, a, g, h, j, k
3, a
3, c, d, f, g
3, c, d, f, o
3, c, d, g, p
3, d, k, n, p
3, d, g, n, p
4, c, d
4, d, n, p
4, a, h, j
4, a, d, h, p
4, a, c, h, o
4, a, d, j, n
4, b, d, o, p
1, b, d, g, j

1, 3, d, g
1, 4, d

CUT SETS WHICH CUT COMMAND 2 AND SUBORDINATE 1 OF COMMAND 1

c, d, e
c, d, m
b, h, o, p
c, d, h, n, o
c, d, h, o, p
c, d, j, n, o
c, d, j, o, p
d, e, i, n, p
d, i, m, n, p
d, e, f, n, p
d, f, m, n, p
b, d, e, o, p
b, d, m, o, p
b, d, e, n, p
b, d, m, n, p
c, g, h, o, p
d, f, h, o, p
b, c, h, n, o
c, h, i, n, o
c, g, h, n, o
b, c, j, n, o
f, g, h, o, p
d, i, j, n, o, p
d, f, j, n, o, p
a, e, f, g, h, j
a, f, g, h, j, m
a, c, e, g, h, j
a, c, g, h, j, m
a, c, e, g, h, o
a, c, g, h, m, o
a, d, e, f, h, j
a, d, f, h, j, m
a, d, e, f, h, p
a, d, f, h, m, p
a, d, e, i, j, n
a, d, i, j, m, n
a, d, i, j, n, o
a, d, e, f, j, n
a, d, f, j, m, n
a, d, f, j, n, o
a, f, g, h, j, n, o

1, d, e
1, d, m
1, d, h, o
1, d, j, o
1, g, h, o
1, b, h, o
1, a, e, g, h, j
1, a, g, h, j, m
2, c
2, b, p
2, f, p
2, a, f, j
3, c, n, o
3, d, e, n, p
3, d, m, n, p
5, c, d
5, c, g, h
5, d, f, h
5, d, f, n
5, d, i, n
5, f, g, h

1, 2
1, 5, d
1, 5, g, h
2, 5, f

CUT SETS WHICH CUT COMMAND 2 AND SUBORDINATE 2 OF COMMAND 1

e, k
k, m
h, k, n, o
h, k, o, p
j, k, n, o
j, k, o, p
e, g, i, p
g, i, m, p
e, f, g, i
f, g, i, m
a, e, h, i
a, h, i, m
a, e, g, i
a, g, i, m
g, i, j, o, p
a, h, i, n, o
b, e, g, j, p
b, g, j, m, p
b, e, i, o, p
b, i, m, o, p
b, e, f, j, o
b, f, j, m, o
b, f, j, n, o
b, e, f, g, j
b, f, g, j, m
b, e, f, i, o
b, f, i, m, o
e, f, h, i, o
f, h, i, m, o
f, h, i, n, o
f, g, i, j, n, o
a, g, i, j, n, o

1, h, k, o
1, j, k, o
1, e, g, i
1, g, i, m
1, g, i, j, o
1, b, e, g, j
1, b, g, j, m
1, b, e, i, o
1, b, i, m, o
2, k
2, a, i
2, f, i
2, b, f, j
3, k, n, o
3, e, f, g
3, f, g, m
3, e, f, o
3, f, m, o
3, f, n, o
3, e, g, p
3, g, m, p
4, e
4, m
4, h, n, o
4, j, o, p
5, k
5, g, i
4, j, n, o

1, 3, g, m
1, 3, e, g
1, 4, j, o
2, 3, f
2, 4
3, 4, n, o
4, 5

CUT SETS WHICH CUT ONLY SUBORDINATE 1 OF COMMAND 1

c, d	1, d
d, i, n, p	1, a, g, h, j
d, f, n, p	3, d, n, p
b, d, o, p	
b, d, n, p	
a, f, g, h, j	
a, c, g, h, j	
a, c, g, h, o	
a, d, f, h, j	
a, d, f, h, p	
a, d, i, j, n	
a, d, f, j, n	

CUT SETS WHICH CUT ONLY SUBORDINATE 2 OF COMMAND 1

k	1, g, i	1, 3, g
g, i, p	1, b, g, j	
f, g, i	1, b, i, o	
a, h, i	3, f, g	
a, g, i	3, f, o	
b, g, j, p	3, g, p	
b, i, o, p	4	
b, f, j, o		
b, f, g, j		
b, f, i, o		
f, h, i, o		

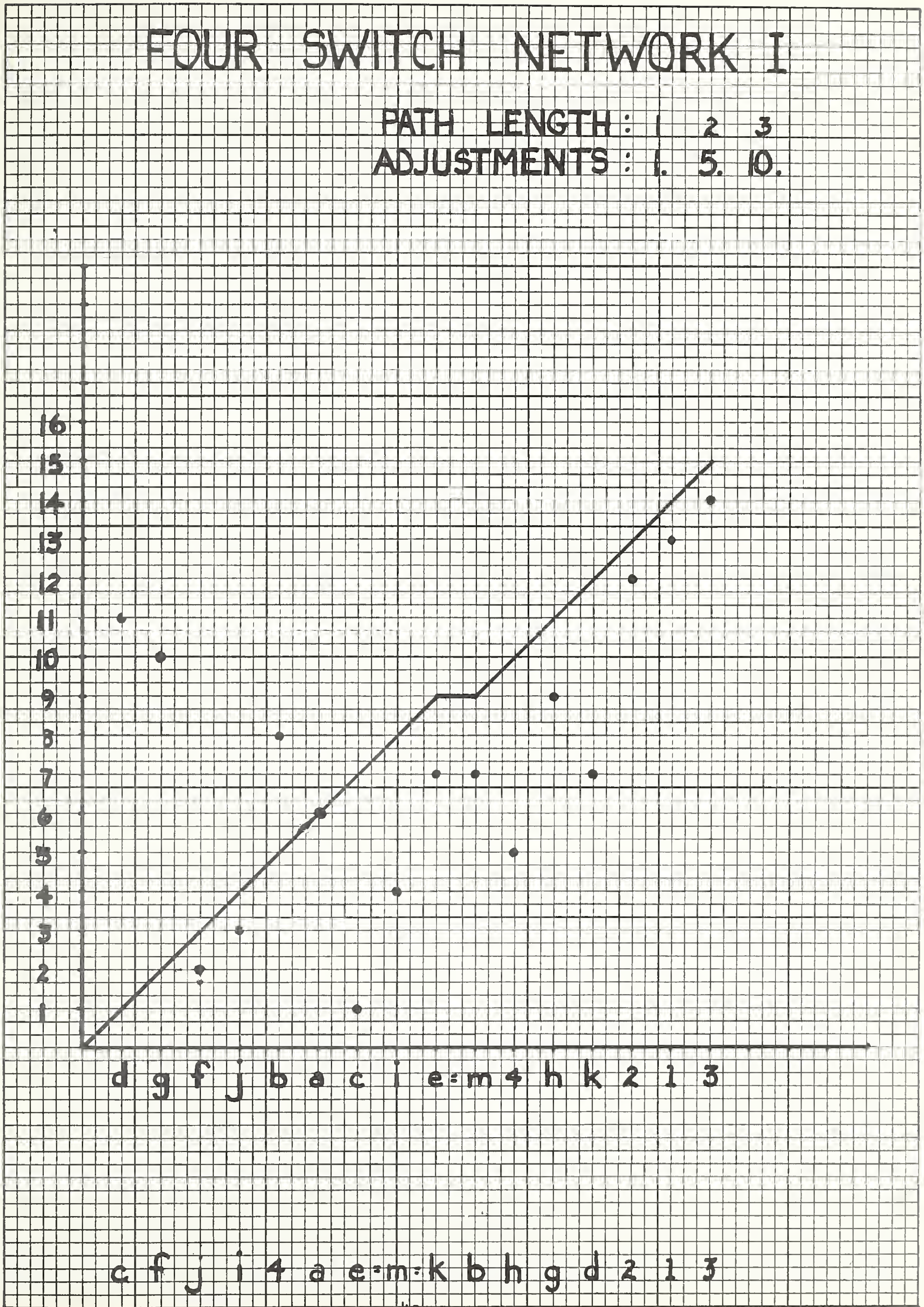
CUT SETS WHICH CUT ONLY THE SUBORDINATE OF COMMAND 2

e	1, h, o
m	1, j, o
h, n, o	2
h, o, p	3, n, o
j, n, o	5
j, o, p	

6.4 ORDERING COMPARISONS

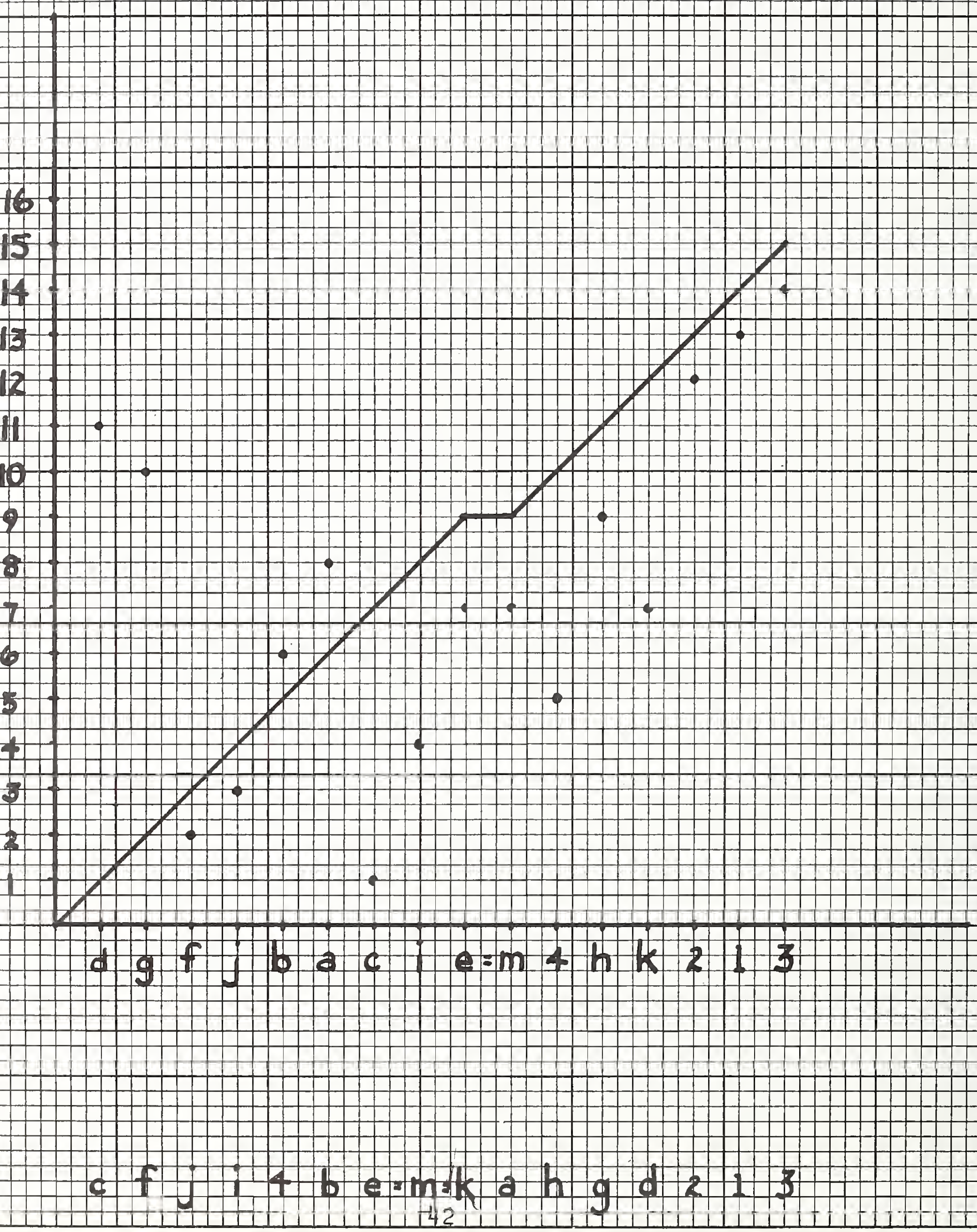
For each of the networks and runs, the following graphs give the comparison between the orderings obtained using the cut set method and using PATHFINDER-WORTH. Rank positions are listed on the vertical axis of each graph. Along the horizontal axis is the cut set ordering found for the given network, and a diagonal line is drawn through the chart based on the ordering given. Note that if two network elements were found to have equal values using either the cut set method or PATHFINDER-WORTH, they are given identical rank orderings, there being no way to distinguish between the two elements.

