1973

The design of an optimal typewriter-like keyboard

R. F. Nickells
Lehigh University

Recommended Citation

Nickells, R. F., "The design of an optimal typewriter-like keyboard" (1973). *Theses and Dissertations*. 4212.
https://preserve.lehigh.edu/etd/4212

THE DESIGN OF AN OPTIMAL TYPEWRITER-LIKE KEYBOARD

by

Robert Frederick Nickells, Jr.

A Thesis

Presented to the Graduate Committee

of Lehigh University

in candidacy for the Degree of

Master of Science

in

Industrial Engineering

Lehigh University

1973

This thesis is accepted in partial fullfillment of the requirements for the degree of Master of Science of Industrial Engineering.

April 12, 1973
DATE

Gary E. Whitehouse
Professor in Charge

Chairman of Department

TABLE OF CONTENTS

CHAPTER SIX

LIST OF FIGURES

ACKNOWLEDGEMENTS

The author would like to express his gratitude to Mr. C. A. R. Kagan for spurring my interest in optimal keyboards and sponsoring my work at the Western Electric Engineering Research Center.

Also, I would like to thank Drs. E. S. Binkowski and R. P. Thayer at the Research Center and Dr. S. Munro of Lehigh University for their continual reassurance of the statistical significance of my results.

And finally, I must thank Dr. G. E. Whitehouse for his many suggestions which expedited my efforts and his continual effort in keeping me going every time I lost sight of my objectives.

PREFACE

In view of the fact that this thesis discusses keyboards and typewriters, the author felt a moral obligation to do his own typing. In order to achieve this as efficiently as possible, this thesis was composed at an interactive computer terminal using the Western Electric String Language Processor. A string language can be thought of as the most flexible of all possible text-editing languages and enables the user to define his own procedures for whatever unique requirements he might have.

Unfortunately, short of using a plotter, computer output devices are not suited for the printing of text containing conventional scientific and mathematical notational schemes. Therefore the author was forced to use the standard notation used by most programming languages for subscripts. In other words, X(I,J) does not mean that X is a function of I and J, but rather that X is an array with address paramenters I and J. It is hoped that this notation will not be too confusing to the reader with no computer programming experience.

The following references are given for the reader who might be interested in string language processors.

Kagan, C.A.R., "Dictionary of Built-in Functions for String Language Processors", IEEE Computer Society Repository, Ref. R73-1, 1972

Kagan, C.A.R., "A String Language Processor for Small Machines", Proceedings of the ACM, SIGPLAN Symposium on the Pedagogical Applications of Small Computers, University of Kansas, 1972

ABSTRACT

This thesis investigates the design of an optimal typewriter-like keyboard. An optimal keyboard is defined as one whose design is based upon the statistics of usage of the English Language and the human factors of the typists. The design goal of such a keyboard is to minimize an objective function which relates the relative digraph frequencies (two letter combinations) and the stroking times of two key combinations.

After rejecting standard mathematical programming techniques because the objective function lacked the required "nice" properties, simulation was attempted. While simulation was successful in that it did find several "better" keyboards it was rejected because the size of the population is such that the probability of finding a keyboard near the lower bound of the distribution is approximately zero for practical sample sizes.

A simple branching algorithm was then developed which evaluates all possible exchanges of pairs of assignments of a given layout and chooses the best as the input for the next iteration. The algorithm is stopped when no further improvement is possible. The performance of the algorithm seems to be independent of the value of the input keyboard and converges in an exponential-like manner to within a very narrow range.

After an examination of the mechanisms governing the performance of the algorithm, it is concluded that the algorithm is converging to a value near the lower bound of the distribution of keyboard values and thus, represents a viable technique for developing a near-optimal keyboard, provided accurate data is available.

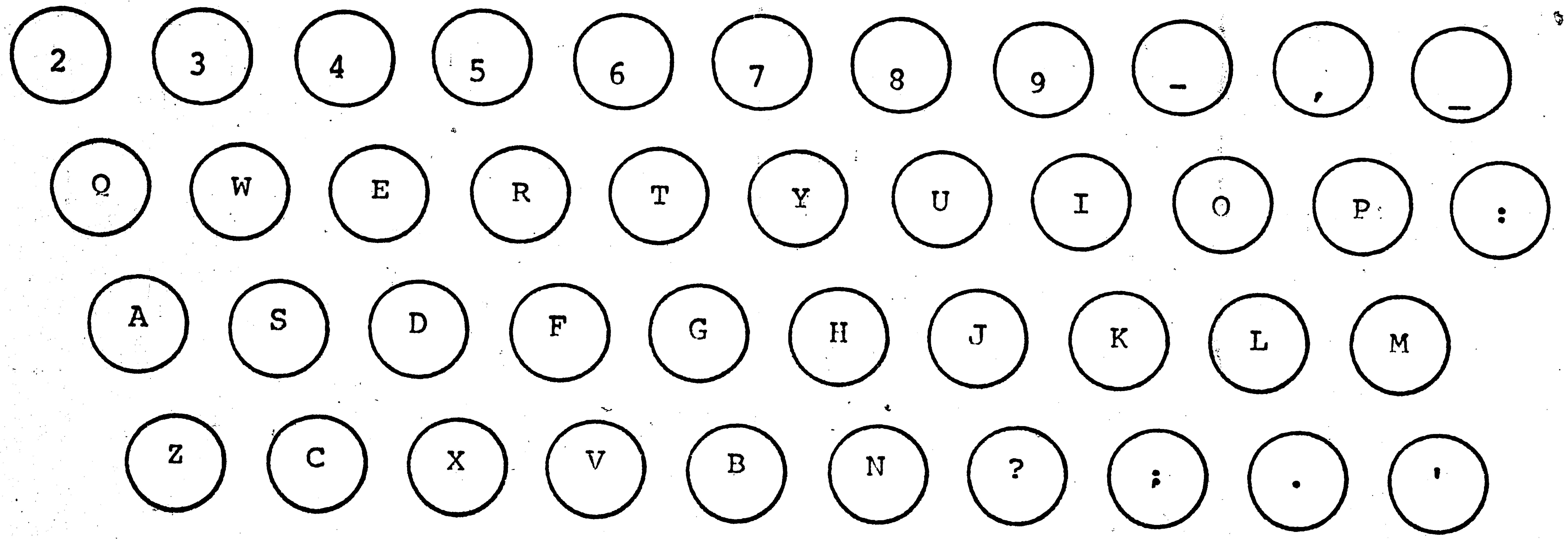
CHAPTER ONE

Introduction to Improved Keyboards

In 1873 Christopher Latham Sholes and the Remington Arms Company introduced the first typewriter capable of being mass produced. Within a few years the typewriter had become an indispensible tool of our society--as it remains today.

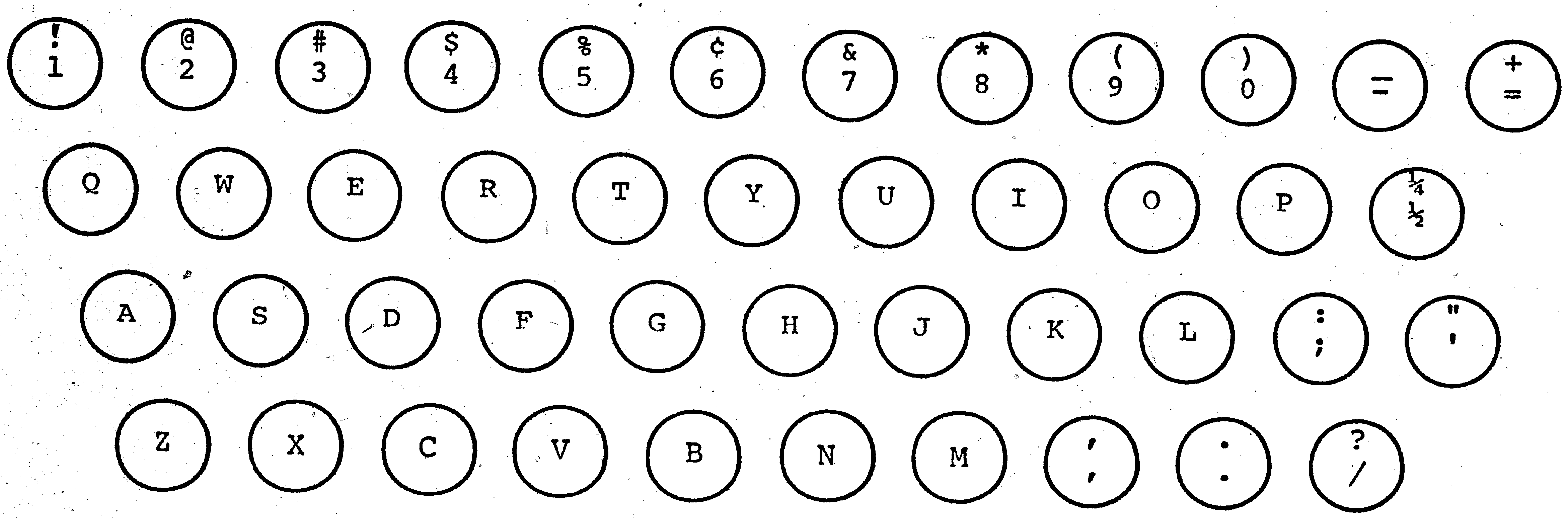
The keyboard design of the Sholes-Remington typewriter (See Figure 1.1) is essentially identical to that used on our typewriters today. Figure 1.2 shows the current typewriter keyboard as standardized by the American National Standards Institute. [1] The intent of this work is to examine in detail, not the mechanical design of the keyboard (44 keys arranged in four rows), but the assignment of graphic symbols or characters to the key positions.

Legend has it that Sholes had to design his keyboard to circumvent the mechanical limitations of his machine. [2] Even though it was intended that the typist would type with only two fingers (touch-typing was not developed until the twentieth century), the Sholes prototypes tended to jam frequently. Jamming was caused by adjacent typebars being activated too quickly. Jams were difficult to clear because the typebars were located beneath the carriage and as a result, Sholes reacted by developing a layout which he thought would minimize jams by forcing the operator to slow



REMINGTON-SHOLES KEYBOARD (1873)

FIGURE 1.1



ANSI X4.7 STANDARD TYPEWRITER KEYBOARD

FIGURE 1.2

down on those two-letter combinations assigned to adjacent typebars. An examination of an early typewriter supports the legend and makes it appear quite reasonable.

Regardless of why Sholes designed his keyboard as he did, the fact that the same keyboard designed in 1873 is still in use a hundred years later, in spite of significant advances in machine technology, typewriting techinques, and knowledge of human factors, must appear as an anomoly in the history of social and scientific progress.