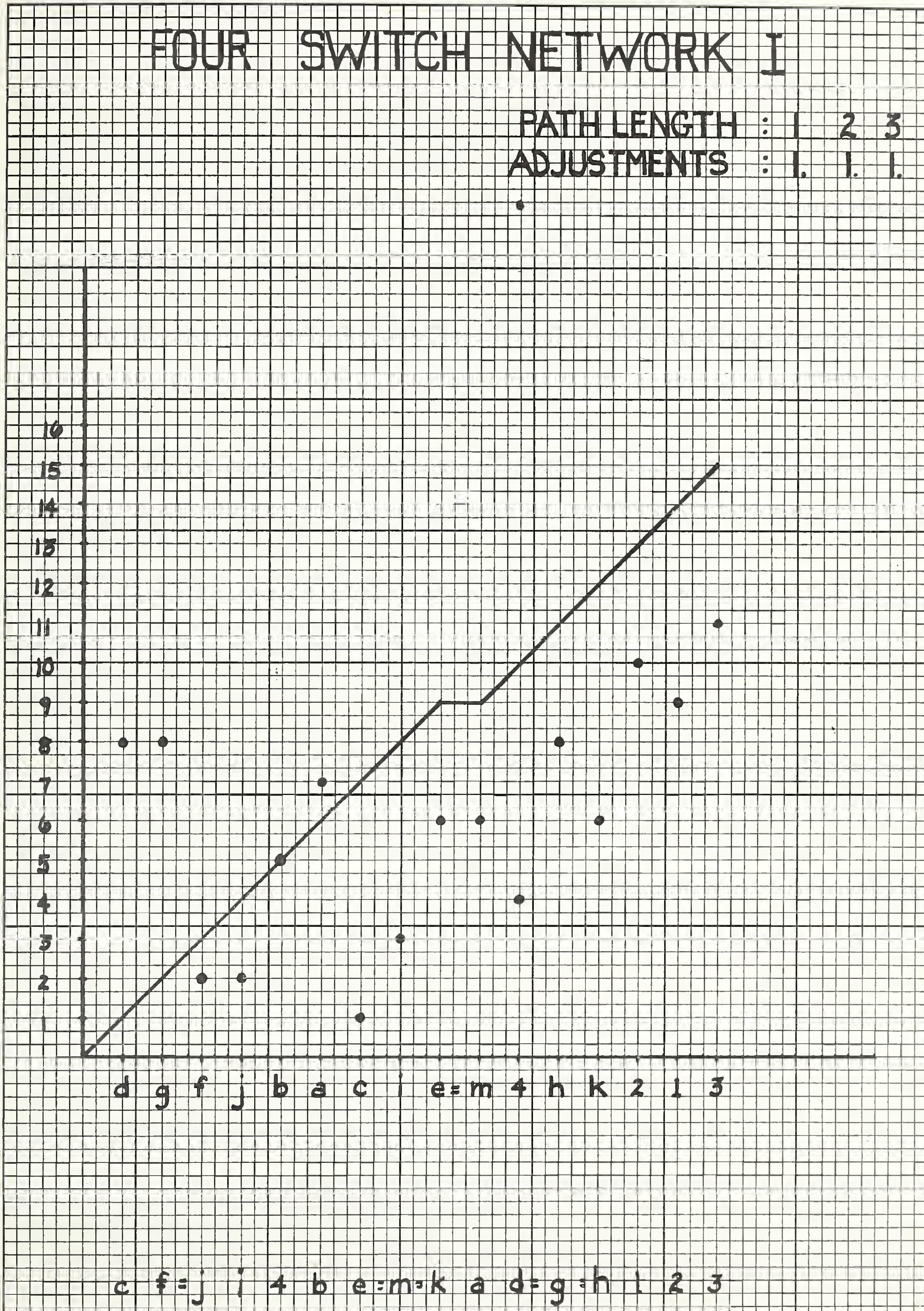
Below the cut set ordering list is the list of the network elements in the order given by PATHFINDER-WORTH, using the adjustments shown in the upper right-hand corner of the figure. This second list is numbered from left to right to give the rank ordering, and, for each element in this list, its proper location on the graph is plotted with a large dot. Finally, the values of P and ρ for the graph are given in the lower right-hand corner of each graph.

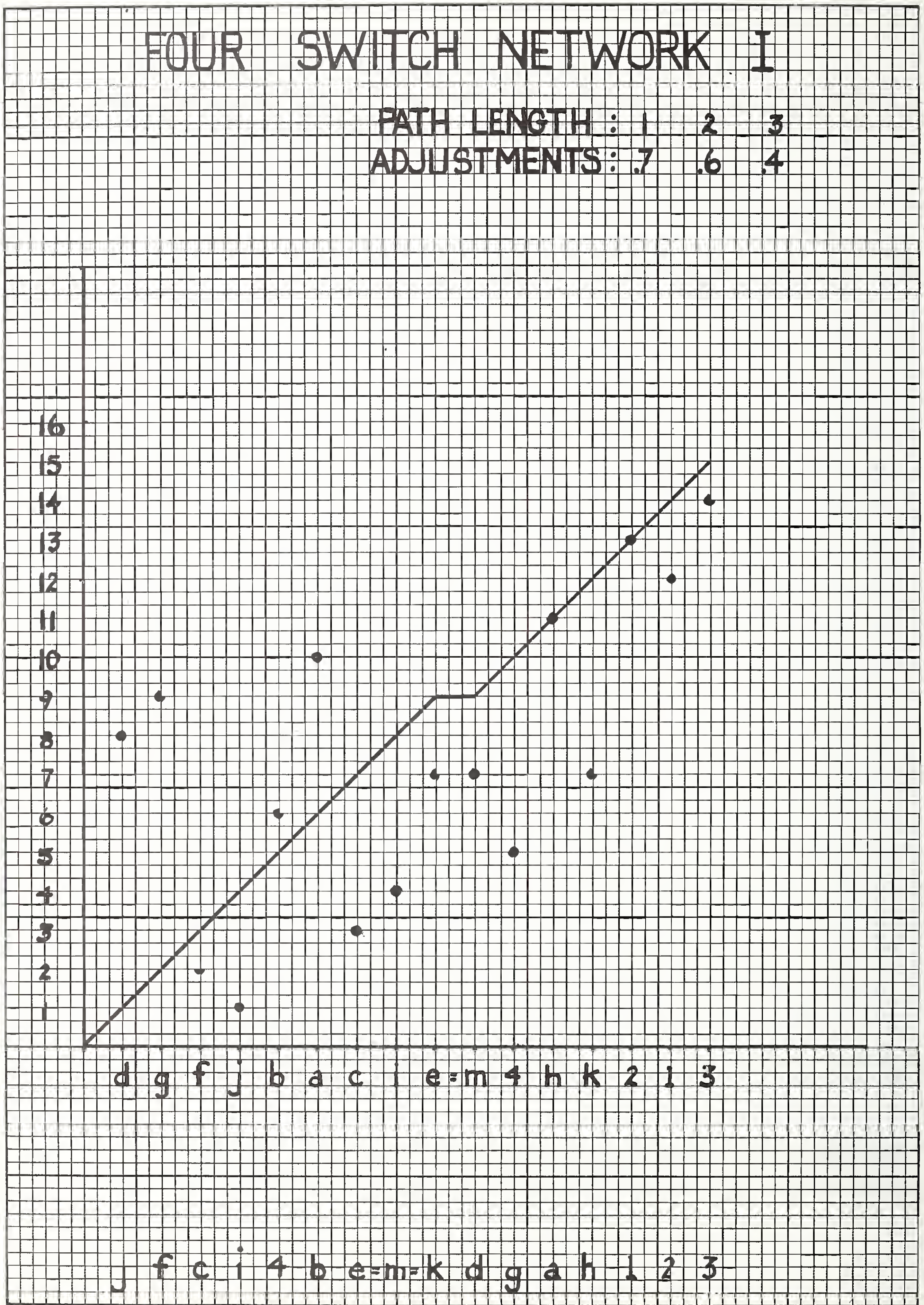

FOUR SWITCH NETWORK I
PATH LENGTH : 1 2 3
ADJUSTMENTS : 1. 5. 10.
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
d g f j b a c i e=m 4 h k 2 1 3
c f j i 4 a e=m=k b h g d 2 1 3

FOUR SWITCH NETWORK I

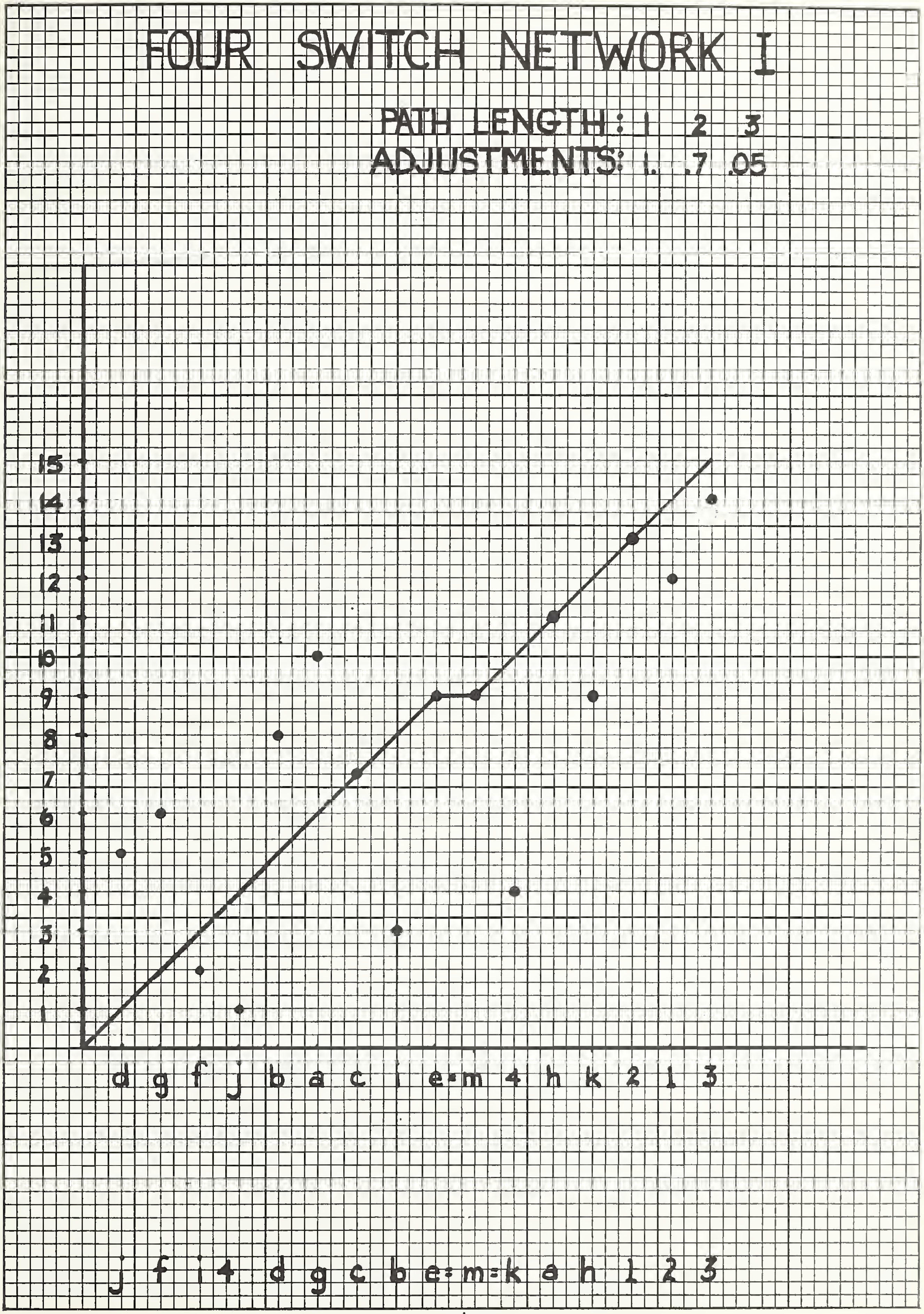
PATH LENGTH :	1	2	3
ADJUSTMENTS :	1.	1.1	1.3




FOUR SWITCH NETWORK I
PATH LENGTH : 1 2 3
ADJUSTMENTS : 1. 1. 1.
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
d g f j b a c i e=m 4 h k 2 1 3
c f=j i 4 b e=m=k a d=g=h 1 2 3

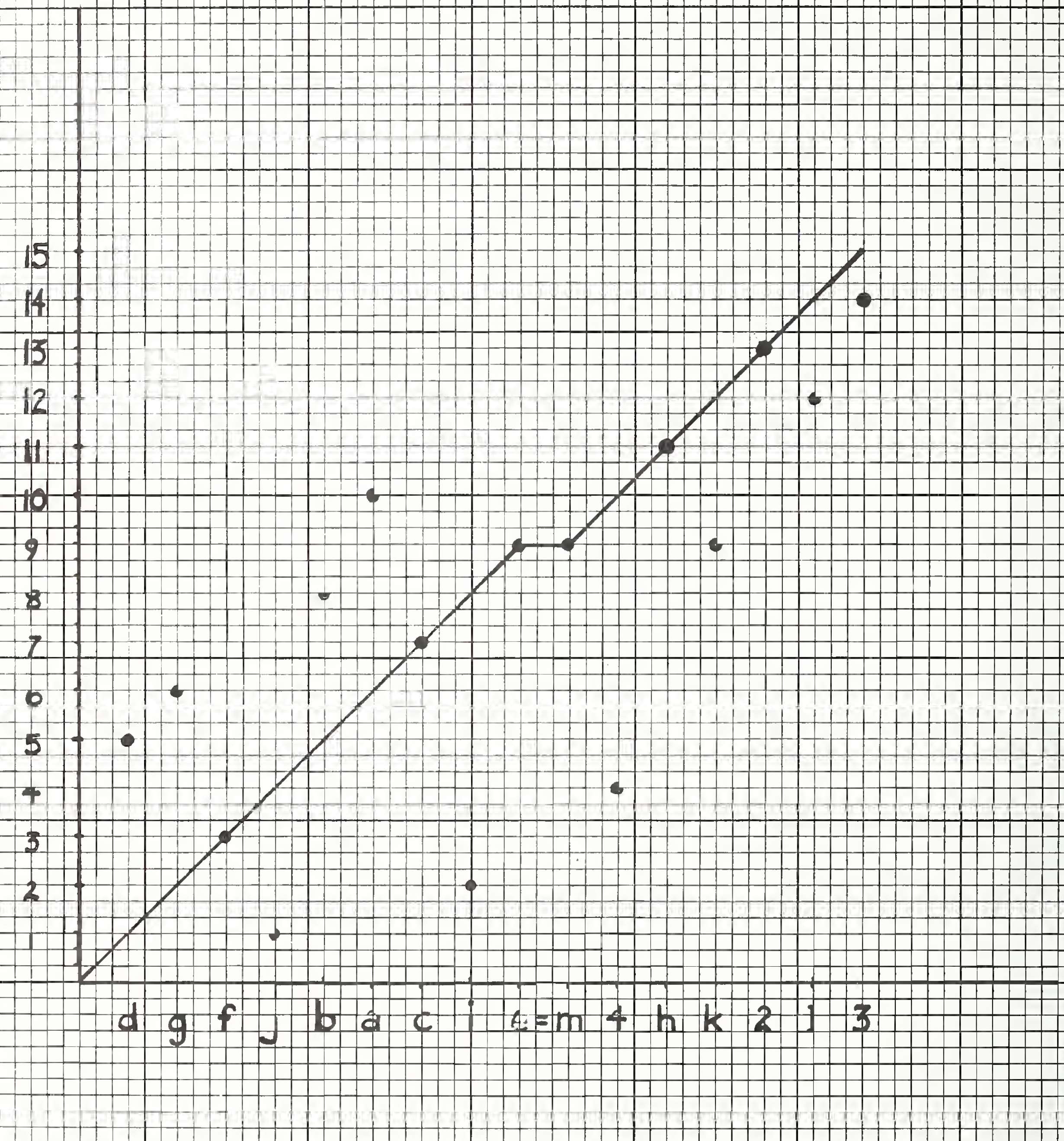


FOUR SWITCH NETWORK I
PATH LENGTH : 1 2 3
ADJUSTMENTS : .7 .6 .4
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
d g f j b a c i e=m 4 h K 2 1 3
j f c i 4 b e=m=K d g a h 1 2 3


FOUR SWITCH NETWORK I
PATH LENGTH: 1 2 3
ADJUSTMENTS: 1. .7 .05
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
d g f j b a c i e=m 4 h k 2 1 3
j f i 4 d g c b e=m=k a h 1 2 3

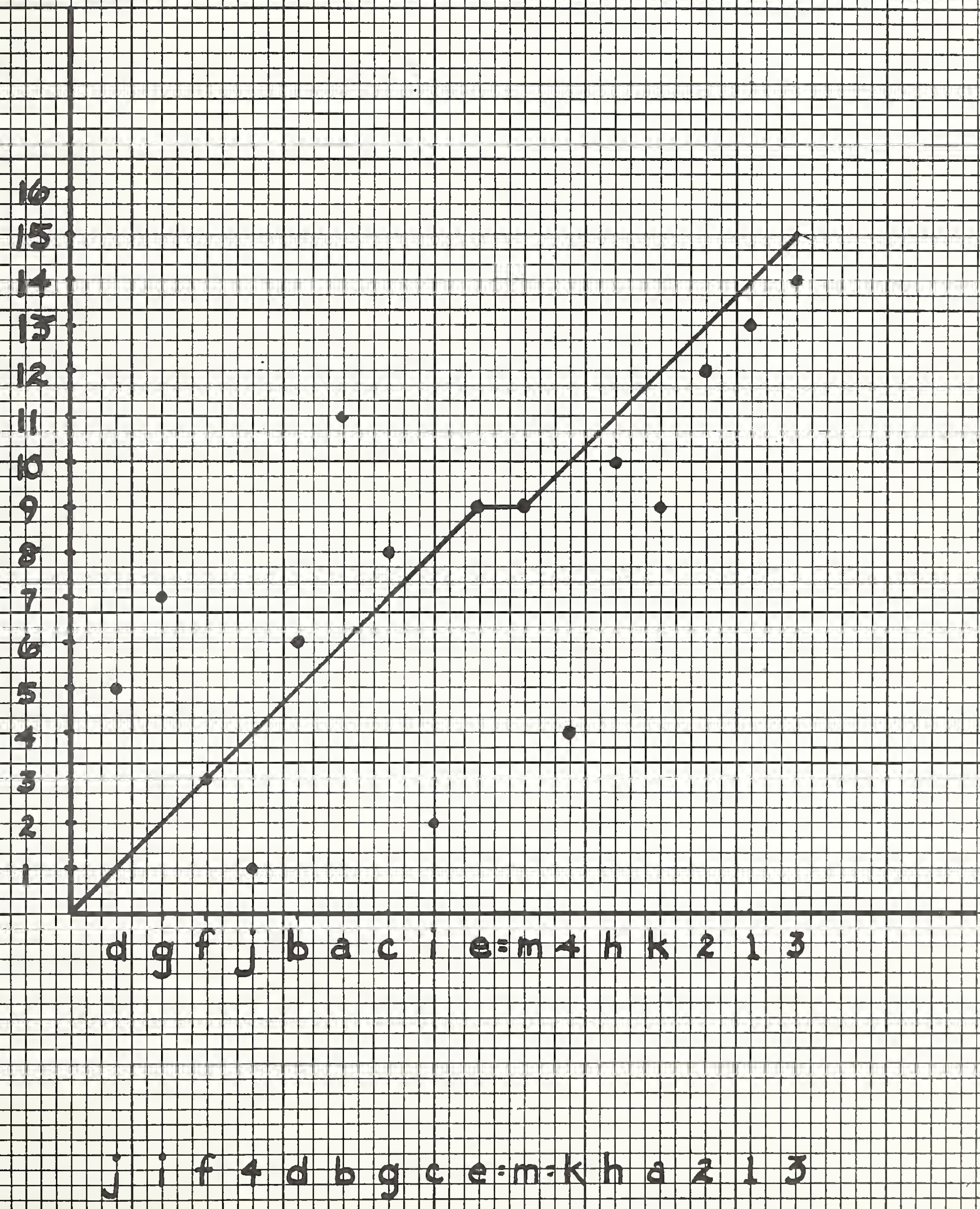
FOUR SWITCH NETWORK I

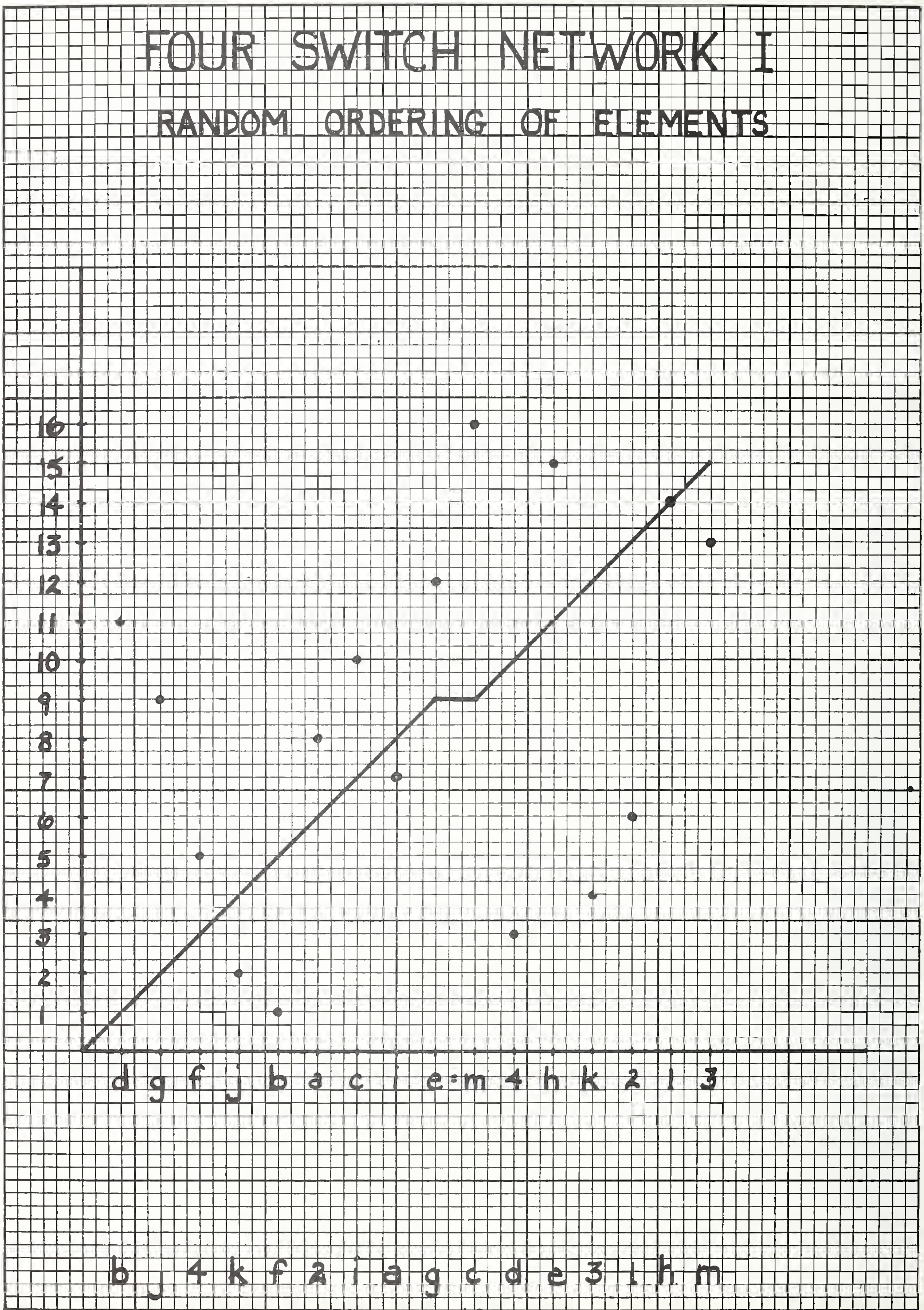
PATH LENGTH : 1 2 3
ADJUSTMENTS : 1. .4 .05