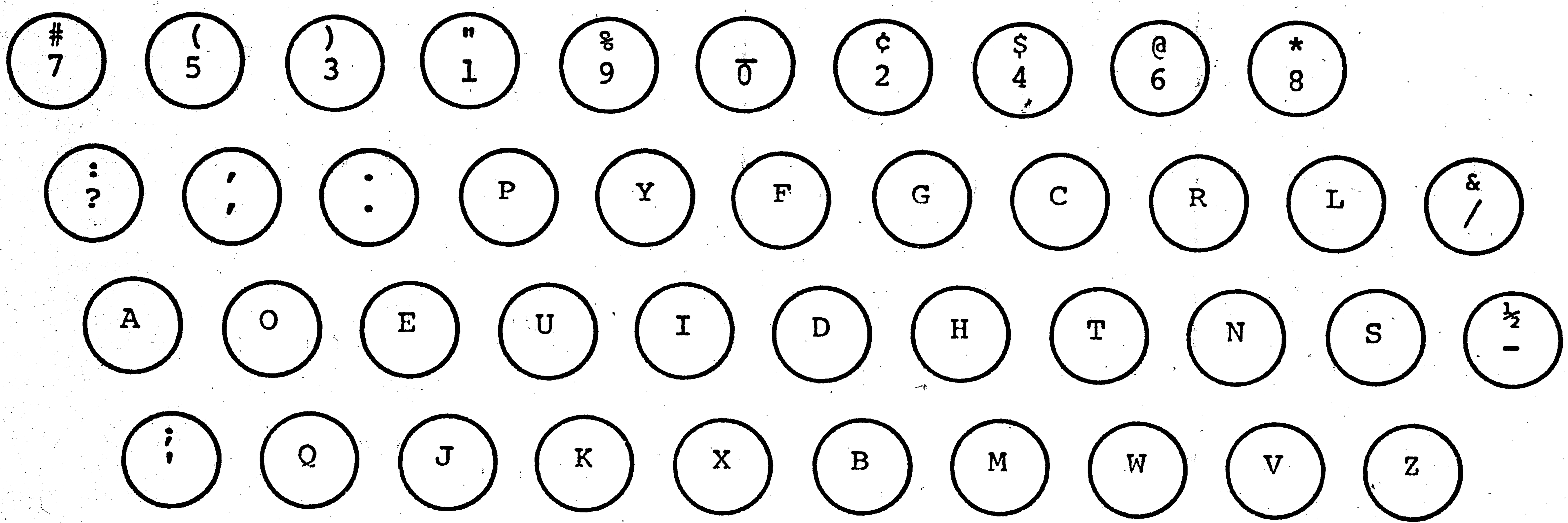
Even before the advent of touch-typing, several typewriter manufacturers attempted to market machines with different keyboard layouts (the reasons for the changes are unknown) but the market position of Remington was such that the Sholes keyboard rapidly became the defacto standard.

In the early twentieth century the all-finger typewriting technique was developed and typing became a popular course in high schools and colleges. Unfortunately, educators soon learned that typing was a difficult skill to teach effectively. By the 1920's, the dichotomy between the ever increasing popularity of typing courses and the obvious difficuly students had in acquiring commercialy acceptable typing skills motivated the Carnegie Foundation for the Advancement of Teaching to provide funds in 1929 to enable Drs. August Dvorak and William Dealey to study the teaching of typing. [3]

Dr. Dvorak worked with Frank and Lillian Gilbreth in conducting micromotion studies of typewriting. The micromotion studies suggested, and analysis of the frequencies of usage of letters and combinations of letters in the English language confirmed, that the standard keyboard (commonly known as the QWERTY keyboard) was an extremely poor design for the typing of English language text. While the detailed faults of the QWERTY keyboard will be discussed in detail in the next chapter, the QWERTY was inadequate because high frequency letters and combinations of letters were assigned in a manner that required awkward and inefficient finger movements.

Dr. Dvorak then developed what has since become known as the Dvorak Simplified Keyboard (DSK). Figure 1.3 shows the DSK. Although the DSK has not been widely accepted or even adopted by any large organization, there can be no doubt that it is indeed, vastly superior to the QWERTY in terms of ease of learning, efficiency, throughput, and even inherent speed capabilities.

NOTE: Even though what follows cannot fail to be interpreted as an enthusiastic endorsement of the DSK--which it really is--the author contends that an even better keyboard is possible. The emergence of electric typewriters and electronic keyboards has



DVORAK SIMPLIFIED KEYBOARD

FIGURE 1.3

sufficiently altered the relative difficulties of stroking patterns to allow even more opportunity for improvement.

As part of the Carnegie Study, the DSK was tested in the Tacoma, Washington schools in 1931 and 1932. [4] The 250 junior high school students trained on the DSK achieved typing skills in one semester (an average of 27.1 words-per-minute) that took QWERTY students three semesters to attain (26.8 wpm). In two semesters, the junior high students outperformed four semester QWERTY students (36.1 wpm-vs-33.4 wpm). The 110 senior high students performed even better. After one semester they outperformed three-semester QWERTY students (37.5 wpm-vs-35.0 wpm) and after two semesters had outperformed six-semester QWERTY students (48 wpm-vs-47 wpm).

Prior to World War II, the International Typing Contests achieved a high degree of popularity. Most major typewriter manufacturers maintained "stables" of professional typists who gave demonstrations and competed in contests in order to demonstrate the superiority of the machines manufactured by their sponsors. Between 1933 and 1941 DSK typists won 119 first, second, and third place awards at the International Typing Contests. [5]

The final evidence of the superiority of the DSK over the QWERTY are the results of retraining experiments. Only two significant experiments have been conducted which attempted to determine what happened when QWERTY trained typists were retrained on the DSK.

The first was an experiment performed by the U.S. Navy in 1944. Briefly, in the Navy Study, 14 QWERTY-trained typists were given an average of 83 hours of training on the DSK. After completion of their training, the typists had increased their performance from 32.9 net wpm on the QWERTY to 57.1 net wpm on the DSK, a 74% increase. This was composed of a 25% increase in gross speed and a 68% decrease in errors. In a companion experiment, 18 QWERTY typists received an average of 158 hours of additional training on the QWERTY. They achieved an increase of 17% in gross speed and a 17% decrease in errors for an increase of 43% in net speed. [6]

In 1956, the Government Supply Agency (GSA) commissioned Dr. Earl Strong to conduct a comparative experiment with ten typists retraining on the DSK and another ten typists receiving additional training on the QWERTY. Unfortunately, this study has become the stumbling block to serious consideration of the DSK and is mentioned only to point out why it is better forgotten. [7]

As a detailed analysis of the GSA Study is inappropiate in this paper, the following comments are offered to refute the findings of the study. If interested, the reader is urged to obtain a copy of the report from the GSA for detailed analysis.

The GSA Study subjects spent four hours each day in class. This is in opposition to standard practice of allowing a maximum of two hours per day.

The GSA Study showed the QWERTY subjects increasing their average gross speed from 83.5 wpm to 113 wpm.

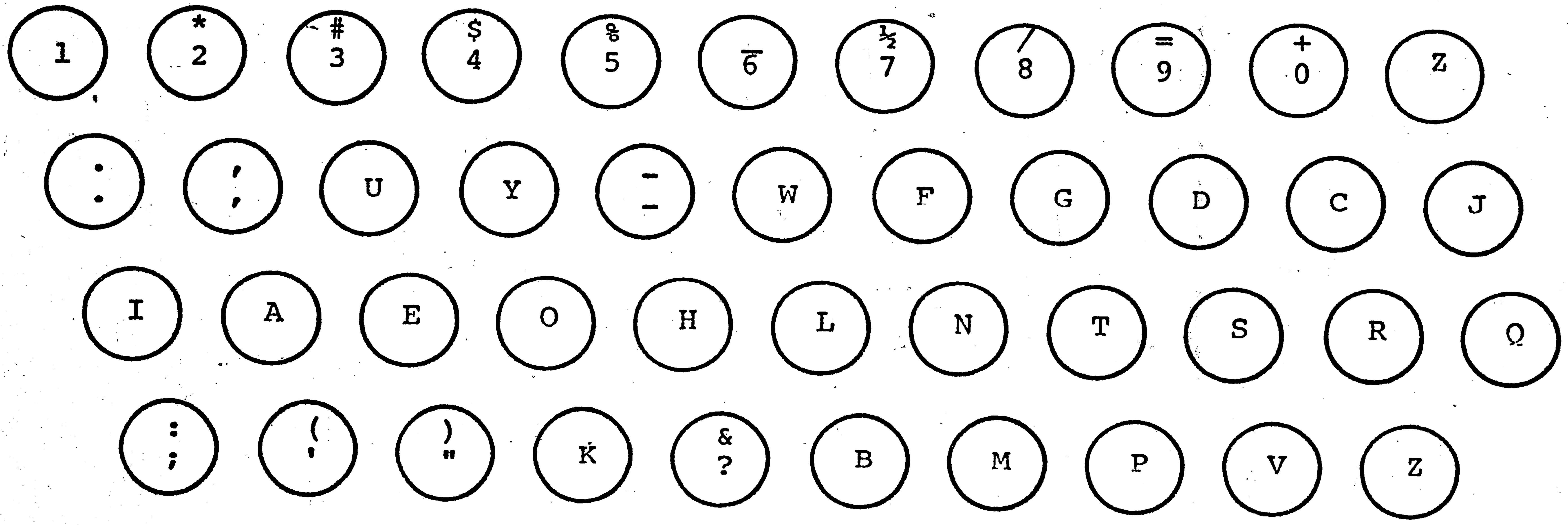
With the exception of Dr. Strong, no one involved in typing education believes these gains are possible in the maximum of nine months the subjects could have had for retraining. (The actual times are not specified in the report. However, the report does state that the QWERTY students began on March 5, 1956 and as the report was released in 1956, nine months is a reasonable maximum.)

No detailed data collected during the course of the study has ever been made available by either the GSA or Dr. Strong. In fact, Dr. Strong had indicated that all records from the study have been destroyed.

> Finally, Dr. Strong's insistence that he was an impartial conductor of the experiment must be subjected to scrutiny. On September 13, 1949, Dr. Strong wrote, "...I have developed a great deal of material on how to get increased production on the part of typists on the standard keyboard. Consequently, I am not in favor of purchasing new keyboards and retraining typists on the new keyboard when we can easily get increased production on our present keyboard. I strongly feel that the present keyboard has not been fully exploited, and I am out to exploit it to its very utmost in opposition to the change to new keyboards." [8]

Could Dr. Strong have been the impartial reporter he claims to have been seven years later? This author doesn't know but surely, sufficient doubt about the validity of the GSA Study exists in order to reject it as relevant to the subject of the DSK.

Another highly regarded keyboard is the Minimotion, developed by R.T. Griffith in 1949. (See Figure 1.4) [9] Mr. Griffith attempted to sequentially optimize a set of ordered objectives and is quite explicit about the procedures which he used. As no evidence can be found that the Minimotion keyboard has ever been tested, it is presented here as another example of the attempts to improve upon the QWERTY.



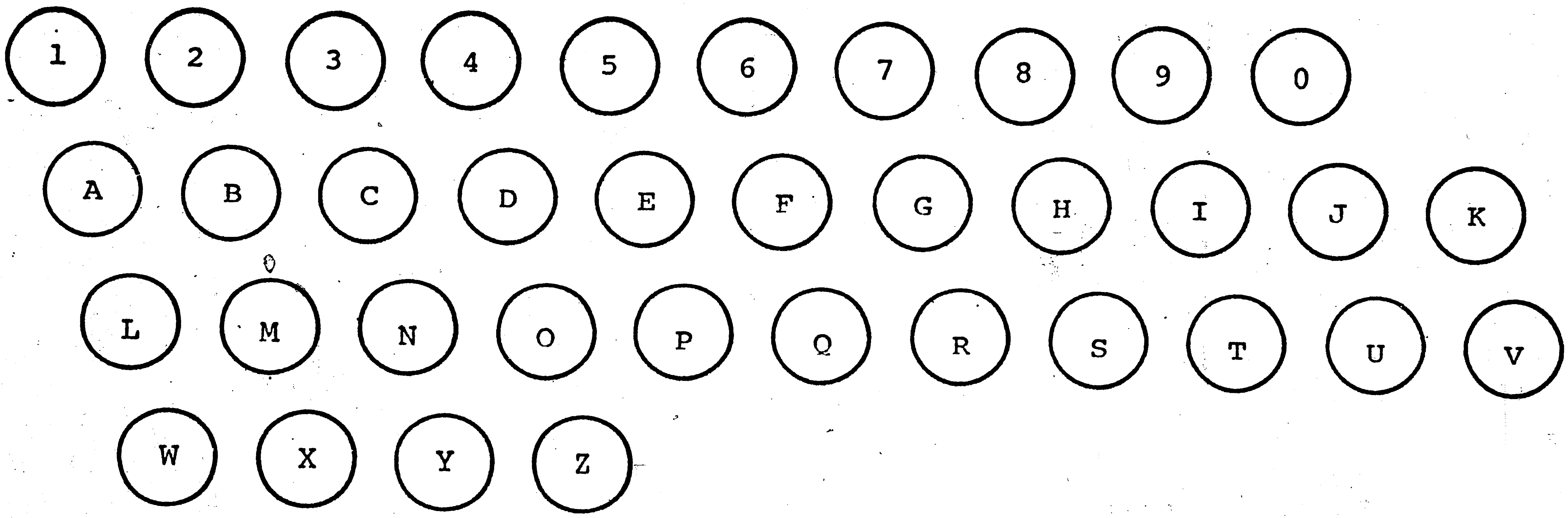
GRIFFITH MINIMOTION KEYBOARD

FIGURE 1.4

A good example of the unexpected pitfalls in keyboard design is the alphabetic keyboard. Intended for a stockmarket query system, an alphabetic layout (See Figure 1.5) was chosen because the system users had little, or no keyboard experience. What would seem to be a logical and sound choice has been shown by Hirsch to be a poorer choice than the QWERTY for hunt-and-peck typists. [10] Evidently, the operator tends to determine the position of the character in the alphabet before he attempts to locate it and strike the key. The point is that many, not necessarily obvious factors influence the efficiency of a keyboard and no amount of logic can supercede testing with operators.

In fact, there have been so many attempts to develop improved keyboards that the U.S. Patent Office has established a separate sub-group classification for typewriter keyboard patents. This means that several hundred patents have been granted for different keyboards--each with its own set of objectives and claims.

The existence of eight to ten million alphanumeric keyboards in use today is often cited as a reason for not adopting a radically different keyboard. On the other hand, several factors are beginning to assume an important role in exploding that ten million to well over a hundred million. Should this tremendous increase in the number of keyboard



HIRSCH ALPHABETIC KEYBOARD

FIGURE 1.5

users occur, the failure to adopt an optimal keyboard would be an insane mistake. The factors are simply listed below and any speculation about their importance is left to the reader.

> The growing utilization of keyboards in education--both in computer-aided instruction and as a motivator in elementary education.
>
> The increasing application of on-line data collection networks in industry and government is requiring ever increasing numbers of people to use keyboards--even if they have had no formal keyboard training.
>
> The increasing probability that homes of the future will contain a data terminal connected to some form of time-sharing computer network. When it will happen depends upon whose crystal ball you gaze into, but no one is predicting that it won't happen.

If all three factors achieve the degree of utilization predicted by experts in the respective fields, there will eventually be few, if any, persons left who will not utilize a keyboard in their normal, day-to-day lives. Does it make any sense to inflict upon that many people, a keyboard designed in 1873 with the goal of slowing down two-finger typists?

CHAPTER TWO

Mathematical Programming Models

In the first chapter, the case for an improved keyboard was established. Apparently, Dr. Dvorak has come the closest to the optimum keyboard. In his book, TYPEWRITING BEHAVIOR, Dr. Dvorak lists the defects of the QWERTY and defines the design goals of the DSK as the minimization or elimination of those defects. The defects he discovered were:

1. The QWERTY keyboard overloads the commonly weaker left hand.
2. The QWERTY keyboard overloads certain finger and does not assign enough work to others.
3. Too little typing is done on the home row and conversely, to much work is done on the upper and lower rows.
4. The QWERTY requires the fingers to execute too many "hurdles" over the home row and too many "reaches" from the home row.
5. Too few words can be typed exclusively on the home row, requiring a reach or hurdle for almost every word typed.
6. The QWERTY keyboard requires that too many words be typed using only one hand.

A study by Provins and Glencross supports Dvorak's findings and explains the significantly better learning times observed in the Tacoma Experiment. [11] In a study of the dexterity levels of the hands of trained typists and subjects with no typing experience when performing typing-like tasks, the following results were observed.

For letter and tapping exercises, the right hand of non-typists was significantly faster than their left while the trained subjects showed either no difference or a significant difference in favor of the left hand.

For word exercises, no differences were recorded between sides for the untrained subjects whereas the trained subjects showed a highly significant difference in favor of the left hand.

No significant differences in errors were recorded between hands for either group.

The implication is, of course, that the reason students find typing so difficult to master is that the emphasis the QWERTY keyboard places on the left hand forces them to acquire an abnormal level of motor skills with their non-preferred hand.

An examination of the defects found by Dvorak and why they are defects yields an interesting conclusion. Item 1 is a defect because the left hand is less efficient for a right-handed person. Item 2 is a defect because of the tiring of overloaded fingers and the overloading of fingers with low dexterity while more dexterous fingers go underloaded. Items 3, 4, and 5 are undesirable because reaching from the home row takes additional time. Item 6 is undesirable because it is the opposite of alternate-hand stroking, the most efficient stroking pattern available. In other words, all of the QWERTY defects are defects because high frequency letters and combinations of letters must be stroked with high cost (or low dexterity) motions or combinations of motions.

Unfortunately, Dr. Dvorak has not disclosed the procedure he used to develop the DSK and one can only speculate. However, as the DSK was developed in 1930-1931, the use of a mathematical programming technique is probably precluded, which implies that the DSK is suboptimal.

Therefore, this author has attempted to develop a global optimization technique which is based upon the frequencies of digraphs (two-letter combinations) used in the English language and a factor which is proportional to the difficulty of stroking pairs of keys. The digraph was chosen as it is the smallest unit capable of describing the sequential interrelationships of characters which make up ordinary text.

The factors relating to the difficulty of stroking the pairs of keys might logically be the average time it takes to perform the two-stroke sequence during the typing of normal text. The design of the not insignificant human factors experiment which would be required to collect this data is beyond the scope of this work and will not be addressed.

In this country today, the standard typewriter keyboard has 44 keys arranged in four rows--the top row with 12 keys, the middle two rows with 11 keys each and the bottom row with 10 keys. Typewriters used in other countries and keyboards designed for telecommunications functions usually have 46 or more keys and while these keyboards will not be considered explicitly, the procedures could be easily extended to apply.

The assignment of 44 symbols to 44 key positions (ignoring upper-case graphics) is a problem with 44! or 2.66×10^{54} different possible keyboard arrangements. In order to reduce the problem to a more manageable size, although still one of immense numbers of possibilities, it is not too unreasonable to consider only the hardcore touch-typing region--three rows of ten keys each. This allows for the assignment of the alphabet and four punctuation marks. The numerics can logically be separated as there is little or no significant interaction with the alphabetic characters. The other four positions not being considered are felt to have such low usage as to have little or no effect on the results. Of course, if one were adamant about considering all symbols and positions, the procedures described herein could be easily modified to accomodate them. Figure 2.1

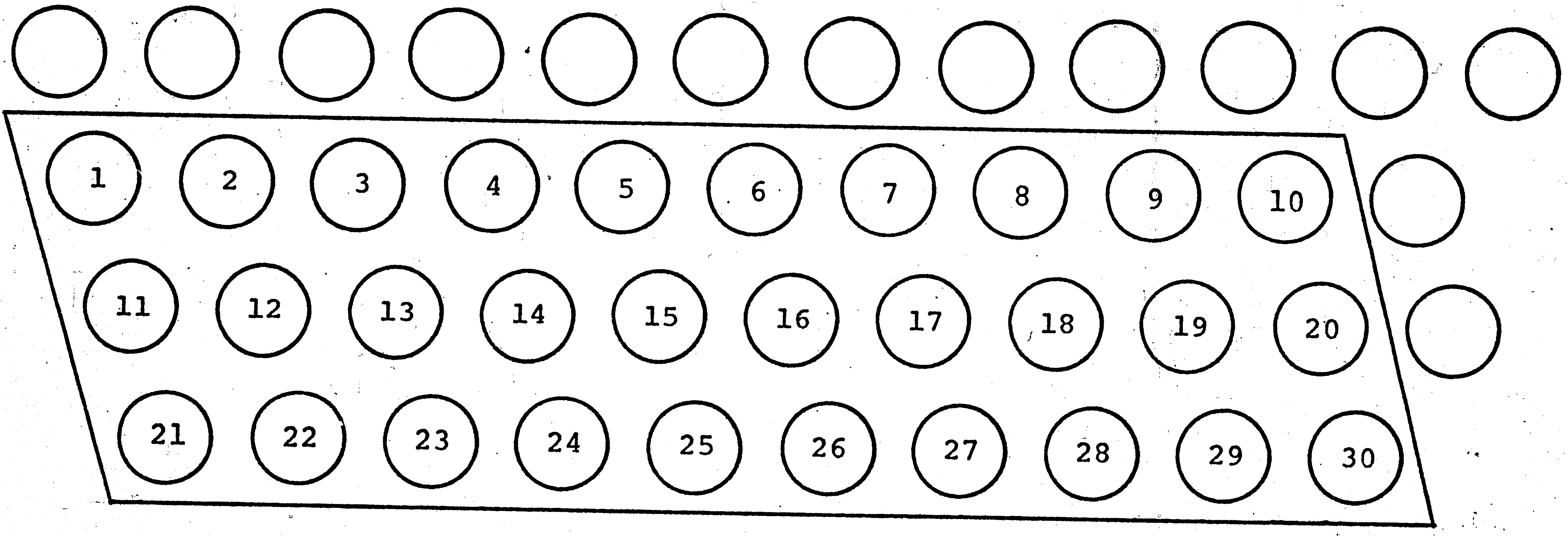
shows the region of the keyboard being considered.