FOUR SWITCH NETWORK I

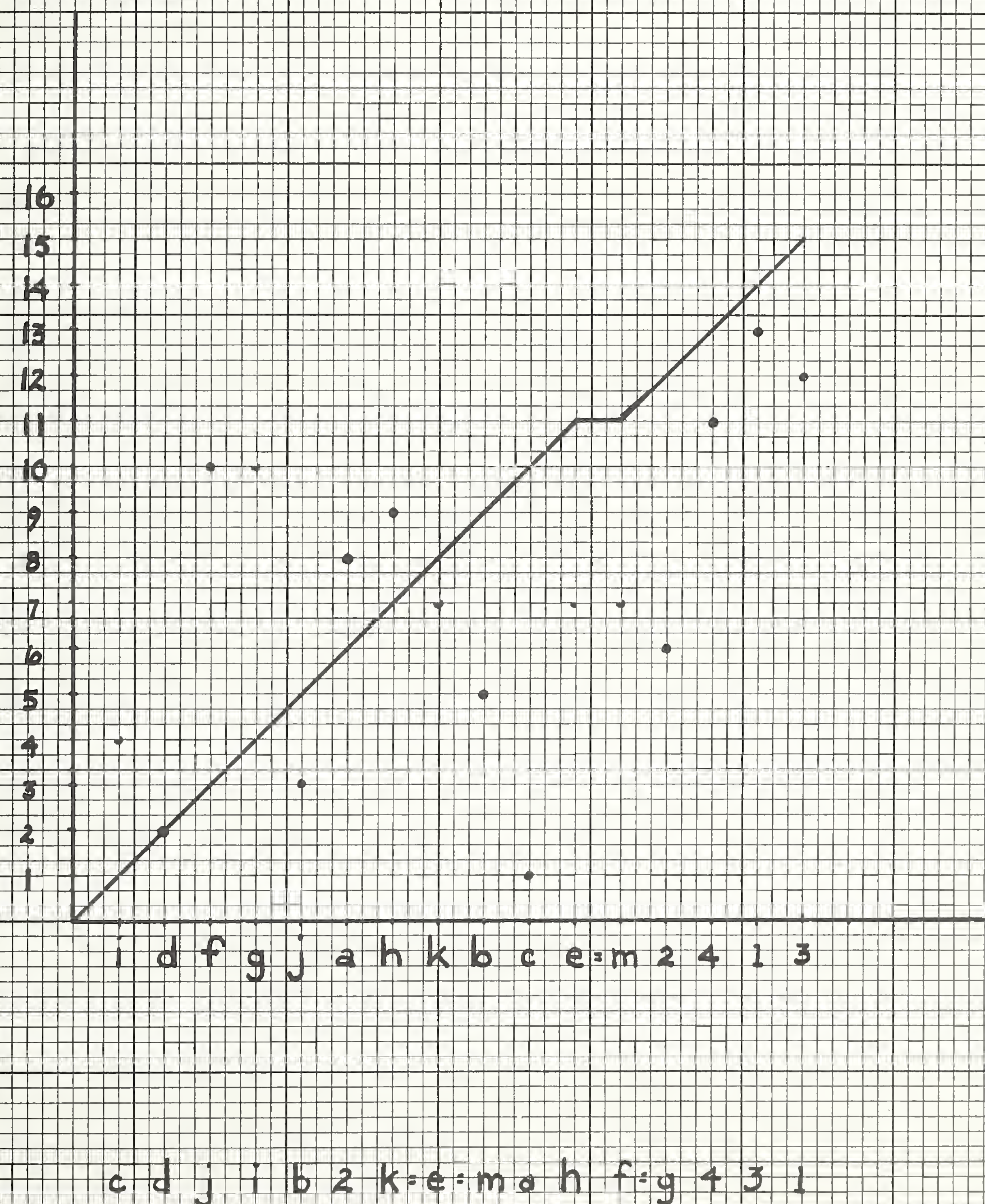
PATH LENGTH: 1 2 3
ADJUSTMENTS: 1. .1 .01

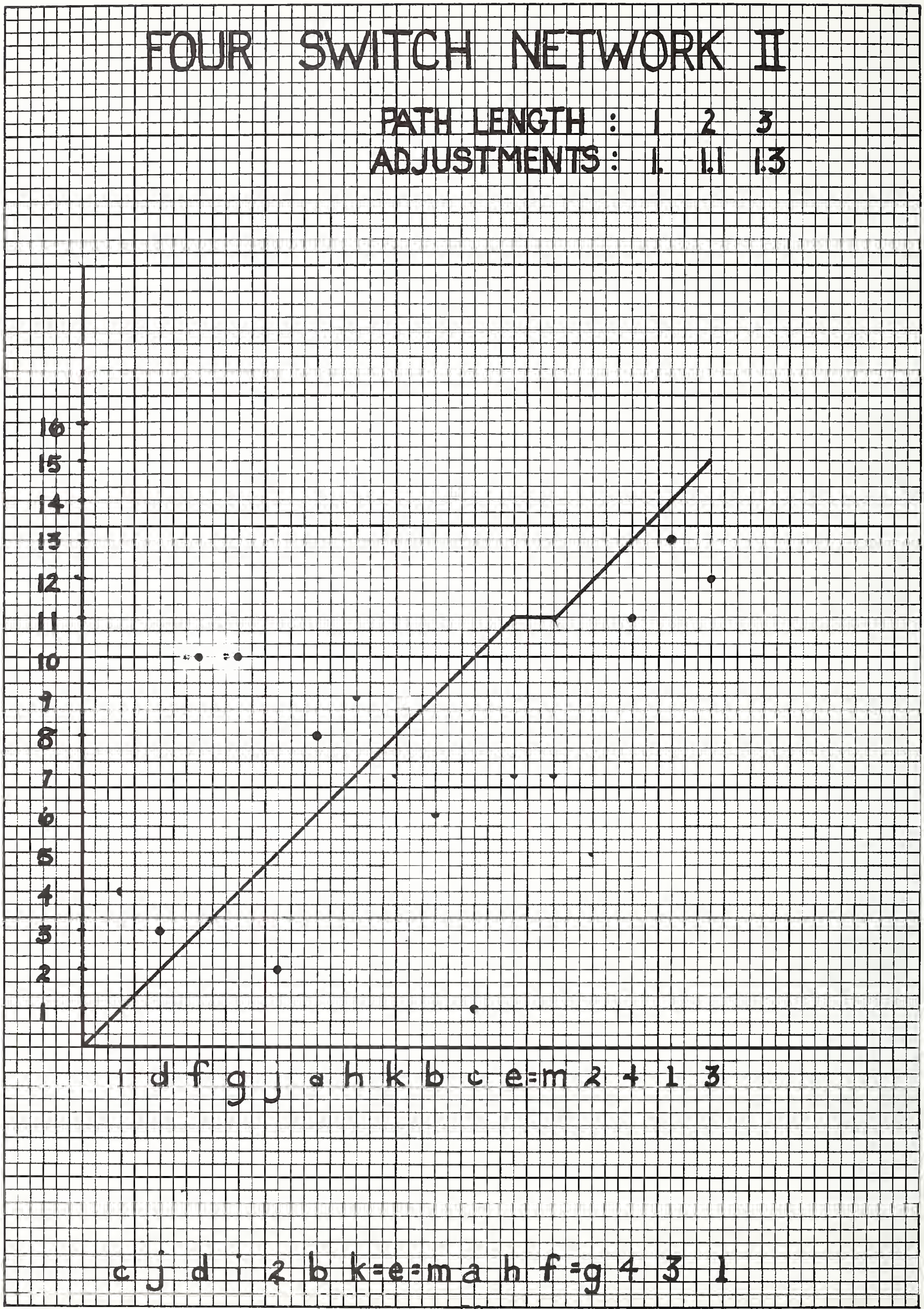



FOUR SWITCH NETWORK I
RANDOM ORDERING OF ELEMENTS
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
d g f j b a c i e=m 4 h k 2 1 3
b j 4 k f 2 i a g c d e 3 l h m

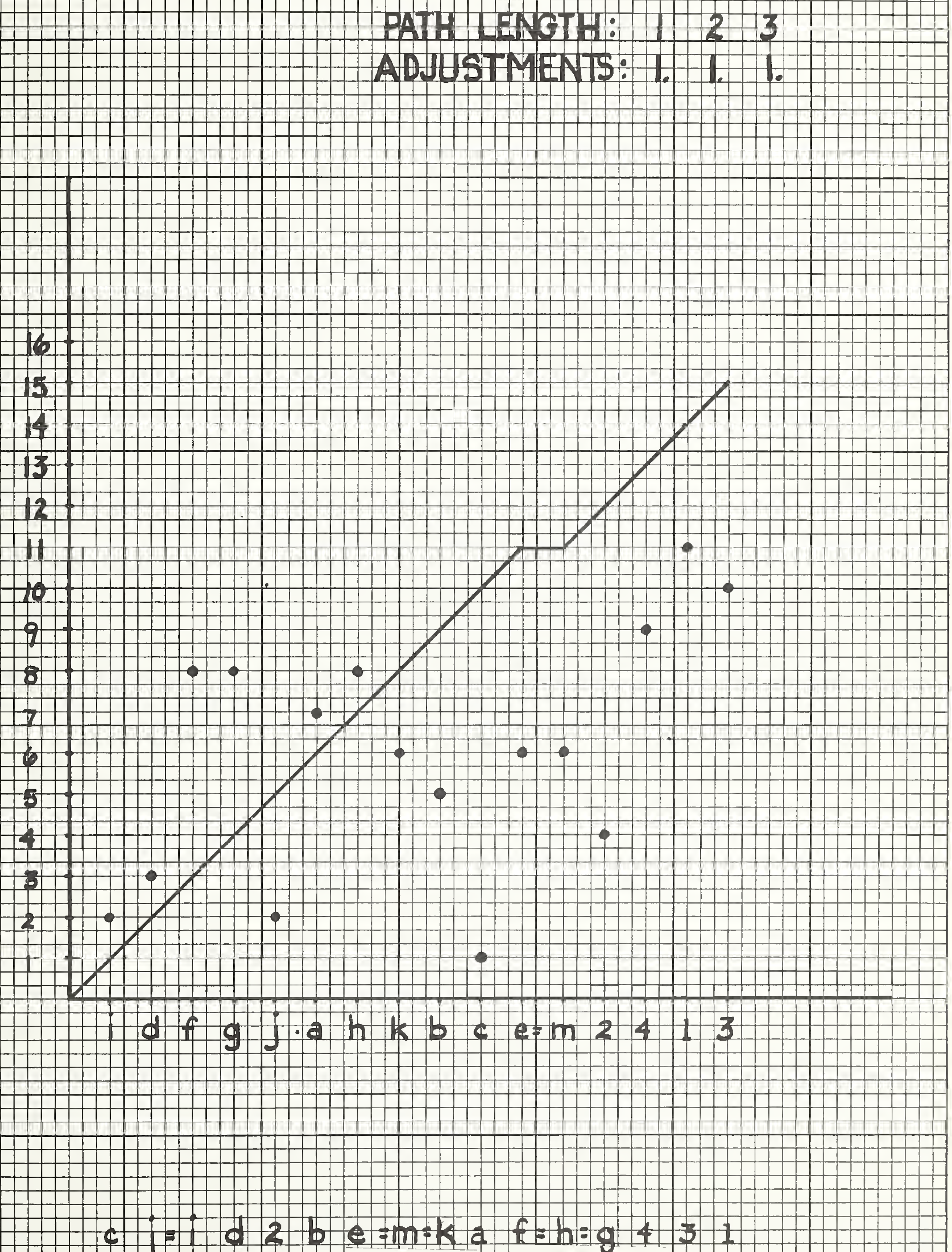
FOUR SWITCH NETWORK II

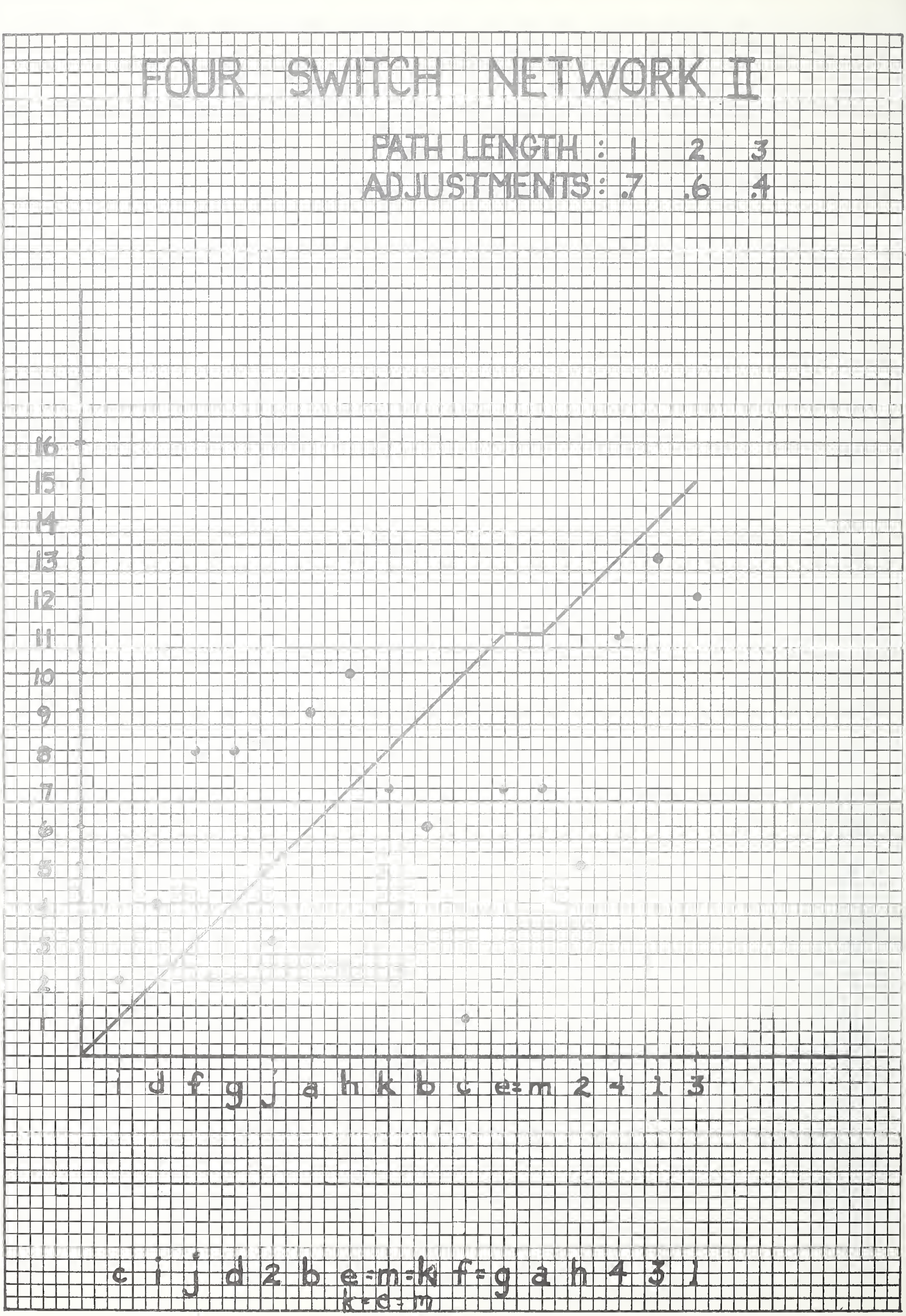
PATH LENGTH : 1 2 3
ADJUSTMENTS : 1. 5. 10.