The space bar and shift keys have also been eliminated from consideration--the space bar as it is insensitive to keyboard arrangement and the shift keys due to lack of sufficient data. In a more comprehensive effort, with adequate data, the shift keys should be included as the little fingers, which activate them, do not have equal levels of dexterity.

The digraph frequency data used were taken from a Master's Thesis done by C.E. Rowe in 1930. [12] It should be emphasized that this data is inadequate for a comprehensive effort to design an optimal keyboard, but as the goal of this paper is to develop the technique for designing such a keyboard, it is not felt to be inadequate.

The relative stroking times for each two-key combination are even more arbitrary than the digraph frequencies. Unfortunately, the only published reference which included any stroking times was a paper published by J.E. Coover in 1923. [13] His paper included the stroking times for seven two-key combinations as achieved by an expert typist working at a rate of 130 words per minute.

In order to have somewhat representative data to fill the 30 by 30 stroking time matrix, the author had to devise,



HARDCORE TOUCH-TYPING AREA
AND POSITION NOTATION

FIGURE 2.1

arbitrarily, a scheme for extrapolating those seven values into 900 values. By itself, the Coover data was inadequate to describe the relationships of the dexterity levels of the individual fingers.

Fortunately, Dr. Dvorak included in TYPEWRITING BEHAVIOR, the tapping rates for each finger as reported by Reimer. [14] This data, with the Coover data, are shown in Figure 2.2 and is sufficient for a model of the form:

$$S(i,j) = k(T(i) + T(j) + C(1)) \qquad (2.1)$$

In this arbitrarily chosen model, $T(i)$ and $T(j)$ are the relative tapping times for finger i and finger j and are referenced to the right index finger which is normalized to one and were derived from the Reimer data.

The Coover data was used to determine the values of k, the factor which converts the sum from a relative number to seconds, and the $C(1)$, the extrapolation factors for the various classes of motions.

Through the years, a rather descriptive set of names has evolved for the classes of motions and are defined below.

TAP - Same hand, same finger, same key.

REACH - Same hand, same finger, one stroke on the home row and the other off the home row.

FINGER	L4	L3	L2	L1	R1	R2	R3	R4
STROKES/15 SECS.	48	57	66	63	70	69	62	56
RELATIVE TIMES	1.46	1.23	1.11	1.06	1.00	1.02	1.13	1.25

REIMER TAPPING RATE DATA

L	DIGRAPH	CLASS	FINGERS		POSITIONS		TIME (SECS)	C(L)
1	D-K	ALTERNATE	L2	R2	13	18	.082	-1.62
2	J-J	TAP	R1	R1	17	17	.142	-1.24
3	J-K	TRILL	R1	R2	17	18	.104	-1.45
4	J-L	ROCK	R1	R3	17	19	.098	-1.60
5	U-M	HURDLE	R1	R1	7	27	.171	-1.08
6	J-M	REACH	R1	R1	17	27	.161	-1.13
3	L-;	TRILL	R3	R4	19	20	.173	-1.45

COOVER STROKING TIME DATA

FIGURE 2.2

HURDLE - Same hand, same finger, one stroke above the home row and the other below the home row.

TRILL - Same hand, adjacent fingers, same row.

ROCK - Same hand, fingers separated by one or two positions, same row.

ALTERNATE HAND- One stroke with one hand and the other with the opposite hand.

An examination of Figure 2.2 shows that there is only one class of motion with two entries--the trill. Substituting these values into Equation 2.1 and solving for k and C(3), we get k equal to .186 and C(3) equal to -1.45. Knowing k, we can solve for the remaining values of C(1), which are also shown in Figure 2.2.

An examination of the stroking time matrix, S(i,j), indicates that it is convenient to subdivide it into nine sub-matrices as shown in Figure 2.3. The center matrix, A(2,2), represents those combinations performed entirely on the home row and it should be clear that all of its elements can be calculated with Equation 2.1. Likewise, A(1,1) and A(3,3) represent those combinations performed entirely on the top and bottom rows, respectively. Unfortunately, the Coover data doesn't yield a clue as to what additional costs exist when typing a digraph on the top or bottom row. It was therefore, decided to arbitrarily let their entries be a scalar multiple of A(2,2) and thus:

$$A(1,1) = 1.05 \times A(1,1) \qquad (2.2)$$

$$A(3,3) = 1.10 \times A(3,3) \qquad (2.3)$$

It should be clear that the hurdle and reach classes of motion yield only the major diagonals of the remaining sub-matrices, $A(1,2)$, $A(1,3)$, $A(2,1)$, $A(2,3)$, $A(3,1)$, and $A(3,2)$. The values for the major diagonals are easily calculated using Equation 2.1. The Coover data doesn't explicitly define the remaining values which have not been calculated. However, the data does include three different classes which all involve the same finger--the tap, reach, and hurdle, from which we can extract additive constants which may or may not accurately represent the relationships between rows. At least this procedure is in the right direction if not the right magnitude.

REACH DOWN $S(17,27) - S(17,17) = .161 - .142 = .019$ secs.

HURDLE $S(7,27) - S(17,17) = .171 - .142 = .029$ secs.

REACH UP $.029 - .019 = .010$ secs.

With these constants, the remainder of the stroking time matrix is calculated as shown below. Excepting the major diagonals,

$$A(3,1) = A(2,2) + .029$$

$$A(3,2) = A(2,2) + .019$$

$$A(2,1) = A(2,2) + .010$$

and

$$A(1,3) = A(3,1)$$

$$A(2,3) = A(3,2)$$

$$A(1,2) = A(2,1)$$

It should be stressed that we went through the above machinations to get data which would be somewhat representative of what we would find from a human factors experiment designed to gather this data. Any results based upon this data cannot be accurate and are presented only as an example of what might be found if adequate data were used.

The data elements described in detail above allow us to evaluate any keyboard which meets the constraints discussed earlier. If we let

S(i,j) = Mean time to stroke keys i and j

F(k,l) = Relative frequency of digraph kl

$$X(i,k) = X(j,l) = \begin{cases} 1 & \text{When character } k(l) \text{ is assigned to position } i(j) \\ 0 & \text{Otherwise} \end{cases}$$

then the figure of merit or objective function for a given keyboard layout can be defined as

$$X = \sum_{i=1}^{30} \sum_{j=1}^{30} \sum_{k=1}^{26} \sum_{l=1}^{26} S(i,j)\ F(k,l)\ X(i,k)\ X(j,l) \qquad 2.4$$

It should be clear, by this time, that a zero-one programming model has been formulated. The constraints on the model are simple--each character can be assigned to only one position and each position can have only one character assigned to it. The constraint equations must, therefore, be

$$\sum_{i=1}^{30} X(i,k) = 1 \qquad k = 1,2,\ldots,26 \qquad (2.5)$$

$$\sum_{k=1}^{26} X(i,k) = 1 \qquad i = 1,2,\ldots,30 \qquad (2.6)$$

Unfortunately, while the model formulated above is a simple zero-one model, the objective function is not linear and must be transformed to a linear model. If we let

$$X(i,j,k,l) = X(i,k)\ X(j,l) \qquad (2.7)$$

the following constraints must be added to force X(i,j,k,l) to have the proper characteristics. 15

$$X(i,k) + X(j,l) - X(i,j,k,l) \leq 1 \qquad (2.8)$$

$$-X(i,k) - X(j,l) + X(i,j,k,l) \leq 0 \qquad (2.9)$$

Equation 2.8 insures that when both X(i,k) and X(j,l) are equal to one, then X(i,j,k,l) must also be equal to one. Equation 2.9 forces X(i,j,k,l) to zero when either, or both, X(i,k) or X(j,l) is equal to zero.

Unfortunately, the model has grown from a relatively small one with 780 variables and 56 constraint equations to a monster with 608,400 variables and 1,216,856 constraint equations--hardly a model which is solvable on any computer extant today.

As the more obvious approach has failed to yield a practical model, a search was begun for another model which could be solved.

Actually, a more obvious approach was considered first. This problem is very similar to the classical assignment problem. Unfortunately, the objective function is neither linear nor monotonic and thus, the techniques which have been developed to solve the assignment problem do not apply.

If one examines the QWERTY defects listed earlier it isn't too difficult to conceive of a linear programming model if one is willing to forego global optimization. As global optimization is our primary goal, L.P. was discarded as a possible solution technique. [16]

Branch-and-bound techniques don't apply for there appears to be no way to calculate reasonable bounds due to the interaction of assignments and the lack of monotonicity. Dynamic programming models suffer from the same problem. [17]

Enumeration was briefly considered but the number of possible keyboards is 30! / 4! (The 4! term is present due to the four unused positions.) or 1.1×10^{31}. Far too many to be examined by any computer available.

CHAPTER THREE

Random Keyboard Generation

As we saw in Chapter Two, classical optimization techniques simply don't apply or are too large to be solved using today's machines. A more pragmatic attitude is therefore necessary if any solution is to be found at all. The goal of finding the optimal keyboard must be modified to finding a near optimal keyboard. The goal of global optimization will be unchanged as long as practical.

It was decided to generate random keyboards and evaluate them according to the objective function defined in the last chapter (Equation 2.2), in order to learn something about the distribution of values of the objective function.

The Law of Large Numbers was used to estimate the required sample size as there was no prior knowledge of the expected distribution. [18] If we let

$f(k)$ = proportion of keyboards lying below $X(1)$

p = probability of occurence of a keyboard with a value below $X(1)$

n = sample size required

e = positive error term which is allowable.

The Law of Large Numbers can then be represented with

$$\text{PROB}\left[\,|f(k) - p| < e\,\right] \geq 1 - p(1 - p)/ne^2 \qquad (3.1)$$

As we are primarily interested in the extreme lower values of the distribution, we arbitrarily let

$$p = .005$$

$$e = .0005$$

$$\text{PROB} = .95$$

Solving Equation 3.1 for n, we learn that the sample size must be greater than 400,000. Therefore, the sample is chosen to be 500,000.

Equation 2.2 in the last chapter is prohibitively inefficient when applied to sample sizes of this magnitude. Therefore, a more efficient technique was devised to evaluate the objective function.