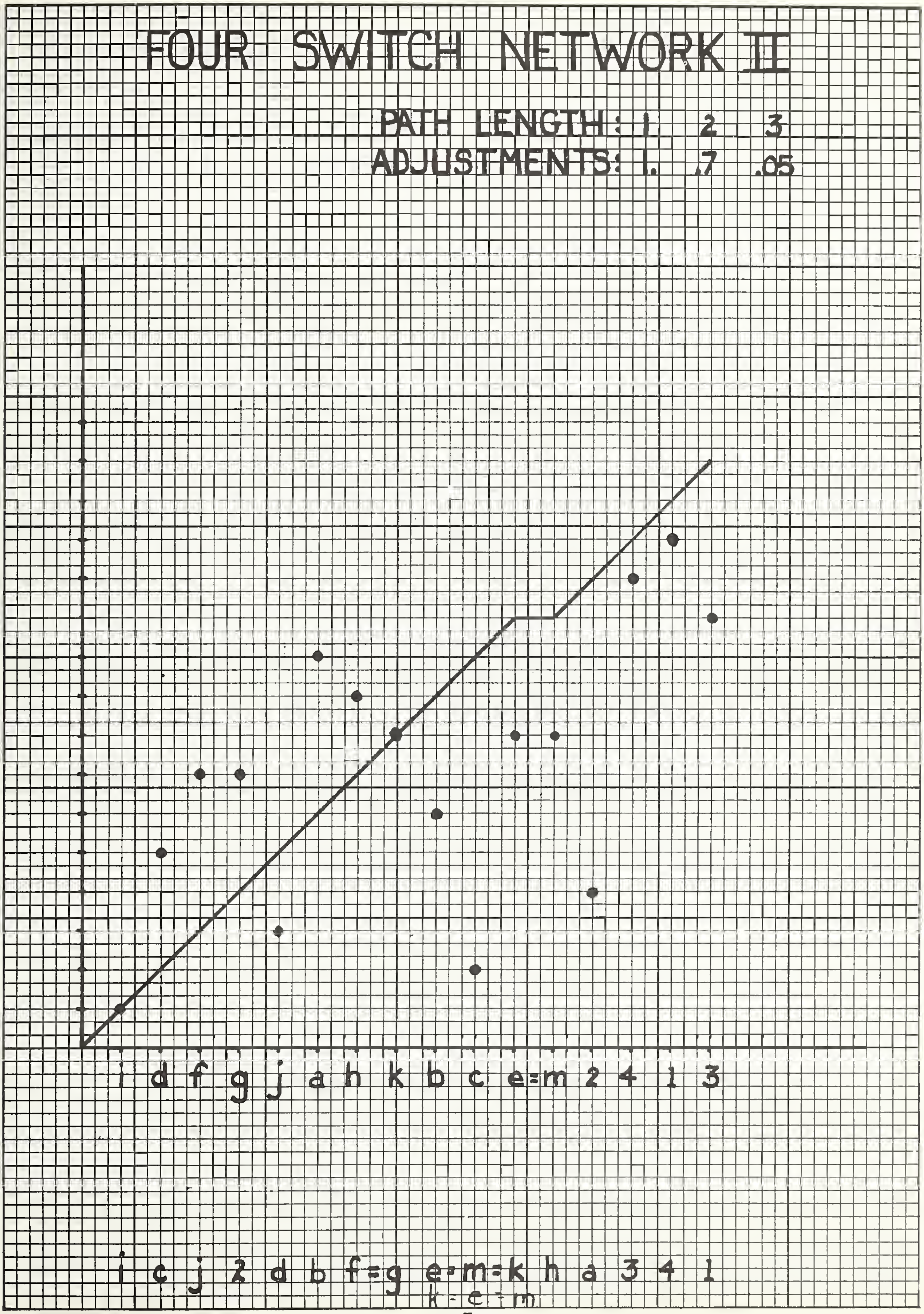



FOUR SWITCH NETWORK II
PATH LENGTH : 1 2 3
ADJUSTMENTS : 1. 1.1 1.3
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
i d f g j a h k b c e=m 2 4 1 3
c j d i 2 b k=e=m a h f=g 4 3 1

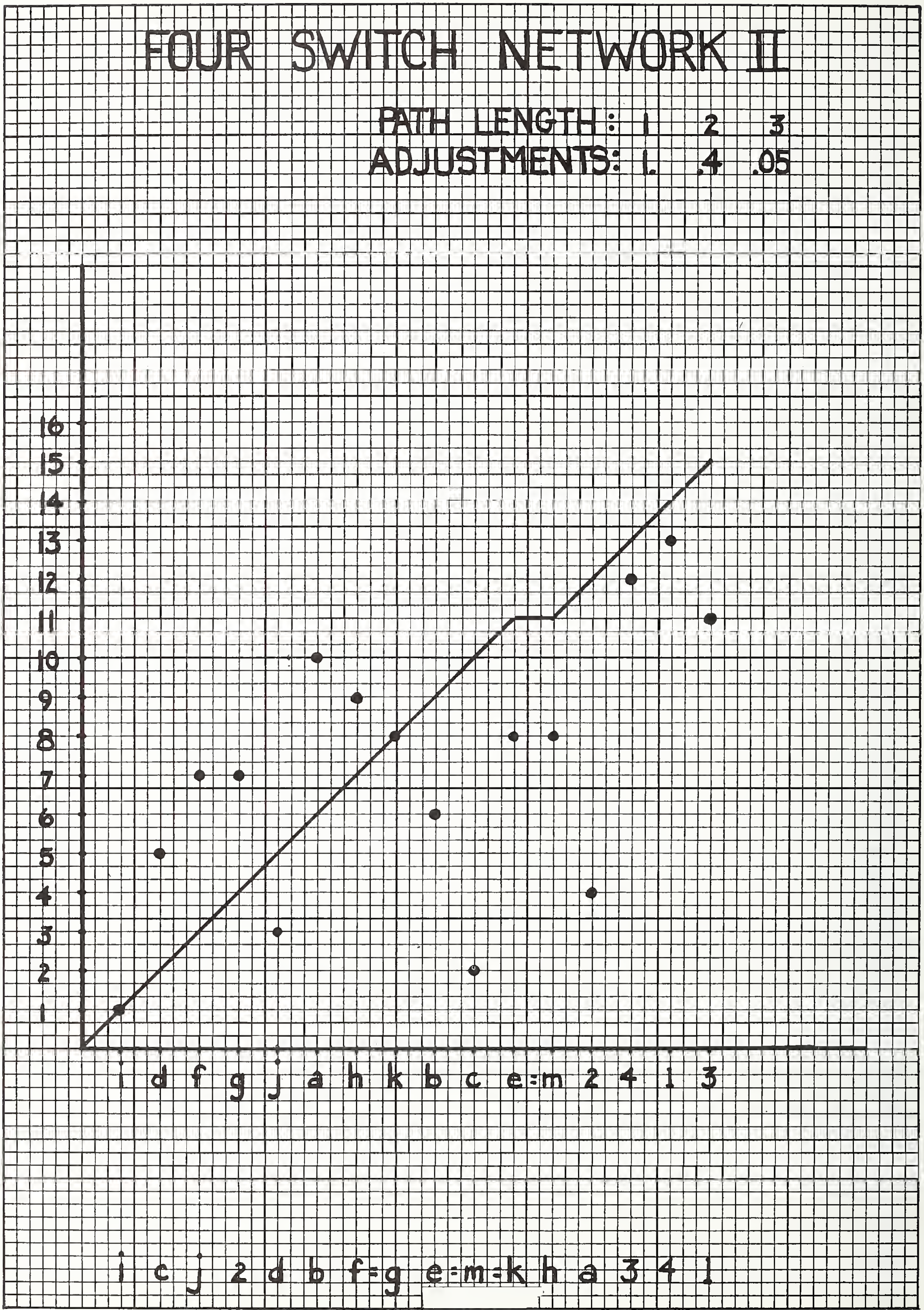
FOUR SWITCH NETWORK II




FOUR SWITCH NETWORK II
PATH LENGTH : 1 2 3
ADJUSTMENTS : .7 .6 .4
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
i d f g j a h k b c e=m 2 4 1 3
c i j d 2 b e=m=k f=g a h 4 3 1
k=e=m

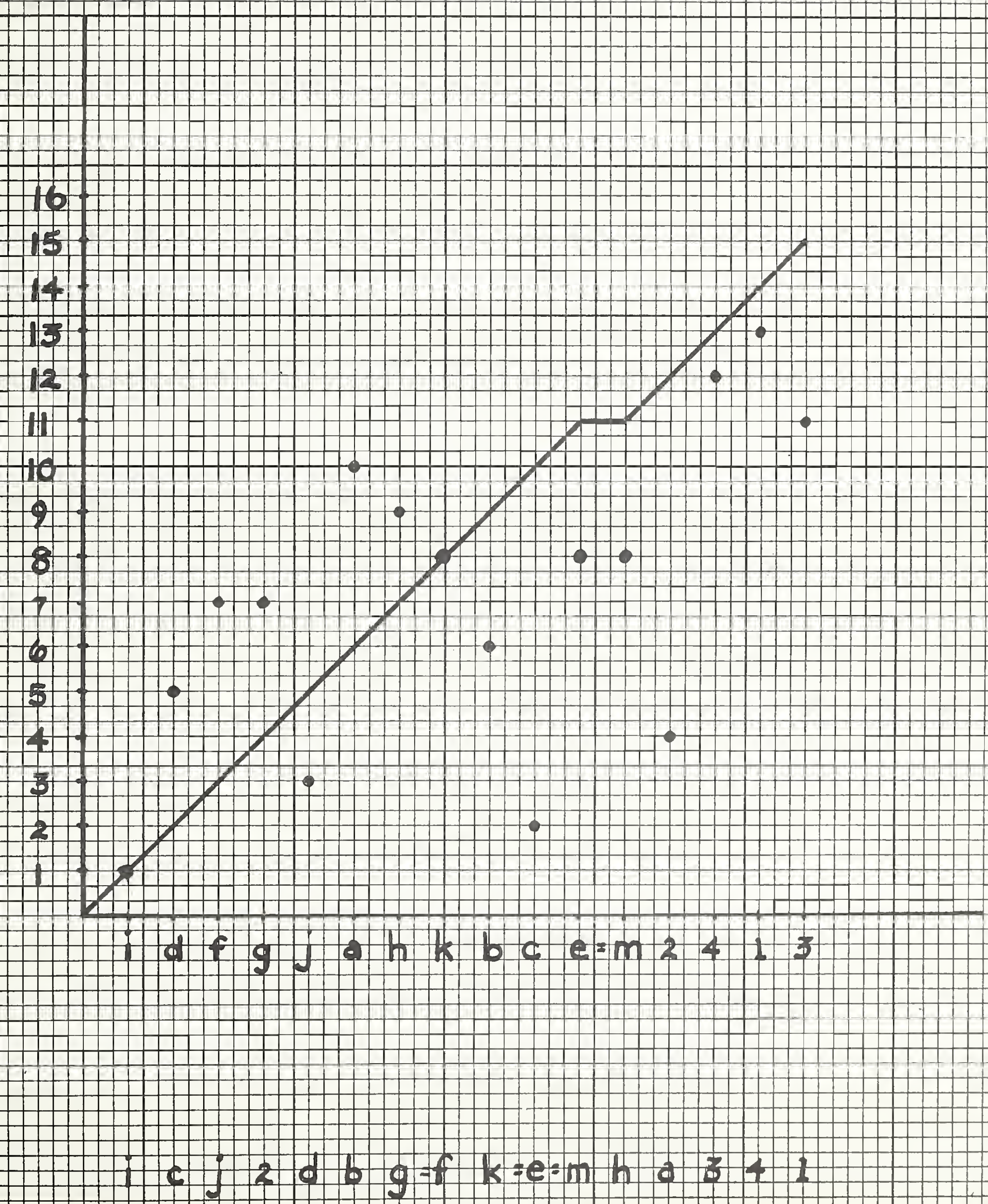


FOUR SWITCH NETWORK II
PATH LENGTH: 1 2 3
ADJUSTMENTS: 1. .7 .05
i d f g j a h k b c e=m 2 4 1 3
i c j 2 d b f=g e=m=k h a 3 4 1
k = e = m