The digraph frequency data used in this effort has values for only 250 digraphs which were significant in the sample. If we define a table which contains the position to which each character is assigned, the evaluation can be reduced from 608,400 iterations to only 250 in the following manner. Let

$$S(i,j) = \text{Stroking time matrix}$$

$$F(k,l) = \begin{cases} l = 1 & \text{First character of digraph} \\ l = 2 & \text{Second character of digraph} \\ l = 3 & \text{The relative frequency of the digraph} \end{cases}$$

$$P(i) = \text{Position that character } i \text{ has been assigned}$$

$$X = \sum_{k=1}^{250} S\left[P(F(k,1), P(F(k,2)) \right] \bullet F(k,3) \qquad (3.2)$$

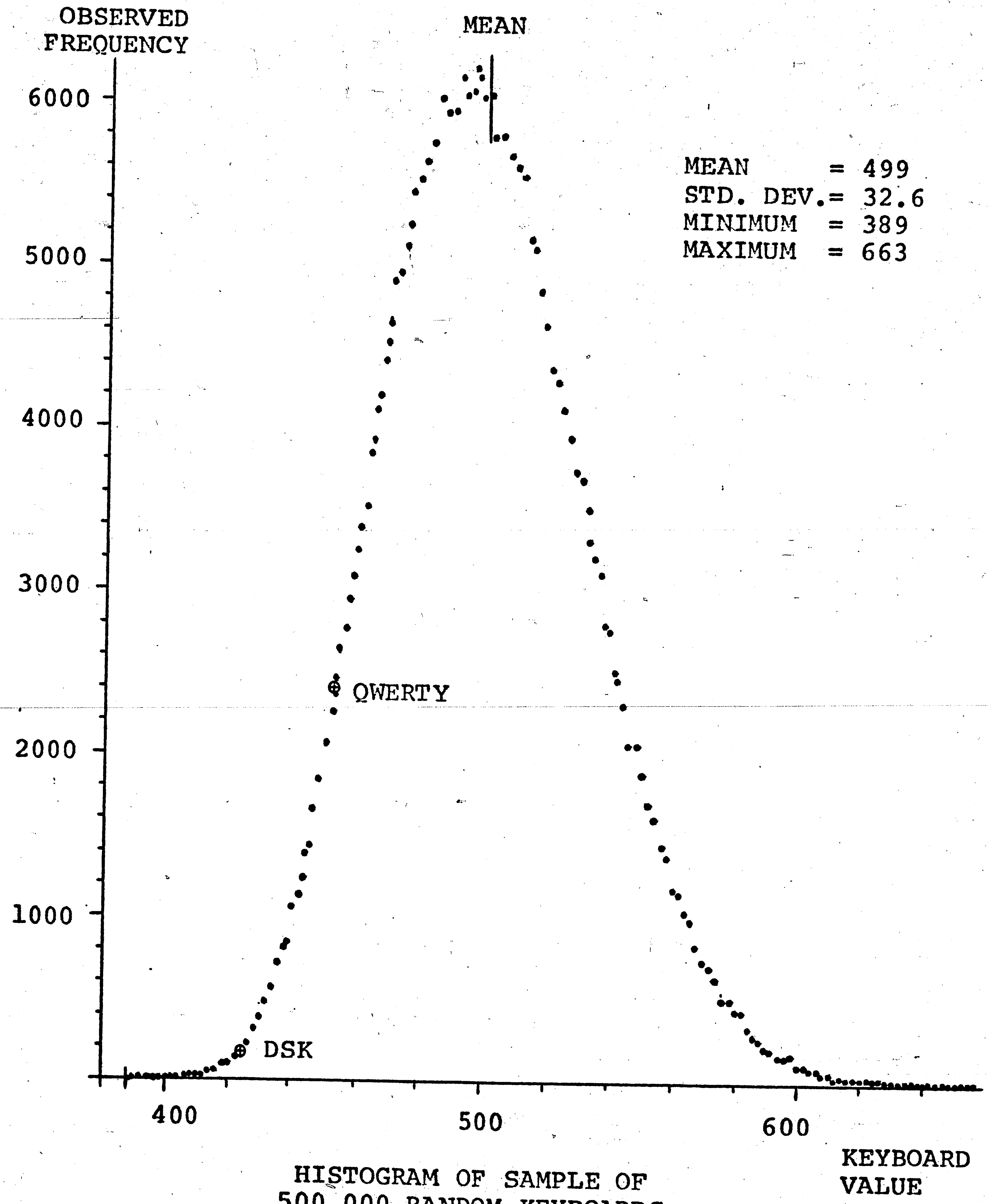
Obviously, this method of evaluating the objective function is much more efficient in terms of computer execution times than Equation 2.2 would be.

500,000 keyboards were then generated and evaluated according to Equation 3.2. Figure 3.1 shows a histogram of the values of the objective function for the sample and also shows the DSK and QWERTY values.

It was then noted that if one was really interested in the extreme values of the distribution, the sample of 500,000 was inadequate. In other words, while simulation has found several keyboards which are "better" than the DSK and QWERTY, the probability of finding a value very near the lower bound of the distribution is so low as to approach zero when compared to the size of the population.

Fitting a curve to the left portion of the distribution was considered but rejected because a continuous model wouldn't help estimate the lower bound of a discrete distribution.

However, the value of the lower bound is important, even if we are unable to find the corresponding keyboard or



HISTOGRAM OF SAMPLE OF
500,000 RANDOM KEYBOARDS

FIGURE 3.1

keyboards, as it can be used to determine the validity of heuristic search techniques and also aid in determining when further searching would be uneconomical.

As the theory of Extreme Value Statistics has received attention in the last few years, an attempt was made to determine if the theory could be applied to the problem of determining the lower bound of the distribution of keyboard values. [19,20] A cursory examination of the literature seems to indicate that the theory has been developed for the case when either the distribution is known or at least is known to be symmetrical. As we don't know the distribution, and Figure 3.1 indicates that the distribution is definitely not symmetrical, it would seem that Extreme Value Statistics offer no solution to our problem. Of course, one could attempt to extend the theory to cover this situation but that would seem to be a major effort in itself.

CHAPTER FOUR

Development of Branching Algorithm

In Chapter Three, simulation yielded several keyboards which were better than the DSK and QWERTY. The shape of the histogram shown in Figure 3.1 implies that even better keyboards exist. However, we have been unable to predict or calculate what the lower bound of the distribution would be.

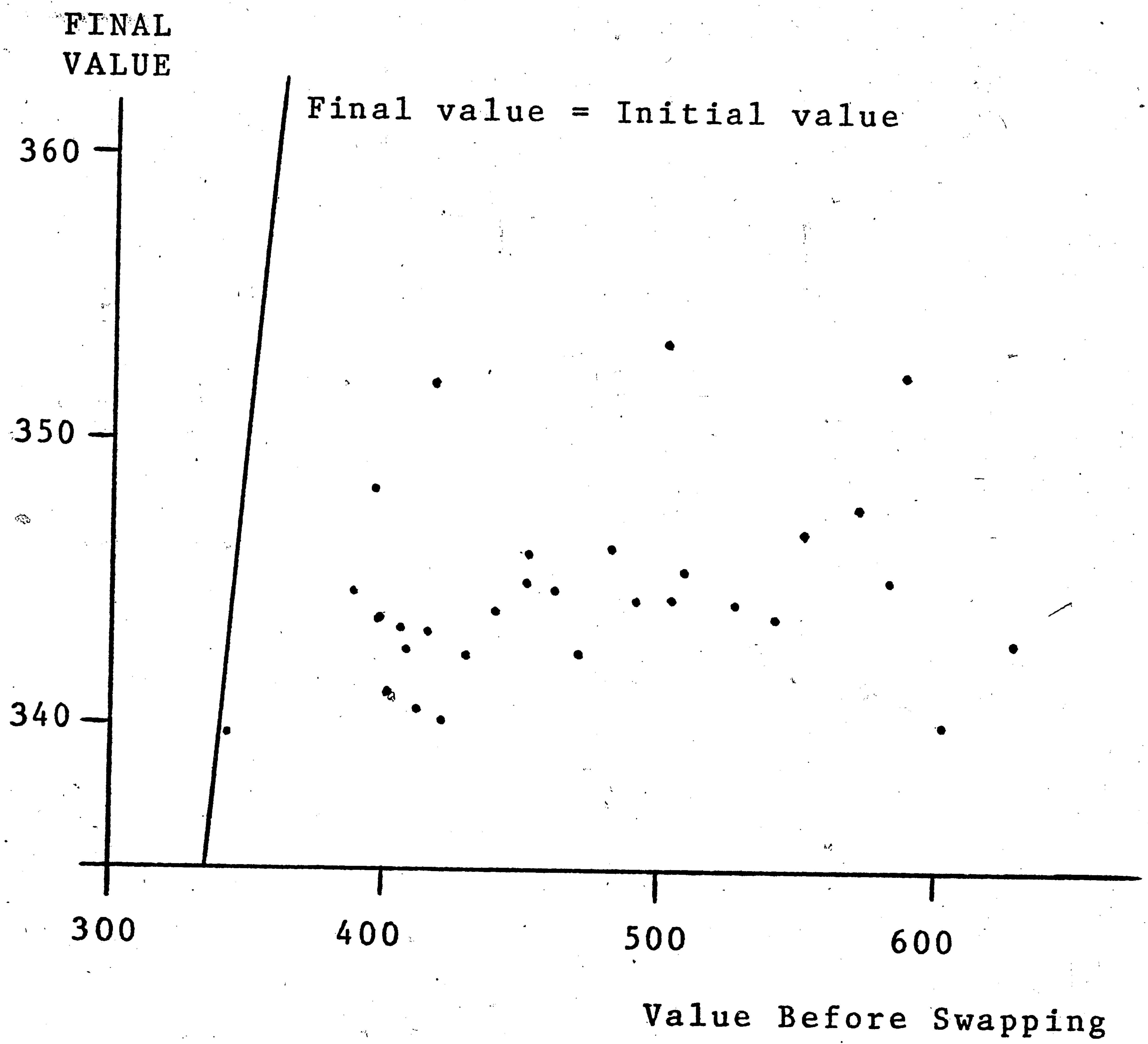
We can, however, compute the absolute lowest bound (ignoring feasibility) which is possible with the data being used. One simply selects the lowest 250 entries in the stroking time matrix and rank-orders them. Similarly, the digraph frequencies are ranked. Multiply the highest digraph frequency by the lowest stroking time, continue the process until all 250 products have been computed, and then sum the products. This value, for the data being used, is 288. While this bound has no statistical value, it can be used to lend credence to the results of a branching algorithm.

When the decision was made to apply some type of branching algorithm to the problem of finding a near-optimal keyboard, it was decided to initially use the simplest approach in order to eliminate the coding problems inherent in the complex file structures necessary to implement more sophisticated branching algorithms. The branching technique chosen was to start with an initial layout, evaluate all possible interchanges of pairs of assignments, choosing the best as the initial layout for the next iteration. The process continues until no further improvement is possible.

A program was written to implement the algorithim and the best keyboard found during the simulation was entered. The routine halted after reaching a keyboard with a value of 344.