FOUR SWITCH NETWORK II
PATH LENGTH: 1 2 3
ADJUSTMENTS: 1. .4 .05
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
i d f g j a h k b c e=m 2 4 1 3
i c j 2 d b f=g e=m=k h a 3 4 1

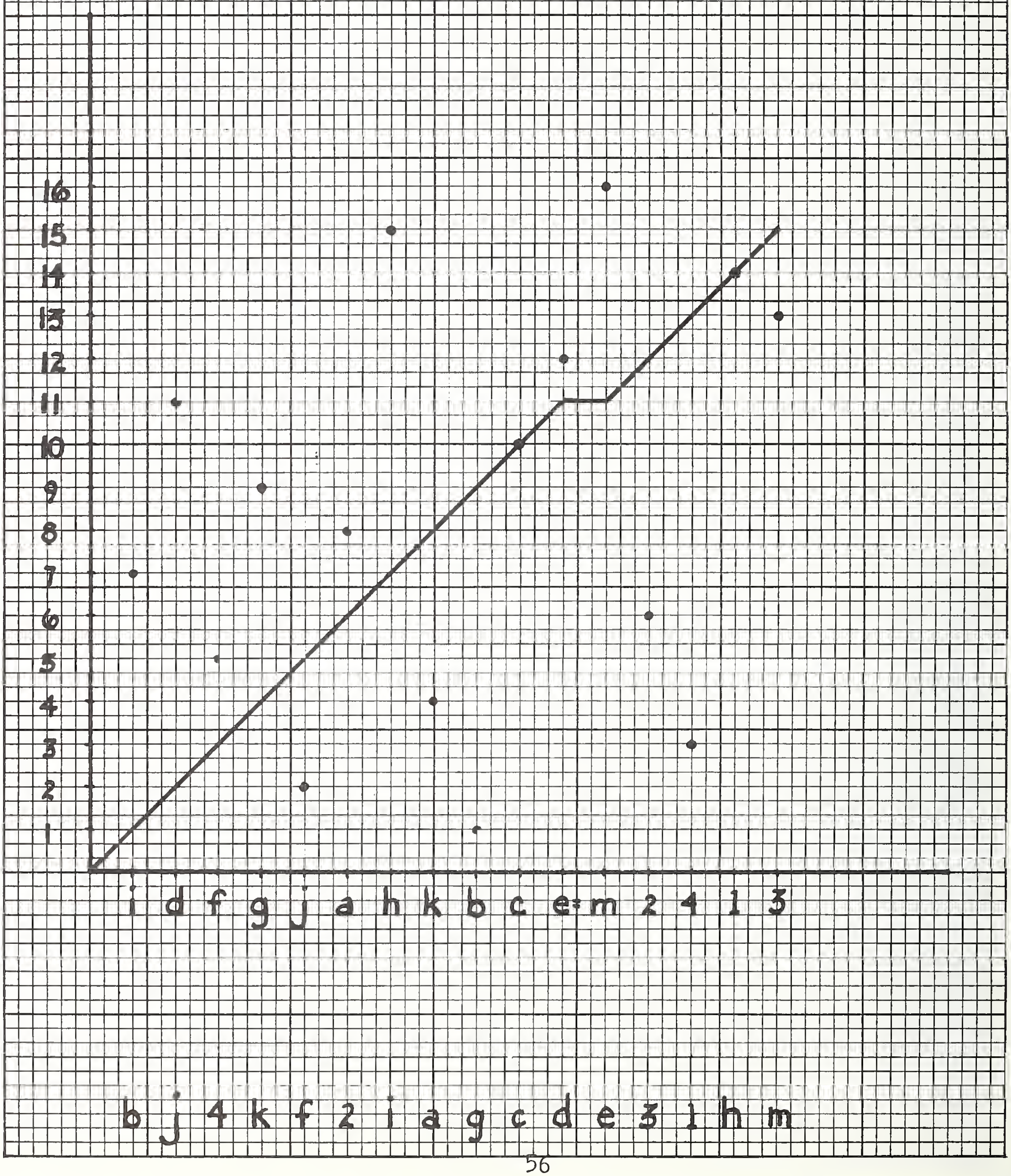
FOUR SWITCH NETWORK II

PATH LENGTH : 1 2 3
ADJUSTMENTS : 1. .1 .01



FOUR SWITCH NETWORK II

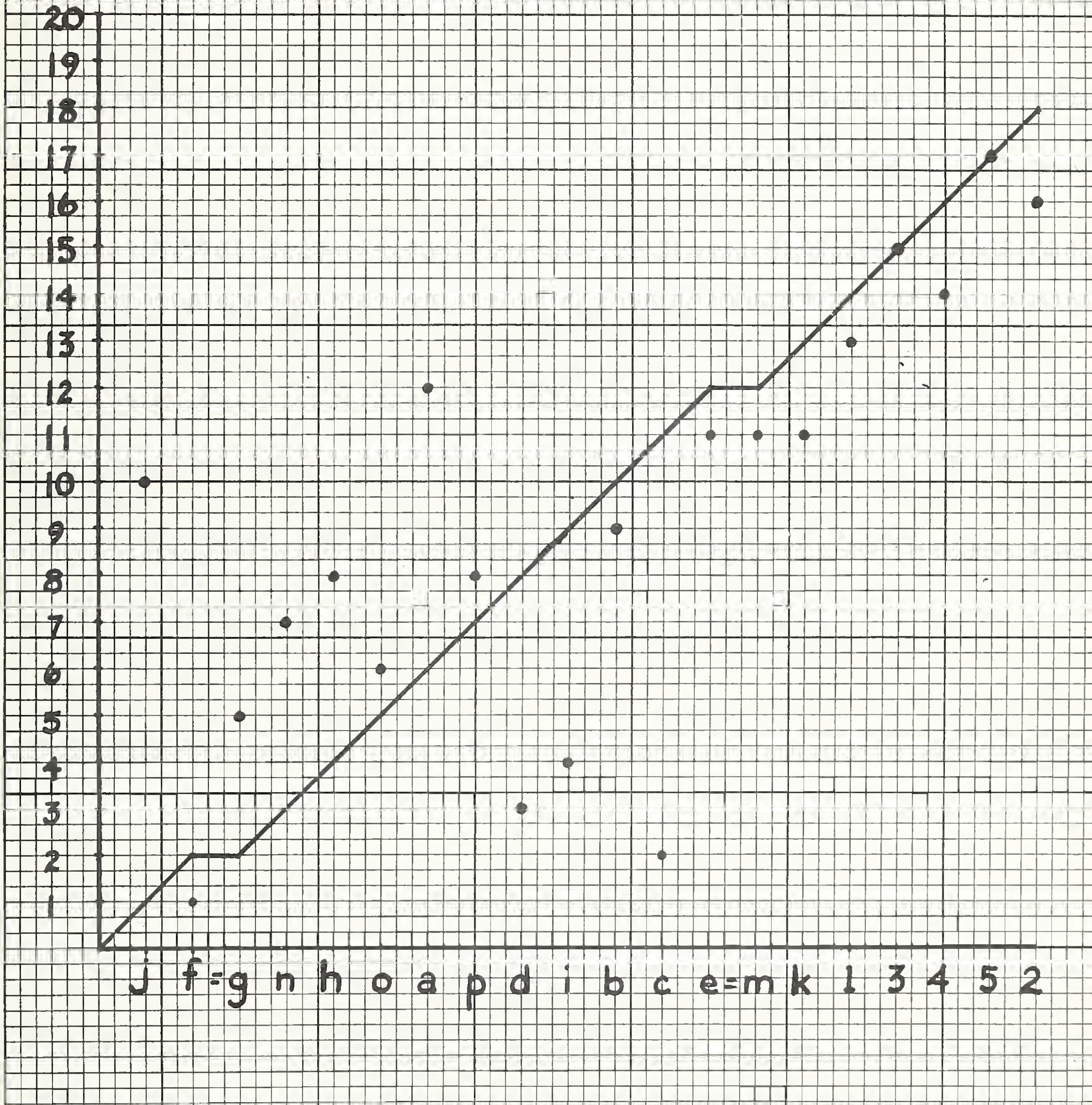
RANDOM ORDERING OF ELEMENTS

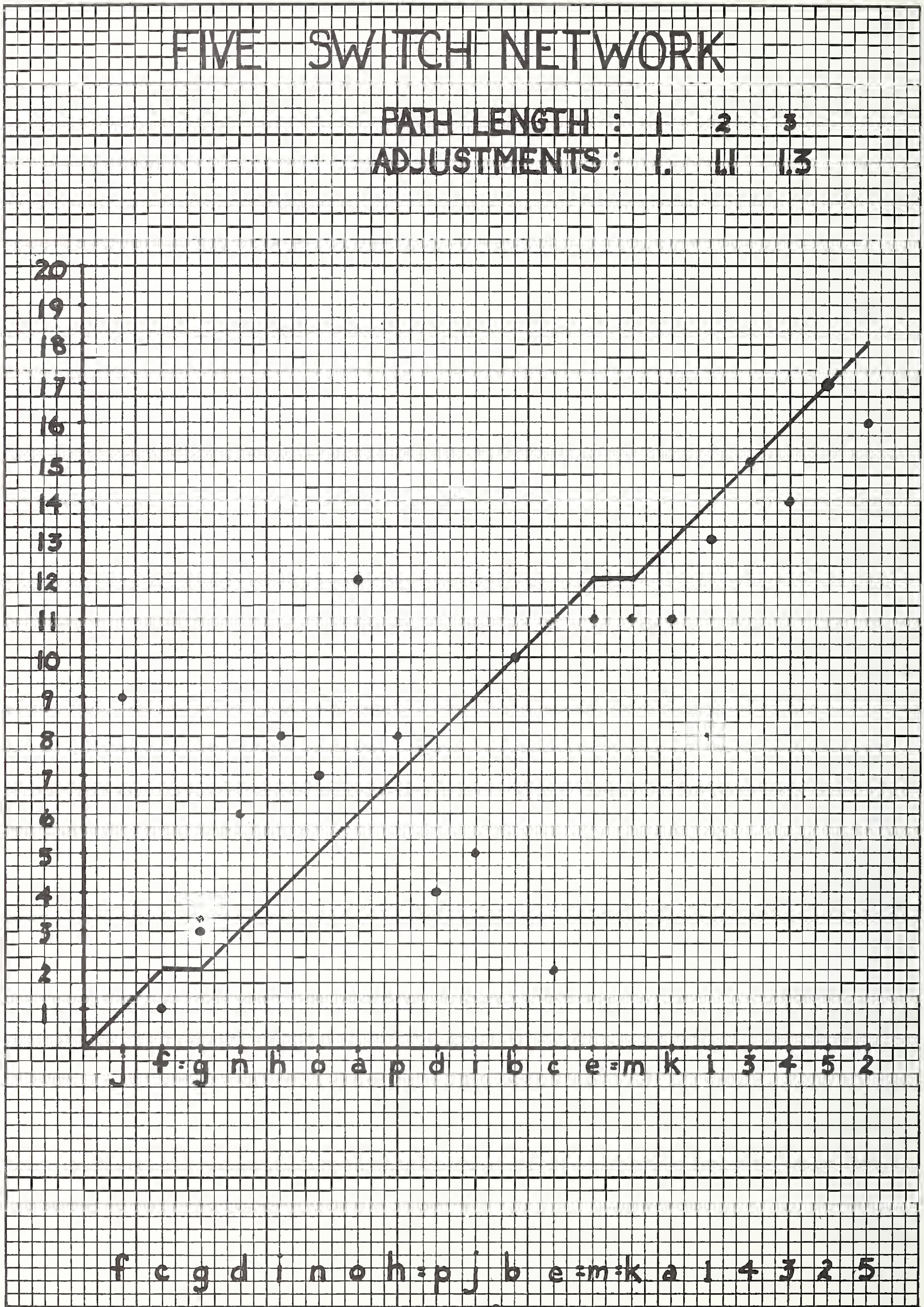


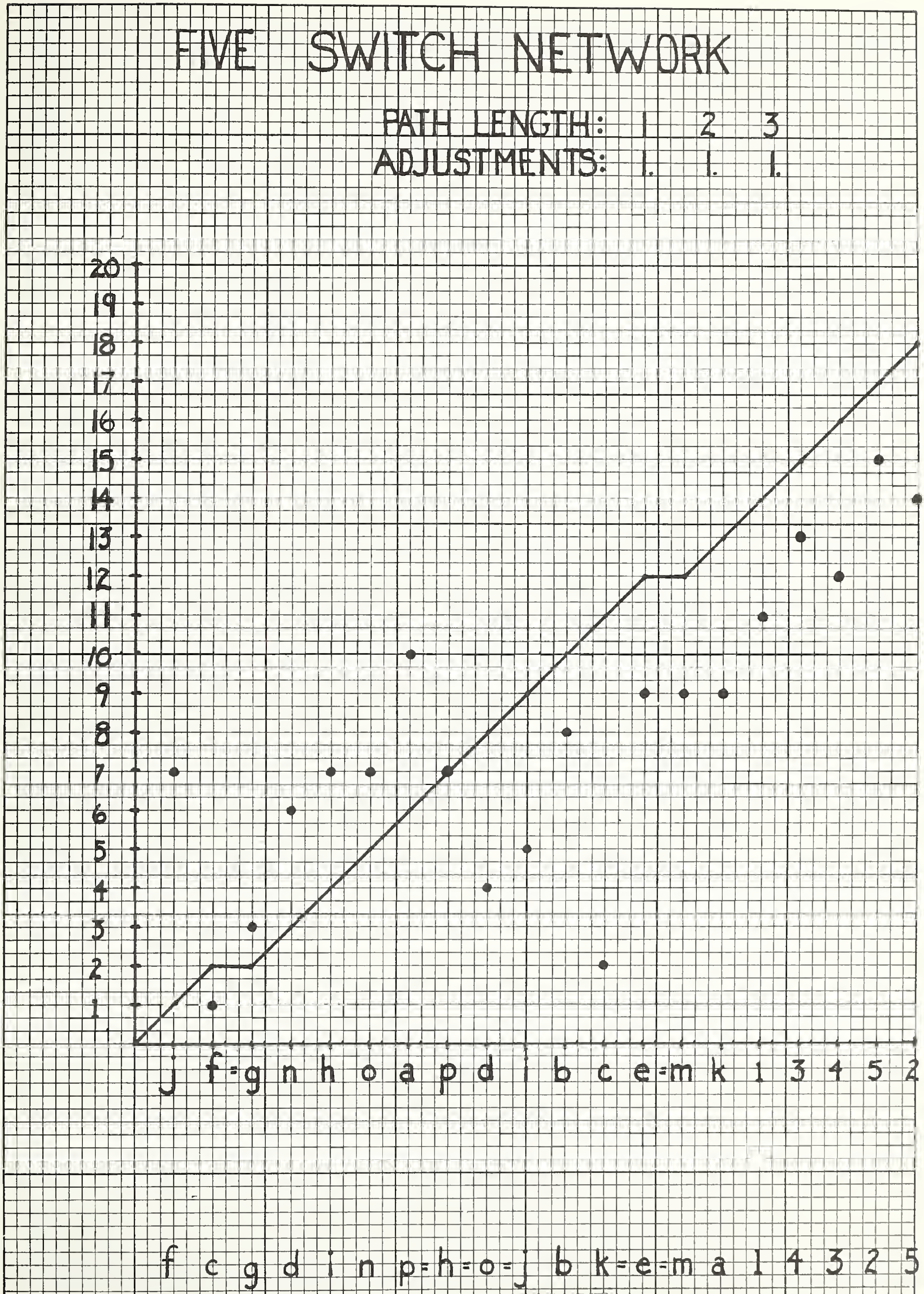
FIVE SWITCH NETWORK

PATH LENGTH : 1 2 3

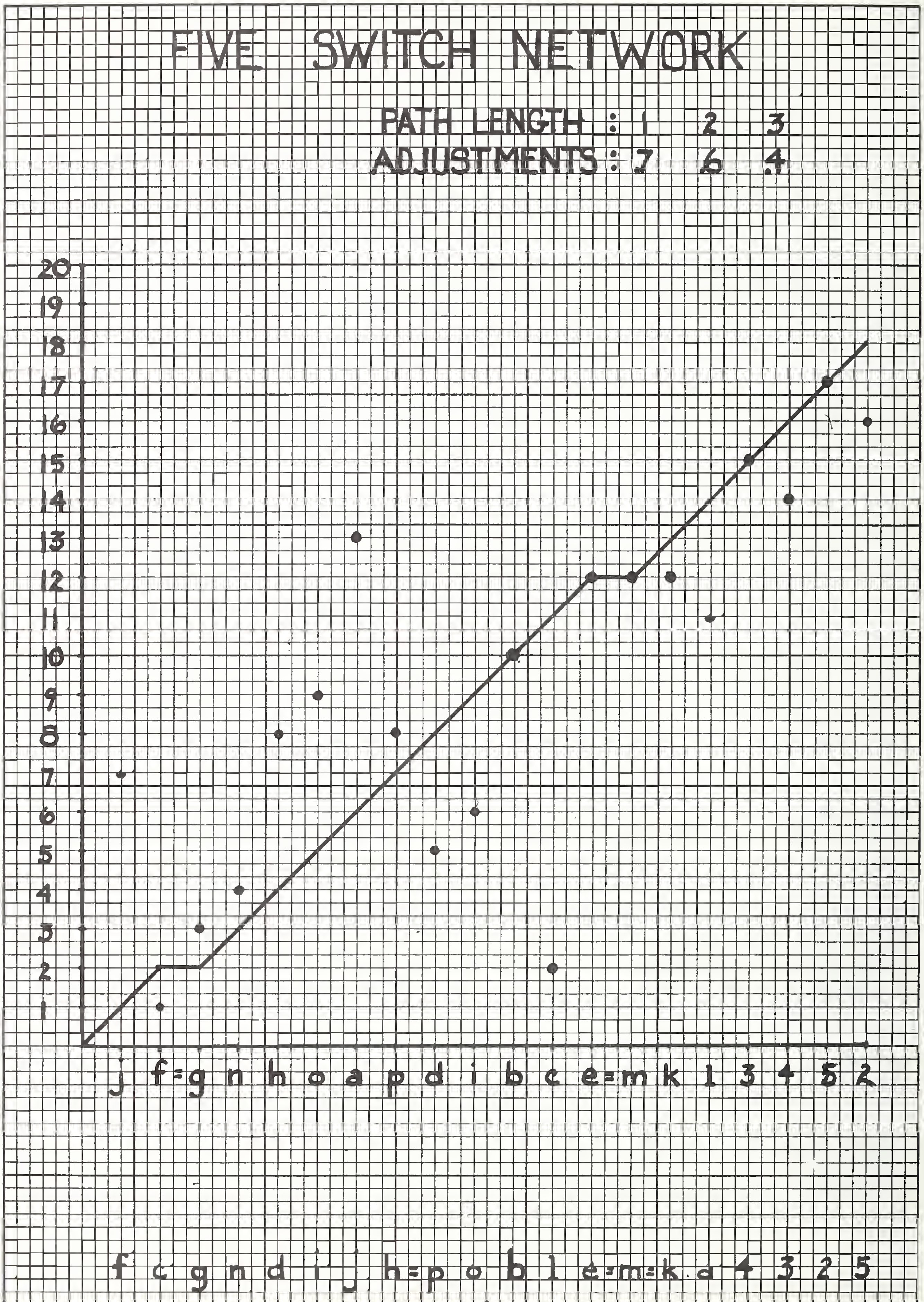
ADJUSTMENTS : 1. 5. 10.

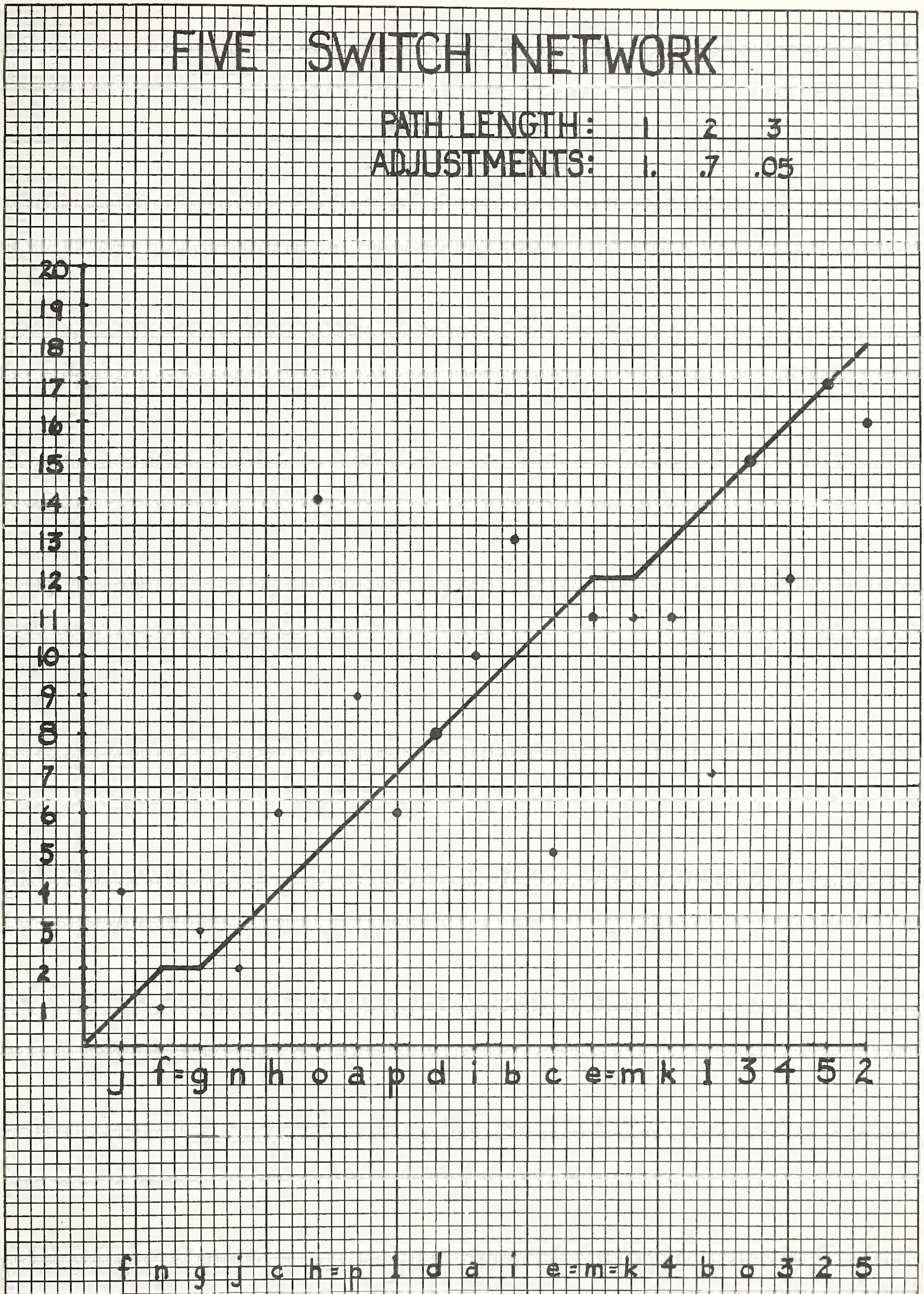



FIVE SWITCH NETWORK
PATH LENGTH : 1 2 3
ADJUSTMENTS : 1. 1.1 1.3
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
j f=g n h o a p d i b c e=m k l 3 4 5 2
f c g d i n o h=p j b e=m=k a l 4 3 2 5



FIVE SWITCH NETWORK
PATH LENGTH: 1 2 3
ADJUSTMENTS: 1. 1. 1.
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
j f=g n h o a p d i b c e=m k 1 3 4 5 2
f c g d i n p=h=o=j b k=e=m a 1 4 3 2 5