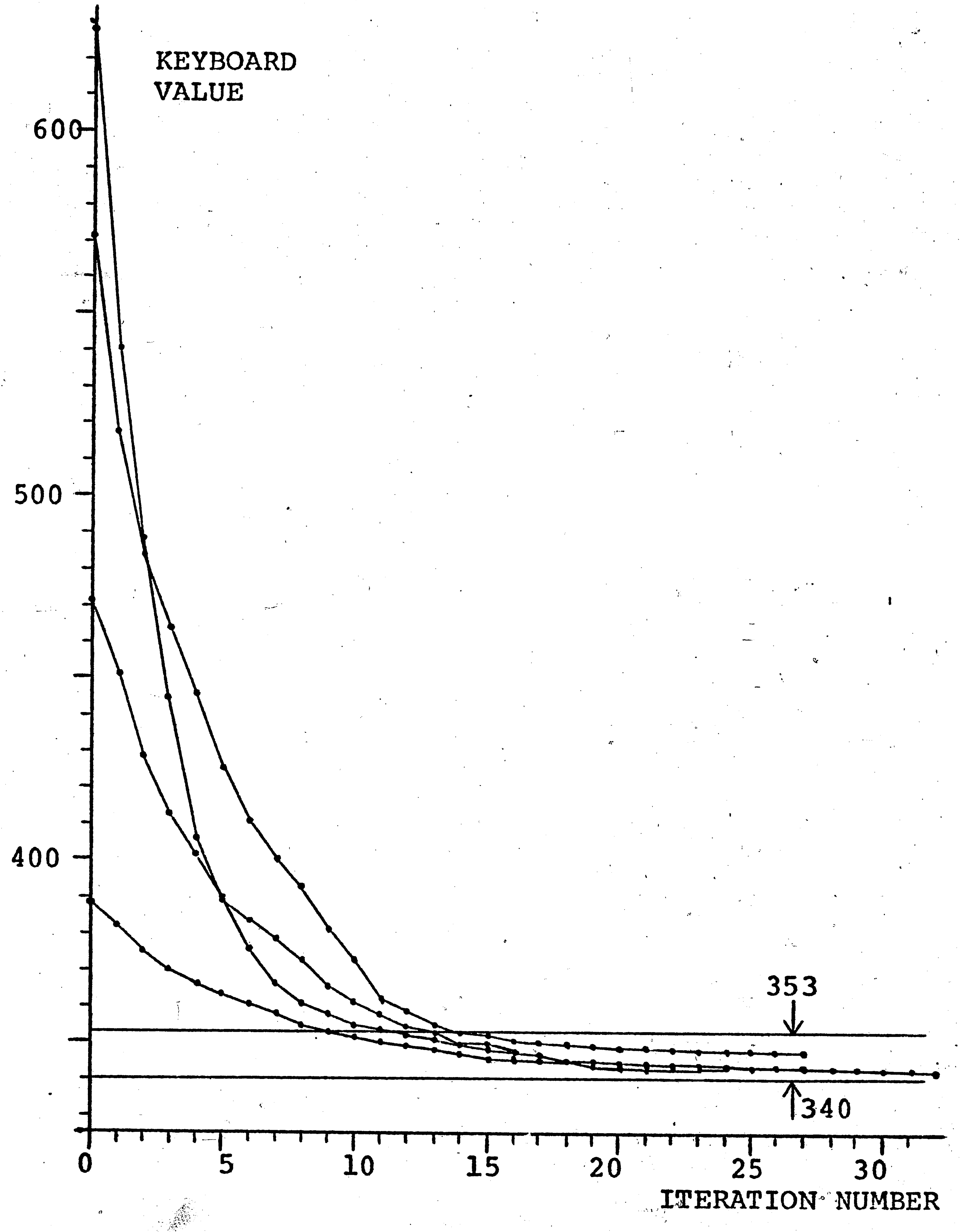
Next, a small sample of keyboards was entered and the swapping seemed to be equally successful, regardless of the value of the input keyboard. So far, 29 different keyboards with original values ranging throughout the original distribution have been applied to the algorithm. (See Figure 4.1 for scatter plot) The best keyboard found has a value of 340 and the worst, after the algorithm, has a value of 353. There is no significant correlation between the original values and the final values.

Next, in an attempt to learn more about how the algorithm progresses, the algorithm was modified to print out the value of the best keyboard at each iteration. Several keyboards were then run with the modified algorithm. The results for some of the trials are shown in Figure 4.2. Three parameter, exponential models were fitted to a few of the curves with excellent results, using non-linear regression techniques. [21] The results for two of these



INITIAL-vs-FINAL VALUE
SCATTER DIAGRAM OF
SWAPPING ALGORITHM PERFORMANCE

FIGURE 4.1



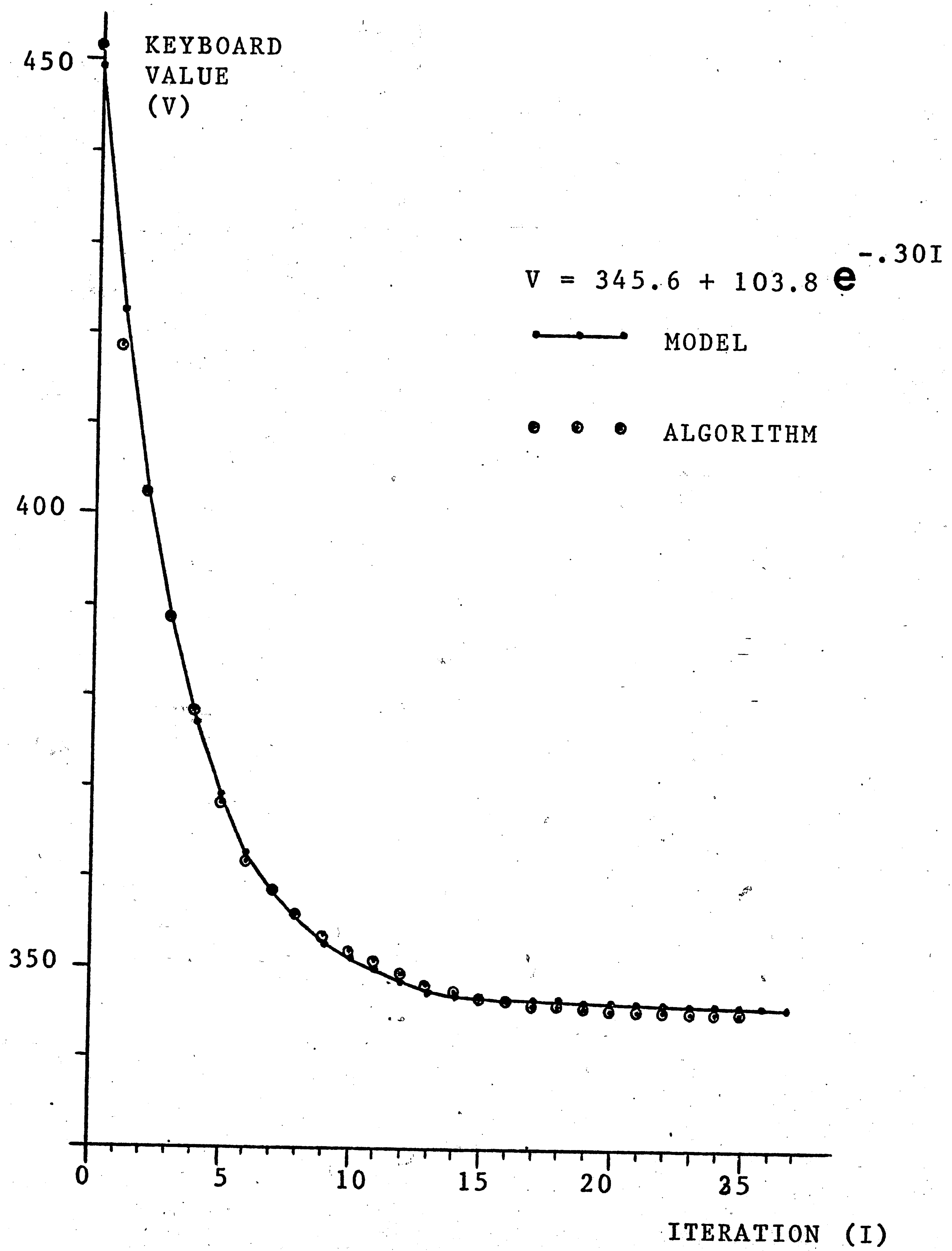
KEYBOARD-VS-ITERATION

FIGURE 4.2

non-linear regression models are shown in Figures 4.3 and 4.4.

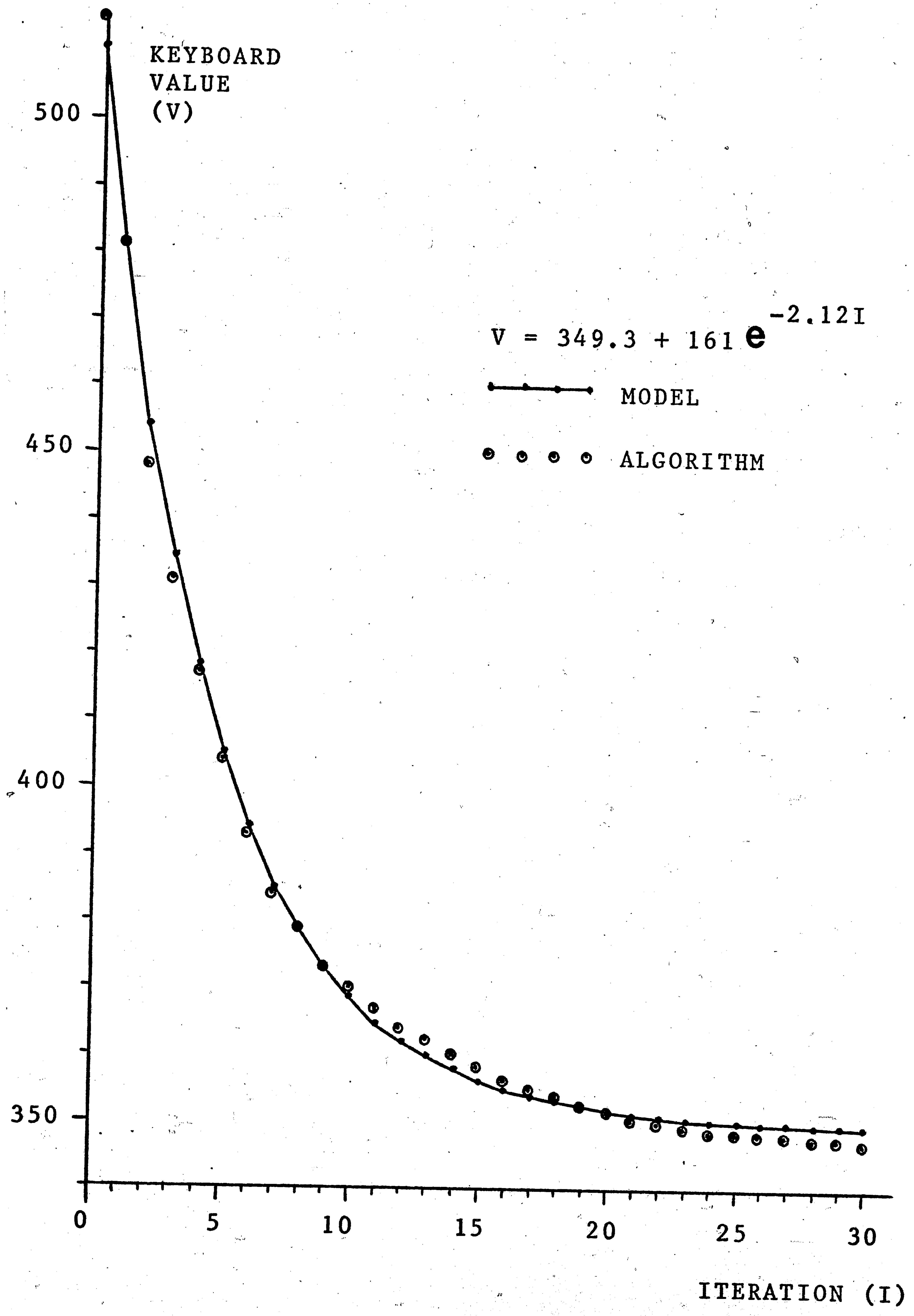
At this point, the apparently overwhelming success of the algorithm made it imperative to take a hard look at whether the algorithm was, in fact, converging near the lower bound of the distribution of possible keyboards. Regardless of the value of the keyboard inputted into the algorithm, the algorithm converged to within a relatively narrow band. After considerable thought, it appears that only two conclusions are possible--either the algorithm itself has an inherent limit or the lower bound of the distribution is, in fact, near 340.

Assuming the algorithm is self-limiting, the only apparent mechanism of the limitation could take is that duplicate keyboards are being evaluated by subsequent iterations and each succeeding iteration has fewer new keyboards to evaluate and, hence, a lower probability of finding a better keyboard. But this is possible only if a small number of assignments, say four or five, are being permuted within themselves with all of the other assignments being held fixed for most of the iterations. To check this possibility, the number of characters whose position was changed by the algorithm was counted for each keyboard submitted to the algorithm. It was determined that an



ALGORITHM PERFORMANCE-vs-EXPONENTIAL MODEL

FIGURE 4.3



ALGORITHM PERFORMANCE-vs-EXPONENTIAL MODEL

FIGURE 4.4

average of 26.4 characters changed positions and since the average number of iterations was only 26 it is reasonable to conclude that a mechanism of this sort is not occuring and that the algorithm does not appear to be self-limiting.

As the algorithm doesn't appear to be self-limiting, the only conclusion possible is that the algorithm is, in fact, converging to near the lower bound, which seems to be near 340. As a final check, the seven best keyboards found by the algorithm (those with values less than 343) were "averaged" on their assignments. Figure 4.5 shows the assignments of the seven keyboards, the "average" keyboard, and the final keyboard after the algorithm was applied to the "average" keyboard. That the value of the "average" keyboard after swapping is 339.913 can only support the conclusion that the lower bound of the distribution is about 340. This conclusion is also supported by the fact that the lowest possible bound with this data is 288.

KEYBOARD	1	2	3	4	5	6	7	"AVG"	FINAL
VALUE / POSITION	340.06	340.27	340.74	341.27	342.57	342.81	342.91	341.40	339.91
1	–	–	–	–	–	–	–	–	–
2	B	G	F	F	B	K	K	F	F
3	W	W	C	W	C	G	W	W	W
4	N	N	R	L	T	A	N	N	R
5	L	L	L	S	L	I	L	L	L
6	I	I	O	I	U	L	–	I	I
7	A	A	A	O	A	N	I	A	O
8	D	D	U	U	H	S	D	U	U
9	Y	F	Y	Y	G	D	G	Y	Y
10	–	Z	J	J	J	P	–	J	Q
11	–	–	–	–	–	–	–	–	–
12	C	C	B	C	F	Y	Y	C	C
13	S	S	S	T	S	U	S	S	S
14	T	T	T	T	N	N	E	H	T
15	R	R	N	R	R	O	R	R	N
16	O	O	I	A	I	H	A	O	A
17	E	E	E	E	E	R	E	E	E
18	H	H	H	H	O	T	T	H	H
19	U	Y	D	D	D	C	C	D	D
20	G	K	G	G	X	B	P	G	G
21	–	–	–	–	Z	Z	Z	–	–
22	X	X	X	Q	Q	–	X	X	X
23	F	B	V	B	V	X	B	B	B
24	M	M	M	M	M	J	M	M	M
25	V	V	W	V	W	–	F	V	V
26	J	Q	Z	Z	–	V	J	Z	Z
27	Q	J	–	X	Y	M	O	–	J
28	P	U	P	P	P	W	U	P	P
29	K	P	K	K	K	F	Y	K	K
30	Z	–	Q	–	–	Q	Q	Q	–

SUMMARY OF "BEST" KEYBOARDS

FIGURE 4.5

CHAPTER FIVE

Sensitivity To Data

The procedures developed in the last chapters are based on two classes of data, the digraph data and the stroking time matrix. An implicit assumption which has been made is that the process is sensitive to the data.

While the digraph frequencies were taken from a thesis written in 1930 and while this data might not be complete enough for a really comprehensive effort to develop an optimal keyboard, the fact remains that the useage of the English language hasn't changed so radically in the last 40 years as to alter the structure of the data significantly.

On the other hand, the same cannot be said about the stroking time data. The extrapolation procedures described in Chapter Two are, to say the least, arbitrary, and if this procedure is to yield an optimal keyboard when used with valid data, it had certainly better be sensitive to changes in the structure of the data.

Unfortunately, there doesn't appear to be a simple metric which could be used to demonstrate or measure this sensitivity, or lack thereof. As the data is used to evaluate keyboards and thus rank them, a reasonable technique might be to take a sample of keyboards, evaluate and rank them before and after modifying the data.

The technique used to extrapolate the stroking time data implies that there are two basic relationships which might affect the outcome of the search for an optimal keyboard. The relationships are the interrelationships among the fingers and the interrelationships among the rows. The simplist method of varying these relationships is the use of multipliers.

Figure 2.3 indicates how this might be implemented in a simple manner. To vary the row relationships, all entries in sub-matrices A(1,1), A(1,2), A(1,3), A(2,1), and A(3,1) are multiplied by the selected multiplier for the upper row. Sub-matrices A(2,1), A(2,2), A(2,3), A(1,2), and A(3,2) are multiplied by the selected home row multiplier, and sub-matrices A(3,1), A(3,2), A(3,3), A(1,3), and A(2,3) are multiplied by the lower row factor. Figure 5.1 presents the results for several sets of multipliers.

Similarly, a set of column (or finger) factors can be chosen and each element in the rows and columns corresponding to each finger, multiplied by the appropiate factor. For the sake of accuracy, the three positions inboard of each index finger were treated as different fingers. Figure 5.2 tabulates the results for several sets of column factors.

The sample of keyboards which was chosen for the sensitivity

		A1	1.0	1.1	1.1	1.2	1.2
		A2	1.0	1.0	1.0	1.0	1.1
		A3	1.1	1.1	1.2	1.2	1.3
KEYBOARD	ORIGINAL VALUES	INITIAL RANK	RANK	RANK	RANK	RANK	RANK
1	389.47	1	1	1	1	2	1
2	395.50	2	4	4	6	6	6
3	397.52	3	2	2	3	3	2
4	400.72	4	3	5	4	4	4
5	405.37	5	7	8	8	8	8
6	409.03	6	5	6	5	5	5
7	412.59	7	9	9	9	9	9
8	415.00	8	8	7	7	7	7
9	418.41	9	11	10	11	11	11
10	420.29	10	10	11	10	10	10
DSK	424.62	11	6	3	2	1	3
QWERTY	453.37	12	12	12	12	12	12
DSK/QWERTY RATIO		.935	.924	.869	.858	.814	.875

ROW FACTOR SENSITIVITIES

FIGURE 5.1

	F1	1.1	1.2	1.3	1.3
	F2	1.0	1.1	1.2	1.2
	F3	1.0	1.0	1.1	1.1
	F4	1.0	1.0	1.0	1.0
	F5	1.0	1.0	1.0	1.1
	F6	1.0	1.0	1.0	1.1
	F7	1.0	1.0	1.0	1.0
	F8	1.0	1.0	1.1	1.1
	F9	1.0	1.1	1.2	1.2
	F10	1.1	1.2	1.3	1.3
KEYBOARD	INITIAL RANK	RANK	RANK	RANK	RANK
1	1	1	1	4	1
2	2	3	2	1	3
3	3	2	3	3	2
4	4	4	4	2	4
5	5	5	5	5	5
6	6	6	7	6	8
7	7	7	6	7	6
8	8	8	8	8	7
9	9	9	9	10	10
10	10	10	10	9	9
DSK	11	11	11	11	11
QWERTY	12	12	12	12	12

FINGER WEIGHT SENSITIVITY

FIGURE 5.2

analysis included eleven keyboards with values evenly distributed in the lower tail, the DSK which fit on the high end of the eleven, and the QWERTY which was included for curiosity's sake. It should be pointed out that since the value of the QWERTY is substantially higher than that of the other keyboards, no inferences should be drawn from its lack of movement in the rankings.

Figure 5.2 indicates that the evaluation technique is insensitive to changes in the relationships of the fingers in the stroking time data. On the other hand, Figure 5.1 indicates that the technique is very sensitive to changes in the row relationships. Note that the DSK ranking ranges from twelfth with the original data, all the way up to first position. Since the tests of the DSK seem to indicate that the DSK's margin of superiority over the QWERTY is greater than Figure 3.1 indicates, the extrapolation from the A(2,2) sub-matrix in Figure 2.3 to the other sub-matrices is probably too conservative.

Figures 5.1 and 5.2 do seem to suggest that this technique for evaluating keyboards is capable of discriminating between "good" and "bad" keyboards. Furthermore, it is reasonable to expect this technique to prove a valid method of measuring the worth of a keyboard, provided accurate data is available.

CHAPTER SIX

Conclusions and Recommendations for Further Work

CONCLUSIONS

In review, a zero-one assignment model was formulated for the optimal assignment of graphics to key positions, based on the digraph frequencies of the usage of the English language and the stroking times for two-key combinations. Unfortunately, the objective function has none of the nice properties required for the solution of the problem. The fact that the objective function requires that all assignments be made before the cost accruing to a single assignment can be computed, eliminates all classical mathematical programming models as possible solution techniques.

Furthermore, the only way this technique could be validated is to collect valid stroking time and digraph data, design a near-optimal keyboard using this data, and perform a comparative test with at least one other keyboard and preferably, several keyboards in order to be able to relate operator performance with the objective function values for the keyboards. At this point it is possible to accept this technique only on the basis of an intuitive feel for the dynamics of keyboard operation.

Simulation was tried and as it yielded several keyboards which were "better" than the DSK and QWERTY, it was somewhat successful. However, the extreme size of the population and correspondingly small probability of randomly finding keyboards near the lower bound with a practical sample size, obviates the need for a more practical approach if a near-optimal keyboard is to be developed.

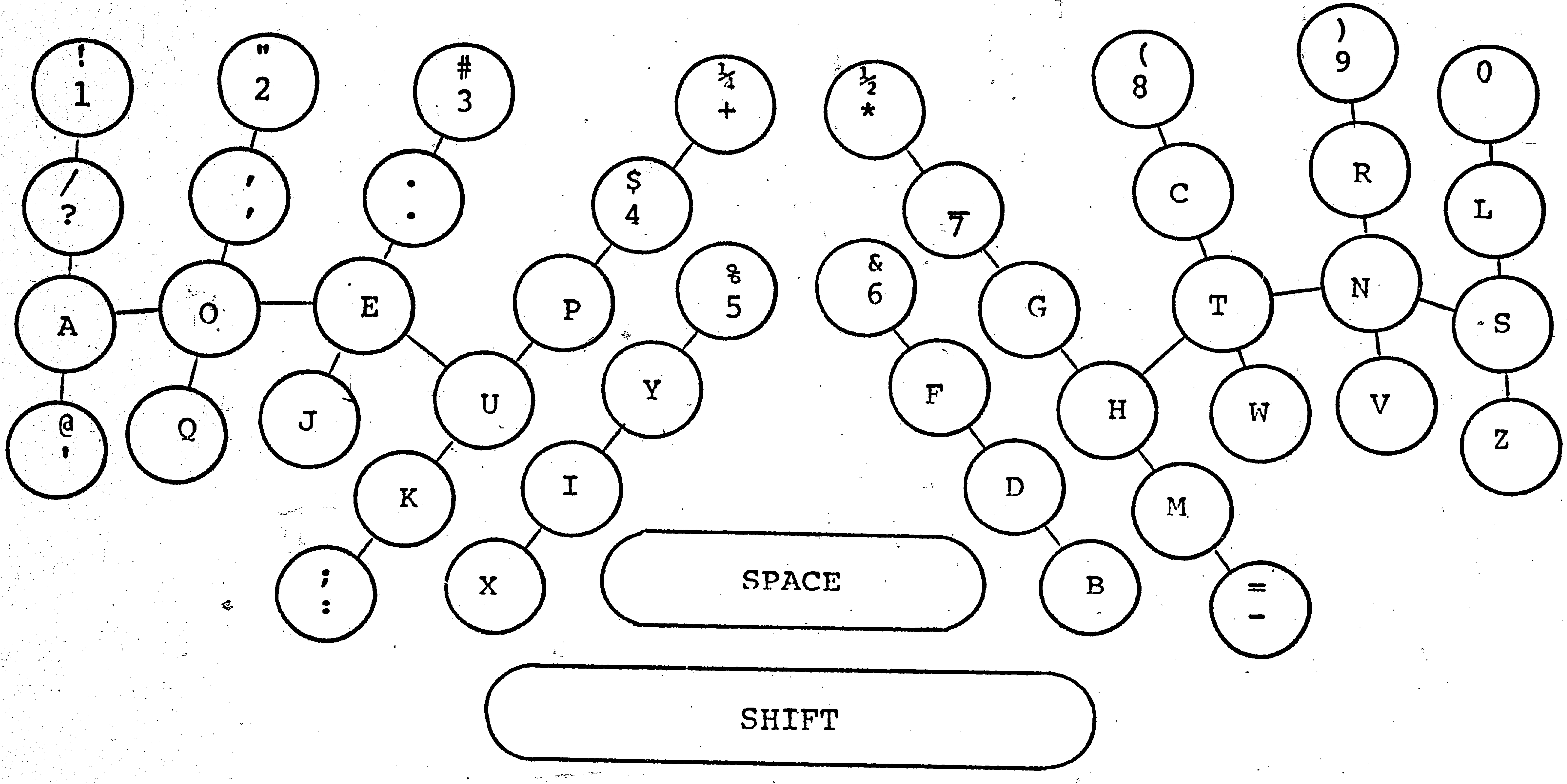
Therefore, a simple swapping algorithm was implemented. The success of the swapping algorithm was surprising. The performance of the algorithm, with the keyboards to which it has been applied, seems to be independent of the original value of the keyboard and the algorithm always converges in an exponential-like manner to within a very narrow range. As there appears to be no reason why the algorithm could be self-limiting and as the lowest possible bound (infeasible) with this data is 288, it is not unreasonable to conclude that the swapping algorithm is converging to near the lower bound and that the bound is approximately 340. Given accurate, representative data, the application of the algorithm to a random sample of keyboards should yield a near-optimal keyboard. Again, even if good data was available, the validation of the keyboard thus developed would require extensive (and expensive) testing with actual keyboard operators.

OPPORTUNITIES FOR FURTHER WORK

The opportunities for further study of keyboards seem endless, and many could be quite rewarding. The most obvious, of course, is the collection of the stroking time data for application with this algorithim.

Another particularly interesting area is the geometry of keyboards--is there any reason, other than historical, for continuing to use the four-row layout used today? Kroemer has experimented with a keyboard which was split into two halves which were hinged to allow them to be angled downwards for operator comfort. However, his tests seem to indicate that the only gains in operator performance realized with this design is a reduction in error rate. [22]

Perhaps a better design would be similar to that shown in Figure 6.1. The keys are arranged to minimize strain in the arms and shoulders resulting from the four-row arrangement. Also, the keys more nearly fit the natural paths the fingers can travel with ease. This arrangement could also be configured anatomically in the vertical direction, somewhat like Kroemer attempted but probably not as radically. The evaluation technique and swapping algorithm would be just as applicable, provided adequate stroking time data was collected.



AN ANATOMIC DVORAK-LIKE KEYBOARD

FIGURE 6.1

Perhaps, the optimum in keyboard design would be a custom designed keyboard for each operator. His keyboard would be tailored to the shape, mobility, and motor skills of his hands and designed for the nature of the material he would be keyboarding. Given appropiate advances in manufacturing and testing technologies, the operator might be fitted for his keyboard as follows. The operator's hands would be measured for size and mobility of his digits. Next, the subject would be tested on an adjustable, test keyboard which was set to his specifications and his personal stroking time data would be collected. His keyboard would then be designed based upon the appropiate set of digraph frequencies for his expected application and his keyboard would be manufactured with a totally automated process. The operator would then have a personalized keyboard which was optimized for his use which could be plugged into a standard keyboard interface on any equipment he might be using. Blue sky? Maybe, but it certainly isn't impossible.

Finally, substantial work needs to be done in the area of keyboard standards. One has to attend only one meeting of the X4-A15 Subcommittee of the American National Standards Institute to appreciate the fact that when keyboard layouts are being discussed (not only within ANSI), everyone immediately becomes a keyboard expert and with a few moments thought can discourse at great length to justify "his" opinions in what would seem to be a logical manner. Unfortunately, the X4-A15 subcommittee is the major influence on keyboard standards and its membership includes no industrial psychologists, no human factors experts, and in fact, no one representing a major user of keyboards outside of the Federal Government. Since the European Countries have finally agreed to a standard family of keyboards, much discussion centers about proposals for the U.S. to develop a standard which is compatible with the Europeans, regardless of any possible human factors impacts on the keyboard users in this country. At best, the situation is deplorable and unfortunately, it isn't likely to change in the forseeable future.

BIBLIOGRAPHY

[1] American National Standards Institute, "Revision, American National Standard Typewriter Keyboard", Doc. X4-A9/33, March 3, 1971, New York

[13] Coover, J.E., "A Method of Teaching Typewriting Based on a Psychological Analysis of Expert Typing", *Proceedings* (National Educational Association, 1923) pp. 561-567

[4] Dvorak, A., Merrick, N.L., Dealey, W.L., and Ford, G.C., *Typewriting Behavior*, New York: American Book Co. 1936

[3] Dvorak, A., "There is a Better Typewriter Keyboard", *National Business Education Quarterly* , Vol. 12, (1943), pp. 51-58

[5] Dvorak, A., *Synergistic Typewriting*, St. Paul, Minn.: Motivational Communications, Preliminary Edition, 1970

[7] General Services Admininstration, *A Comparative Experiment in Simplified Keyboard Retraining and Standard Keyboard Supplementary Training*, Washington, D.C., 1956

[9] Griffith, R.T., "The Minimotion Typewriter Keyboard", *Journal of the Franklin Institute*, Vol. 248 (1948), pp.399-425

[20] Gumbel, E.J., *Statistics of Extremes*, New York: Columbia University Press, 1958

[15] Hillier, F.S. and Lieberman, G.J., *Introduction to Operations Research*, San Francisco: Holden-Day, 1967

[10] Hirsch, R.S., "Effects of Standard versus Alphabetical Keyboards on Typing Performance", *Journal of Applied Psychology*, Vol. 54, No. 6, pp. 484-490

[22] Kroemer, K.H.E., "Human Engineering the Keyboard", *Human Factors*, Vol. 14 (1972), pp. 51-63

[2] Lekberg, C., "The Tyranny of QWERTY", *Saturday Review Science*, Sept. 1972, pp. 37-40

[21] Marquardt, D.W., "An Algorithm for Least Squares Estimation of Nonlinear Parameters", Journal of the Society of Industrial Applied Mathematics, Vol. 11, (1963), pp. 431

[17] McMillan, C., Mathematical Programming: An Introduction to the Design and Application of Optimal Decision Machines, New York: John Wiley and Sons, 1970

[18] Meyer, P.L., Introductory Probability and Statistical Applications, Reading, Mass.: Addison-Wesley, 1965

[8] Parkinson, R., "The Dvorak Simplified Keyboard: Forty Years of Frustration", Computers and Automation, Vol. 21, No. 11, (1972)

[11] Provins, K.A. and Glencross, D.J., "Handwriting, Typewriting, and Handedness", Quarterly Journal of Experimental Psychology, Vol. 20 (1968), pp. 282-289

[14] Reimer, E., "A Revised Keyboard for the Typewriter", unpublished Master's Thesis, New York University, 1929

[12] Rowe, C.E., "Importance of Two, Three, Four, and Five Letter Combinations on the Basis of Frequency in a Word List", unpublished Master's Thesis, University of Pittsburgh, 1930

[19] Sarhan, A.E., and Greenberg, B.G., Contributions to Order Statistics, New York: John Wiley and Sons, 1962

[16] Taha, H.A., Operations Research, An Introduction, New York: MacMillan Company, 1971

[6] U.S. Navy, Division of Shore Establishments and Civilian Personnel, Training Section, "A Practical Experiment in Simplified Keyboard Retraining", Washington, D.C., 1944

VITA

Name: R. F. Nickells, Jr.

Born: July 31, 1943

Graduated:

University of South Carolina - 1969
B.S. - Engineering