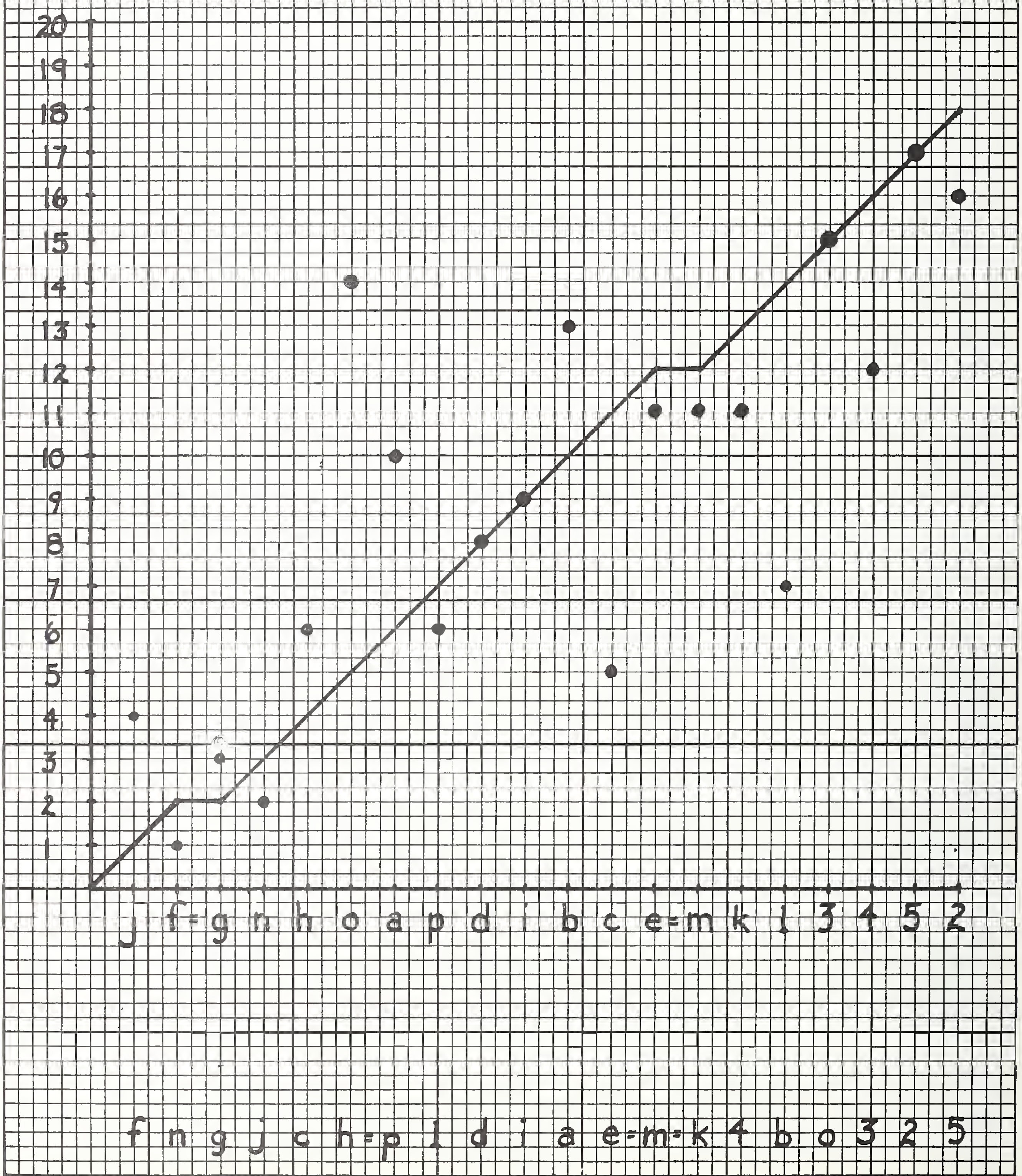

FIVE SWITCH NETWORK
PATH LENGTH : 1 2 3
ADJUSTMENTS : 7 6 4
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
j f=g n h o a p d i b c e=m k 1 3 4 5 2
f c g n d i j h=p o b l e=m=k a 4 3 2 5

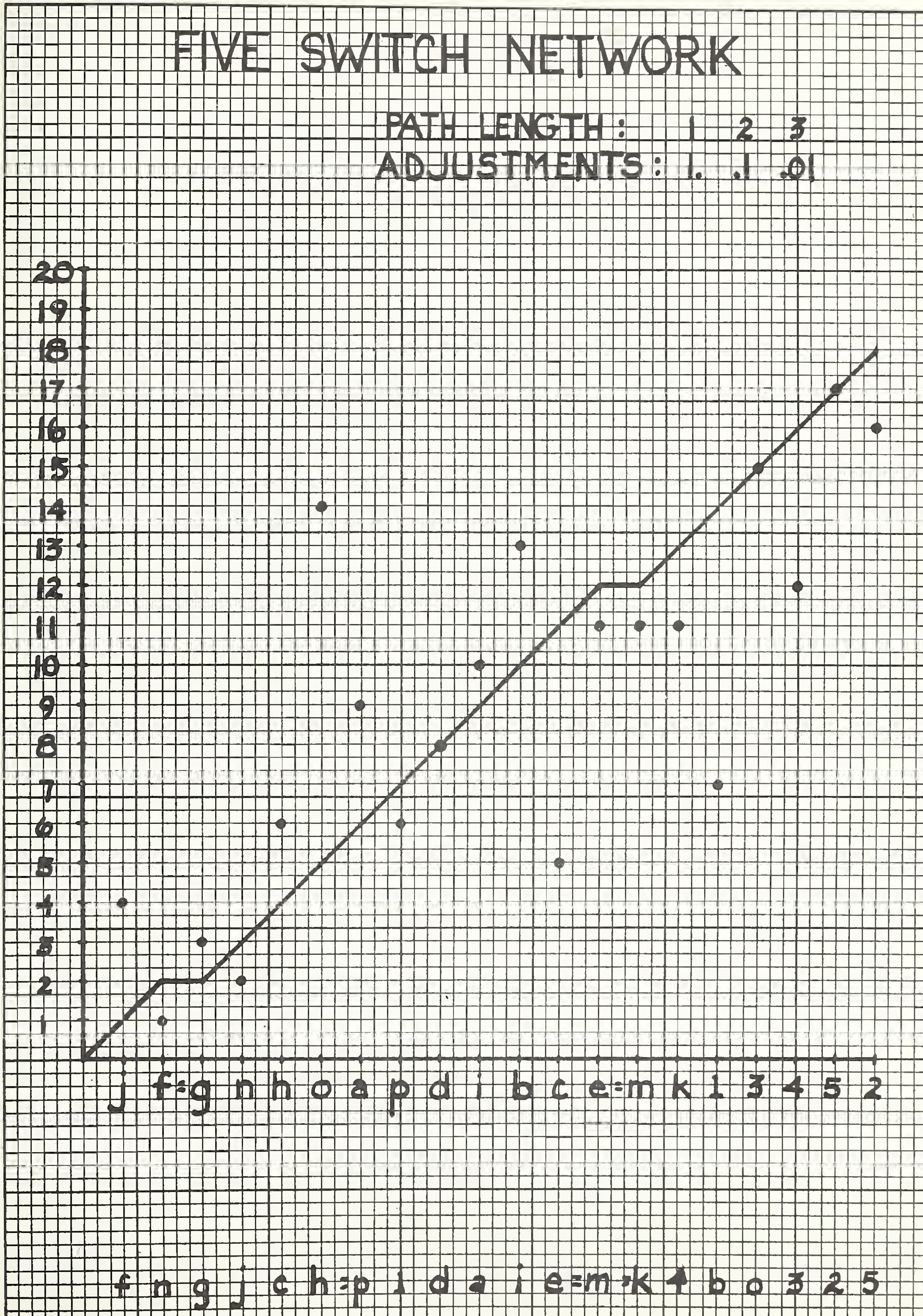

FIVE SWITCH NETWORK
PATH LENGTH: 1 2 3
ADJUSTMENTS: 1. .7 .05
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
j f=g n h o a p d i b c e=m k 1 3 4 5 2
f n g j c h=p 1 d a i e=m=k 4 b o 3 2 5

FIVE SWITCH NETWORK

PATH LENGTH: 1 2 3

ADJUSTMENTS: 1. .4 .05

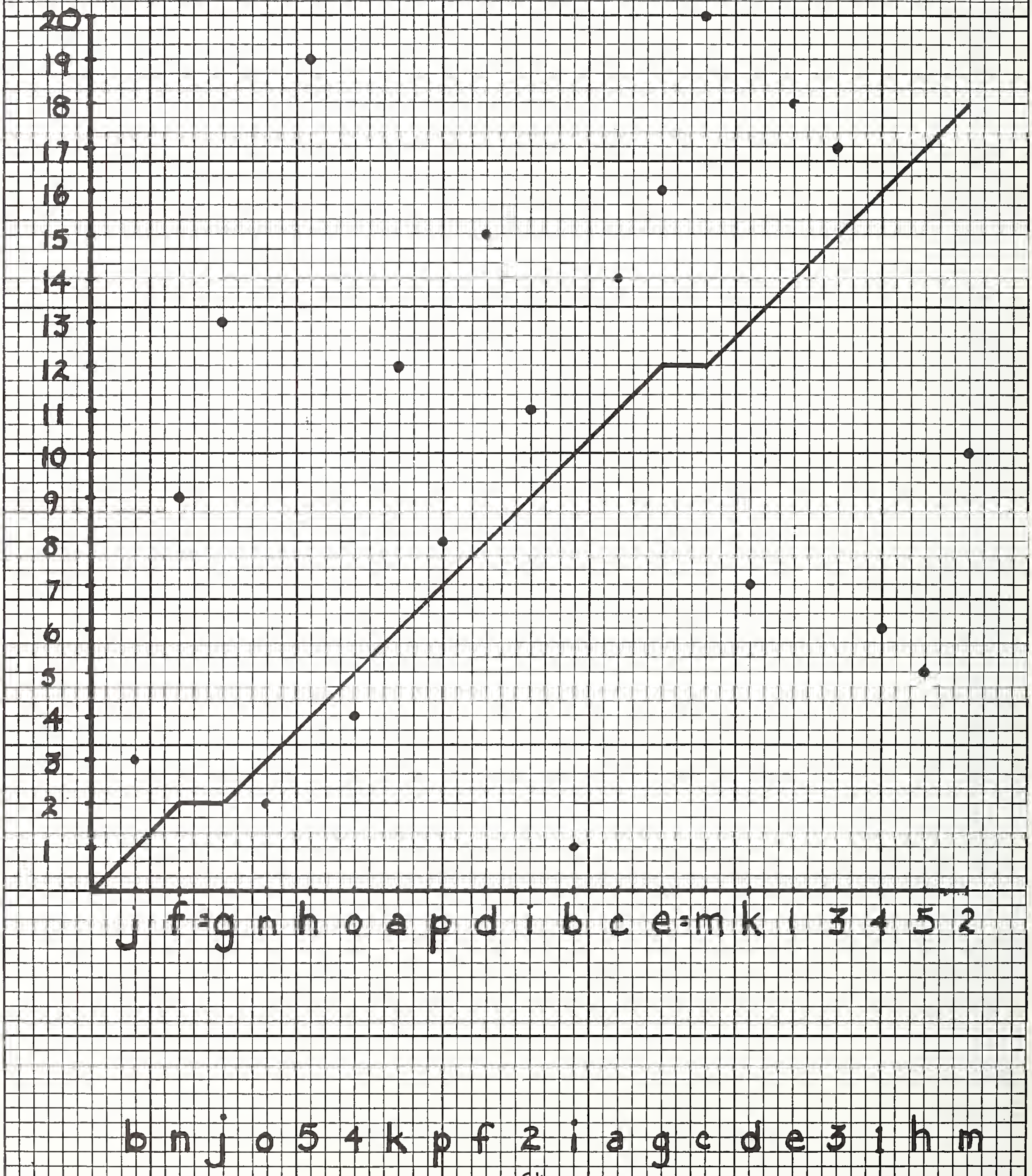




FIVE SWITCH NETWORK
PATH LENGTH: 1 2 3
ADJUSTMENTS: 1. .1 .01
20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
j f=g n h o a p d i b c e=m k 1 3 4 5 2
f n g j c h=p 1 d a i e=m=k 4 b o 3 2 5

FIVE SWITCH NETWORK

RANDOM ORDERING OF ELEMENTS



APPENDIX B

The solution for the following mathematical problem could be directly applied to the problem of determining the optimal number of and locations for the switches and inter-switch links of a military communications network which uses pre-existing civilian facilities with the purpose of attaining the maximum connectivity permitted by those civilian facilities:

Given a graph G, a positive integer k, and a subset S of the set of all paths in G, find a graph H such that:

1) the vertices of H are vertices of G;
2) the edges of H are paths of G;
3) for every path s in S there is a path h in H such that every edge of h is a subpath of s, the first vertex of h is the first vertex of s, the last vertex of h is the last vertex of s, and the number of edges of H in h is less than k; and
4) H minimizes a function from the set of all graphs satisfying conditions 1), 2), and 3) to the set of positive real numbers. (Typically, this function is a cost function.)

REFERENCES

1. Krakowski, M., Additivity and Non-Additivity in Communication Networks, Working Paper, Project 4556455, 12 July 1967, revised 8 August 1967, published as the first part of Some Considerations in Assignment of Values to Elements of Communications Network, by M. Krakowski, Working Paper, Project 4556455, September 1967.

2. Krakowski, M., Linear Programming Approaches to Evaluation of Elements of Communication Network, Working Paper, Project 4556455, 15 September 1967, published as the second part of Some Considerations in Assignment of Values to Elements of Communications Network, by M. Krakowski, Working Paper, Project 4556455, September 1967.

3. Hobbs, Arthur M., Programmers' Manual for the Pathfinding Importance Number Generator, Report of Project 4556455, 6 September 1968, continuation of same, 27 November 1968.

4. C.R.C. Standard Mathematical Tables, Twelfth Edition, page 394, Chemical Rubber Publishing Co., Cleveland, Ohio, 1959.

DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS