DESIGNING OPTIMUM STYLUS KEYBOARD LAYOUT FOR TURKISH LANGUAGE USING A DECOMPOSITION-BASED HEURISTIC

by
Muhammet SOYDAL
B.S., in Industrial Engineering, Istanbul Technical University, 2002

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University
2010

ACKNOWLEDGEMENTS

I would like to thank my thesis supervisor Assoc. Prof. Mahmut Ekşioğlu for his valuable guidance, support, motivation and patience throughout this study. His constructive comments and consistent interest have greatly improved this work.

I would like to thank Assoc. Prof. Necati Aras for his help to this study and taking part in my thesis committee. I also thank to Assoc. Prof. Albert Güveniş for his interest and joining my thesis committee.

I also thank all of my friends, especially Fatih Doğan, Alper Döyen and Soner Özdinç, for their good friendships and assistance. I had very nice hours with them during my study.

Finally I appreciate to my wife, Sevim; I am grateful to her for her continuous support, encouragement, understanding and trust during my life.

I thankfully acknowledge the support of my business friends in T.C. Ziraat Bank during my master.

...dedicated to my endless love, Sevim...

ABSTRACT

DESIGNING OPTIMUM STYLUS KEYBOARD LAYOUT FOR TURKISH LANGUAGE USING A DECOMPOSITION-BASED HEURISTIC

Due to continuous and revolutionary developments in communication technology; the importance and application areas of mobile devices, which can be sampled by PDA’s, hand terminals and 3G cellular phones, are increasing day after day. Owing to physical limitations, generally these devices could only be used by stylus keyboards, which can be typed by stylus, single finger and so on.

In this study; data entry by stylus keyboards is analysed from ergonomics view, also the precautions, which should be taken to keep the stylus keyboard users against the cumulative trauma disorders, are explored. A literature survey for the most important of these precautions, optimization of stylus keyboard layout in other languages, is made in detail. Two stylus keyboard layouts; one for the classical longitudinal shaped as QWERTY and the other for square shaped keyboard which is believed to be more efficient, are designed. For layout optimization, a Quadratic Assignment Problem model is established and solved by a newly developed decomposition based and deterministic heuristic.

New designed two stylus keyboard layouts are tried to be compared with currently in use layouts for Turkish; F and Turkish Q layouts. Comparisons are made with three tecniques; computer aided simulation by randomly selected Turkish e-texts, estimating the typing speeds of expert users for all layouts by using Fitts’ Law - a Human Computer Interaction model - and finally by plotting learning curves of subjects, by the help of a test package coded for this study, following a laboratory procedure. It is concluded that, new generated longitudinal and square layouts have higher typing speeds and average distance per digraph values compared with standard QWERTY and F layouts.

ÖZET

DEKOMPOZİSYON TEMELLİ BİR SEZGİSEL YÖNTEM KULLANILARAK TÜRKÇE İÇİN EN UYGUN SANAL KLAVYE TASARIMI

Sürekli ve hızlı bir şekilde gelişen devrimsel iletişim teknolojileri sebebiyle; PDA, el terminali ve 3G telefonları ile örneklendirebileceğimiz mobil cihazların önemi ve kullanım alanı günden güne artmaktadır. Bu cihazlar, fiziksel kısıtlar sebebiyle özel kalem ve tek parmak benzeri objelerle yazılabilen sanal klavyeler ile kullanılabilmektedir.

Bu çalışmada, sanal klavyelerle veri girişi ergonomi yönünden incelenmiş; sanal klavye kullanıcılarının kümülatif travma hastalıklarından korunması için alınan tedbirler araştırılmış; bu tedbirlerin en önemlisi olan sanal klavye diziliş optimizasyonlarının diğer dillerdeki benzer uygulamaları derlenmiştir. Klasik QWERTY klavye ile aynı şekle sahip olan ve şekilsel olarak daha verimli olacağına inanılan kare şeklinde olmak üzere iki adet yeni sanal klavye dizilişi tasarlanmıştır. Diziliş optimizasyonu için Kuadratik Atama Problem modeli kurulmuş ve bu modelin çözümü için dekompozisyon temelli yeni bir deterministik sezgisel yöntem geliştirilmiştir.

Geliştirilen yeni sanal klavye dizilişleri, halen Türkçe için kullanımda olan F ve Türkçe Q klavye ile mukayese edilmeye çalışılmıştır. Karşılaştırmalar; rassal seçilen elektronik yazılar kullanılarak bilgisayar destekli simülasyon, İnsan-Bilgisayar Etkileşim Modellemelerinden Fitts' Kuralı uygulanması ile tam uzman kullanıcıların erişebileceği maksimum yazma hızlarının hesaplanması ve son olarak bu çalışma için geliştirilen bir test paket programı ortamında deneklerin yazma hızlarındaki gelişimi zamana göre inceleyen labaratuar testleri olmak üzere üç farklı yöntem ile yapılmıştır. Teorik modeller ve labaratuar testleri sonucunda, yeni geliştirilen uzunlamasına ve kare şekilli dizilişlerin halen kullanımda olan standard Q ve F klavyeden yazma hızı ve harf başına katedilen yol açısından daha etkin olduğu sonucuna varılmıştır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

c_{ij}	Cost of assigning facility *i* to location *j*
d_i	One of the digraphs of set Ξ_3^d
$\mathrm{d_{ij}}$	Distance between key *i* and key *j*
$d_{\pi(i),\pi(j)}$	The distances between locations to which *i* and *j* assigned
$dist(d_i)$	The distance between letters of digraph d_i in a layout
f_{d_i}	The frequency of digraph d_i
$f_{m_i}^{opt}$	The ideal load distribution of fingers for a monograph set
f_{m_i}	The load distribution of fingers on a layout for a monograph set
$f_{i,j}$	The flows between facilities *i* and *j*
m_i	One of the elements of the monograph set for a language
p	Assignment vector for QAP
T	Average response time in Hick-Hyman Law
tr	Trace function of QAP
V	Global score for a physical keyboard
$v_{j,ref}$	Division of the score *j* by the respective score of reference keyboard
$\mathrm{W_j}$	Width of the target key *j*

γ_j	Relative weight coefficient for the score j in global score
$\kappa(d_i)$	Big step coefficient for a digraph
Ξ_1^m	Set of all monographs
Ξ_3^d	Set of all digraphs which are typed with the same hand
Ξ_4^d	Set of all digraphs which are typed with the same finger
Ξ_5^d	Set of all digraphs that are typed with same hand but different finger
Ξ_6^d	Set of all digraphs which are produced by using one hand only and whose hit direction is *not* the preferred one

3AP	The three-index assignment problem
ACO	Ant colony optimization
ASK	American simplified keyboard
B&B	Branch and bound algorithm
BiQAP	The biquadratic assignment problem
bps	Bits per second
CHI	Computer-human interaction
cpm	Character per minute
cps	Character per second
DSK	Dvorak simplified keyboard
EA	Evolutionary algorithm
GA	Genetic algorithm

GCK	Gaze-controlled keyboard
GRASP	A Greedy Randomized Adaptive Search Procedure
GRIBB	Great international branch-and-bound search project
HCI	Human-computer interaction
ID	Index of diffuculty in Fitts' Law
IP	Index of performance, equal to *1/b* in Fitts' Law
LAP	Linear assignment problem
MILP	Mixed integer linear programming
MMI	Man-machine interaction
mQAP	The multiobjective QAP
MT	Mean time to point a target in Fitts' Law
OCOB	Optimum connected optimal blocks heuristic
PDA	Personal digital assistant
Q3AP	The quadratic 3-dimensional assignment problem
QAP	Quadratic assignment problem
QBAP	The quadratic bottleneck assignment problem
QSAP	The quadratic semi-assignment problem
QWERTY	Q keyboard layout
SA	Simulated annealing
SFK	Single finger keyboard

TDK	Turkish Language Association
TSP	Travelling salesman problem
VNS	Variable neighbourhood search algorithm
wpm	Word per minute

1. INTRODUCTION

In this chapter, purpose and motivation as well as the social responsibility of this study was presented. For a better understanding of coming chapters, some of the frequently used terms can be described as follows;

- *Stylus*: It is the data entry device for virtual keyboards, such as light pen, pin, single finger...etc.
- *Stylus keyboard*: A stylus keyboard is a system that replaces the hardware keyboard on a computing/mobile device with an on-screen image map. The data are entered by the help of a small pin, usually called a *pen* or *stylus*. It is also sometimes called *onscreen keyboard*, *software keyboard, single finger keyboard*, *soft keyboard, graphical keyboard, on-monitor keyboard, virtual keyboard* or *e-keyboard*.
- *Digraph* and *Monograph*: Digraph refers the sequential combination of consecutive two characters in a text. These characters can be either a letter or a blank. Unique character in a text is named as “monograph”. For instance, the expression “to school” involves 8 digraphs: “to”, “o_”, “_s”,“sc”, “ch”, “ho”, “oo” and “ol” whereas the same expression involves 9 monographs which are “t”, “o” *(3 times)*, “_”, “s”, “c”, “h” and “l”. Here, “_” represents the *blank* character.
- *Distance of a Digraph:* It is the distance taken by the stylus while typing the letters of a digraph using a virtual keyboard layout. For example; distance moved by tip of entry device going from “t” to “e” or “e” to “n” while typing the word “*ten*”.
- *Click typing:* Using a stylus keyboard by clicking the keys with a pointing device, such as using a mouse on a computer screen.
- *Touch typing:* Using a stylus keyboard by directly touching on the key of the board, such as using a stylus keyboard on a touch screen PDA’s.
- *Average Distance per Digraph:* It refers to the distance taken by the stylus to tap a digraph in a text. It can be formulated as:

$$\frac{Total\ distance\ taken\ by\ the\ stylus\ to\ write\ a\ text}{The\ number\ of\ \ digraphs\ in\ that\ text}$$

1.1. Objectives of the Thesis

Up to date, studies on optimization of stylus keyboard layout for some languages, such as English, French, and Spanish, have been made. Due to the importance of stylus keyboard, similar studies should be made for Turkish language. Therefore, the main objectives of this study are to:

i. model the stylus keyboard layout problem as a quadratic assignment problem with the objective of minimizing the distance taken by the stylus,
ii. develop a decomposition-based heuristic to solve this model,
iii. design the optimum longitudinal and square shaped stylus keyboard layouts for Turkish language.

1.2. Text Entry for Mobile Devices

Text entry has never been important as it is today. The need for text entry to mobile devices resulted in numerous different techniques and inventions in recent years, especially after the rapid revolution of GSM and 3G technology. However, most of these newly developed inventions have not been properly researched with either theoretical or empirical human performance studies (Zhai and Kristensson, 2005). One of the vital points to be kept in mind while thinking the text entry systems for mobile device is the multi tasking of the text entry system user. That’s, the users generally enter texts while doing another task, such as walking, driving, drinking and so on. Thus, a new interest area for ergonomists has been the research of effect of these new tools on human body and trying to redesign these tools by applying the ergonomics rules.

1.3. Most Popular Text Entry Systems for Mobile Devices

Among the numerous text entry systems, no one can argue that a specific one has been the most advantageous compared with others. Every text entry system has a number of advantages and disadvantages over another system. That is why; none of these systems have dominated the others. Zhai and Kristensson (2005) briefly describe these new text entry systems as mentioned in following sections.

1.3.1. Classical Physical Keyboards

The physical keyboards were first used in typewriters in 1860's. For this reason, physical keyboards can be sometimes named as typewriter keyboard or 10-finger keyboard in literature.

Zhai and Kristensson (2005) argue that the physical keyboard offers a great number of advantages as a human–computer interface. First of all in addition to speed, a touch typist *(where touch typing means typing without looking at the keyboard)* can focus his or her visual attention on the computer screen, not the typewriter keyboard itself. Interestingly, touch-typing was first applied in the 1880s by L.V. Longley and F. E. McGurrin, many years after the typewriter invention, and was not widely adopted by training schools until about 1915. This means the low attention demand was not a rational design feature, but rather an evolutionary improvement discovered in the process of use.

Although, the typewriter keyboard has been a resilient de facto standard method for text input, it has many weaknesses as a modern interface technology for data entry. First, it takes effort to learn touch-typing skills; and usually it takes hundreds of hours of practice to be proficient. Second, the argument of over specification, to be discussed later on stylus keyboards, can also be made here. The system is unable to take advantage of today's computing power in applying statistical information to reduce the load of input. Third and most importantly, it is not suited for off-desktop computing in its usual size and form, which will be a vital problem for data entry for mobile devices.

Today, the most common physical keyboards use the QWERTY layout, credited to Christopher L. Sholes, Carlos Glidden and Samuel Soule in 1867. Systematic study of human performance in typewriting and optimizing the keyboard layout happened much later, with the best-known example being the Dvorak study and layout for English language.

Some old fashioned mobile devices like cellular phones use small physical keyboards to allow the user enter the data. Generally these small tools are plugged only when it is needed.

1.3.2. Speech Recognition

Speech recognition has been expected to be a compelling alternative to manual typing. Popular text editors started to supply data entry by speech recognition. Despite the progress made in speech recognition technology, however, a recent study showed that the effective speed of text entry by continuous speech recognition was still far lower than that of the keyboard (13.6 vs. 32.5 corrected wpm for transcription and 7.8 vs. 19.0 corrected wpm for composition). Error correction is particularly difficult with speech commands. Furthermore, the study also revealed many human-factors issues that had not been well understood. For example, many users found it ‘harder to talk and think than type and think’ and considered the keyboard to be more ‘natural’ than speech for text entry. For an efficient data entry by speech recognition, the users usually take a training course. During this course, the speech recognition system gets accustomed to the voice of the user. It has been further argued that speech production competes for cognitive/memory resources, which impede the user’s performance. Also, it is important to keep the environment silent for a higher accurateness rating. In short, using speech as a text input method still faces many challenges (Zhai and Kristensson, 2005).

1.3.3. Handwriting Recognition

Zhai et al. (2005) defend that “handwriting is a rather ‘natural’ and fluid mode of text entry, thanks to users’ prior experience from writing on paper. Handwriting recognition technology has made tremendous progress in recent years. The current PDA devices tend to use alphabet character based handwriting recognition, such as Graffiti and Jot. The alphabet used can be either natural or artificially modified for reliable recognition. Edge Write defines an alphabet around the edge of a fixture to help users with motor impairment. The fundamental weakness of (long) handwriting, however, is the limited speed, typically estimated around 15 wpm. For Graffiti and Jot, found between 4.3 and 7.7 wpm performance for new users and 14–18 wpm for more advanced users, although, other informal reports claimed higher peak performance. This speed might be good enough for entering names and phone numbers on a PDA, but too slow for writing a longer text”.

To explain this system technically, an electronic board having a two-dimension coordinate system is used during writing. The user writes the text by the help of a special pen. The electronic components follow the movement of this pen and plot these movements on the coordinate system. This electronic board can either writes exactly what the user writes, just as a photograph, or tries to recognise the letters and transforms the text into electronic text.

1.3.4. Mobile Physical Keyboard

Classical physical keyboards are no more popular nowadays due to their size and weights. So, various ways to reduce the size of physical keyboards have been developed. One is to scale down the size of each key. This method can largely be seen in electronic dictionaries and diaries. Typing on these kind of keyboards is difficult due to their reduced size that prevents 10-finger touch-typing. Another method, which is more popular in mobile devices, is to use the number pads in telephones, whereby each number corresponds to multiple letters. The ambiguity of multiple possible letters is commonly resolved by the number of consecutive taps, or by lexical models. Optical projection keyboards are yet another approach, although a key issue is the lack of tactile feedback of the keys, both vertically (the non-linear resistance of a key) and laterally (key surface features that prevent the finger from drifting away). Furthermore, it requires set-up and a ‘desktop’ space to operate (Zhai and Kristensson, 2005). Writing speeds are tried to be increased by the automatically completion of a word before tapping all letters of that word.

1.3.5. Gesture Input

Gesture input means data entry by the movement of gesture, such as arms or fingers. There have been various continuous-gesture-based text methods. Some of these methods use continuous stylus movement on a radial layout to enter letters. Also, an important portion use continuous mouse movement to pass through traces of letters laid out by a predictive language model. Using such a technique is a novel and intriguing experience, but the primary drawback is that the user has to continuously recognize the dynamically rearranged letters. The visual recognition task may limit the eventual performance of text entry with such a method (Zhai and Kristensson, 2005). Main technical components of the gesture input systems have been the sudden motion sensors. These sensors transform the motion into characters by the help of a three dimensional coordinate system.

1.3.6. Gaze-Controlled Keyboards

When compared with other data entry systems, technology for gaze-controlled keyboard (GCK) is near maturity. Owing to this feature, there are only a few interfaces designed for gaze input effectively.

Figure 1.1. Gaze controlled virtual keyboards for paralysed users

‘‘ERICA’’, which allows input using one’s eyes in the Windows environment, can be given as an example. Ward et al. (2002) showed that a two dimensional eye-tracker can be used to reach an input speed of about 20 wpm for a novice user. All of such applications and demonstrations show the rapidly advancing techniques for numerous applications, especially suitable for disabled users. In a GCK the increased fixation duration, which can be programmed, is the signal to designate a ‘‘key-press’’ (Ward, 2002). To minimize the repetitiveness of eye and head movement, which can give rise to fatigue and even injury, designing a keyboard that minimizes movement time can be very useful. Present day keyboards cannot be easily adopted for SFK or GCK gaze-controlled applications for many different reasons:

i. Due to space considerations, a SFK would not replicate a key unlike on a regular keyboard where keys such as ‘‘Control’’, ‘‘Shift’’, ‘‘Alt’’, etc. are repeated so that they can be used with alternate hands.

ii. Only one key of an SFK can be ‘‘pressed’’ at any one time.

iii. Symmetry and compactness constrain the shape of SFK, whereas a traditional keyboard has been designed with three rows to achieve efficiency using fingers of both hands. (Yanzhi, *et al.*, 2006)

1.3.7. Stylus Keyboards

A stylus keyboard (also sometimes called *onscreen keyboard*, *software keyboard, single finger keyboard, soft keyboard, on-monitor keyboard, virtual keyboard* or *e-keyboard*) is a system that replaces the hardware keyboard on a computing device with an on-screen image map. The data are entered by the help of a small pin, usually called a *pen* or *stylus*. Stylus keyboards are typically used to enable input on a handheld device so that a keyboard doesn't have to be carried with it, and to allow people with disabilities or special needs to use computers. Other devices for tapping letters can be pin, finger, laser, or a special apparatus for abrachias. The displayed keyboard can usually be moved and resized, and generally can allow any input that the hardware version does.

Stylus keyboard arrangement for people using mobile devices such as PDA or for special groups of users such as the disabled or pilots who need to use gaze-controlled execution where actions are predominantly using just a single ‘‘pointer’’ may sometimes be vital (Yanzhi, *et al.*, 2006).

Several different keyboards for single finger keyboard (SFK) entry or for stylus-based text entry such as with the use of pen or stylus have been proposed in the literature. These include the ABC layout, FITALY, OPTI, Metropolis, Hooke, Lewis keyboards and many more (Yanzhi, *et al.*, 2006).

Figure 1.2. Stylus keyboards designed for users

Zhai and Kristensson (2005) describe stylus keyboards by saying “stylus keyboards display letters and numbers on a touch sensitive screen or surface. To input text, the user presses keys with a finger or stylus. Such a keyboard can be scaled to fit computing devices with varying sizes, particularly small handheld devices. One central issue is the layout of the keys in these keyboards. Due to developers’ and users’ existing knowledge, QWERTY tend to be also the default layout of stylus keyboards. However, QWERTY is a poor choice for stylus keyboarding. The polarizing positions of common English digraphs in QWERTY mean that the stylus has to move back and forth more frequently and over greater distances than necessary”.

Additionaly, the key to a good virtual keyboard is exactly opposite to the idea that lise behind QWERTY layout. Because, in stylus keyboards, common digraph letters should be close to each other, as possible, so the hand or stylus does not have to travel much. In other words, the average distance taken by the stylus per one digraph should be minimum. The movement distance concern also points to another problem of QWERTY as a virtual keyboard layout, it is elongated horizontally having three rows and a long space bar, which increases the average stylus movement distances (Zhai and Kristensson, 2005). The human performance effect of relative distances between the letters on a stylus keyboard can be modelled by a simple movement equation which is named as “Fitts’ Law” (Fitts, 1954). This law is accepted as the main movement equation while optimizing the letter positions in stylus keyboards by a great number of ergonomists.

The keyboard represents one of the most popular and effective devices to insert, edit, delete and update long chunk of information. Keyboards used with more than one finger (*which is also called as "n-fingers"*) were firstly introduced more than 130 years ago to support the typists' task. The first keyboard, which is called QWERTY, derived its name from the first keys of the first row of the layout. QWERTY layout is nowadays still used to insert users' data into personal computers. According to Dell'Amico (2009), the most recent proliferation of portable data assistant (PDA), smart phone, cellular phones, chat boxes, and hand terminals have required a strong improvement in the design of input devices, such as the keyboard, to allow the input and management of text, for instance writing of e-mails or messages and allocation of dates in a personal information manager. Especially after the start of the third generation of wireless communication technology (3G), the importance of the text entry systems of portable electronic devices increased.

Typically, keyboards could be used either with many fingers or a single finger (*which is also called as "s-fingers"*). While the n-fingers keyboard has not significantly changed the keys layout, and the major standards have survived for more than a century (despite many alternatives have been proposed), the keyboards for portable systems are a still open design domain. The main reason why the n-finger physical keyboards have not been changed for over a century is the difficulties to shift from the QWERTY layout to a new layout. People cannot easily change their habits which are gained over years, like the keyboard layout they have used. This case is so well known by a large portion of the society that; the difficulty to change a habit of the society is called "QWERTY Effect" in social sciences. Also, the personal computers in the primary schools of many countries are standard physical QWERTY Layout, which force new generations to use QWERTY layout in their rest of life.

The QWERTY keyboard was introduced in 1872 and was primarily designed to slow down the typing speed so as to reduce jamming of mechanical parts. However, such a layout has been controversial in computer applications as it reduces typing speed as well as accuracy.

In other words, a layout (*QWERTY*) which was designed to lower the typing speed has still been used by the great portion of the computer users all over the world.

“However, the situation for the keyboard layouts of mobile devices is different. Many alternatives are available and none of them have definitely dominated the others, in terms of users' acceptance, usage effectiveness and large adoption by the devices developers. Moreover, these portable keyboards are typically used in multitasking conditions (e.g., while walking or driving). Namely, the task of searching and scanning a letter while composing a word could become a problem if it takes too much time. Therefore, to keep these task as short as possible represents a relevant design objective” (Dell’Amico *et al.*, 2009).

The problem of designing new keyboard layouts, especially the physical keyboard layouts, able to improve the typing speed of writing an average message has been widely considered in the literature of the Ergonomics area. The widely used materials to propose new solutions to this problem are the empirical tests with a broad range of users and simple optimization criteria. In spite of these wide considerations, according to Dell’Amico and et.al (2009), very few papers in Operations Research have addressed this optimization problem.

1.4. Comparison of Typing Speed of Physical and Virtual Keyboards

Typing with virtual keyboards is mostly slower than typing with physical keyboards due to the following reasons;

- In physical keyboards, the sectors of the keyboard are distributed by two or more fingers of generally both hands. By this way, during the action of tapping a key, the other finger is directed to the next letter’s key which increases the typing speed. The more fingers involved for typing, the higher speed of typing. However in virtual keyboards, all keys are pressed by the same stylus that it is impossible to get prepared to go the next key before finishing the tapping of the previous key.

- For physical keyboard, users generally tend to “memorize” the place of a letter on the keyboard, which causes typing without looking at the keyboard after enough practise time. It is not the case for virtual keyboards, because even high experienced virtual keyboard users should look at the virtual keyboard screen.
- Physical signs, intervals, even the decrepit keys help to find the place of a key in physical keyboards, but virtual keyboards are placed on smooth monotonous display screens.
- In physical keyboards the force to hit a key is distributed to two hands and to a number of fingers, sometimes to 10 fingers. As a result, the stress of typing is shared by a large number of muscles. However in virtual keyboards, continuous usage of the same muscle groups causes strain and repetitive stress injuries that lowers the typing speed in virtual keyboards.
- Physical keyboards are generally used while sitting, but virtual keyboards are mostly used while walking, standing or driving.
- Physical keyboards are designed mostly for healthy two-handed users, whereas a great portion of virtual keyboards are designed for obstacles and paralysers from whom a high speed of typing cannot be expected.
- Classical physical keyboards are used by generally two or more fingers whereas virtual keyboards are used by a pointer like a finger, stylus, pencil, mouse pointer, etc. to tab the keys during writing.

2. ERGONOMICS BACKGROUND

In this chapter, the necessity of ergonomics to design keyboard and the musculoskeletal problems related with problematic keyboard designs are presented. Also, a brief list of main properties for an ergonomic keyboard is involved.

2.1. Necessity of Ergonomics for Design

"Historically, present day ergonomics evolved from wartime requirements to ensure the ability of operators to control weapon systems or interpret information from newly developed electronic displays and communication systems such as radar. The emphasis was, therefore, primarily on improving the performance of given man–machine-equipment combinations, rather than producing improvements in efficiency measured in terms of value added per man hour" (Beevis and Slade, 2003). That's, ergonomists used to try to change the work conditions to increase the physical and mental satisfaction of the workers. They hardly never thought the overall efficiency of the system.

This attitude is still prevalent today, coupled in some quarters with the idea that ergonomics is some form of welfare service to be provided for the employee by improving his comfort, health or safety. Indeed, although financial savings may be shown to increase from applying ergonomics to job or equipment redesign, Beevis and friends (2003) argue that only a small ratio of organisations and companies establish ergonomics department.

If ergonomists become participators during the design of new tasks, equipment or whole job, rather than in the redesign of existing ones, there will be less opportunity to make before and after cost comparisons. It is worthless to argue that a design could have been made more expensive, or less efficient, although there is admittedly only a shade of difference.

Evaluations made during the design process should select those solutions which will lead to a reasonable payoff between all design factors, including cost and efficiency. If these mentioned points are applied, then evaluations of financial benefits after designing stage will be inessential and rare (Beevis and Slade, 2003).

2.2. Musculoskeletal Problems Related to Office Work

Musculoskeletal problems related to office work, especially computerised office work, are of concern throughout the world (Lingaard and Caple, 2001). These musculoskeletal problems might cause musculoskeletal disorders, if the necessary precautions are not taken. Finally, these musculoskeletal disorders might affect the muscles, ligaments, nerves, joints, tendons and the whole body. A great portion of these musculoskeletal disorders are the results of the task or the conditions of the working conditions. These kinds of disorders come into existence if the body is continuously under affect of the causing effect for a long time, months or even years. One of the widely known results of the musculoskeletal disorders is the Cumulative trauma disorders (CTDs) of upper extremity.

2.2.1. Cumulative Trauma Disorders

The theoretical parts of this chapter mainly follow the instructions of Public Employees Occupational Safety and Health Program of New Jersey State (Peosh, 1997). Cumulative trauma disorders (CTDs) are injuries of the musculoskeletal and nervous systems that may be caused by repetitive tasks, forceful exertions, vibrations, mechanical compression (pressing against hard surfaces), or sustained or awkward positions. Cumulative trauma disorders are also called repetitive motion disorders (RMDs), overuse syndromes, regional musculoskeletal disorders, repetitive motion injuries, or repetitive strain injuries.

These painful and sometimes crippling disorders develop gradually over periods of weeks, months, or years. They include the following disorders which may be seen in office workers;

Carpal Tunnel Syndrome (CTS) - a compression of the median nerve in the wrist that may be caused by swelling and irritation of tendons and tendon sheaths.

Tendinitis - an inflammation (swelling) or irritation of a tendon. It develops when the tendon is repeatedly tensed from overuse or unaccustomed use of the hand, wrist, arm, or shoulder.

Tenosynovitis - an inflammation (swelling) or irritation of a tendon sheath associated with extreme flexion and extension of the wrist.

Low Back Disorders (LBD) - these include pulled or strained muscles, ligaments, tendons, or ruptured disks. They may be caused by cumulative effects of faulty body mechanics, poor posture, and/or improper lifting techniques.

Synovitis - an inflammation (swelling) or irritation of a synovial lining (joint lining).

De Quervain's Disease - a type of synovitis that involves the base of the thumb.

Bursitis - an inflammation (swelling) or irritation of the connective tissue surrounding a joint, usually of the shoulder.

Epicondylitis - elbow pain associated with extreme rotation of the forearm and bending of the wrist. The condition is also called tennis elbow or golfer's elbow.

Thoracic Outlet Syndrome - a compression of nerves and blood vessels between the first rib, clavicle (collar bone), and accompanying muscles as they leave the thorax (chest) and enter the shoulder.

Cervical Radiculopathy - a compression of the nerve roots in the neck.

Ulnar Nerve Entrapment (UNE) - a compression of the ulnar nerve in the wrist.

Work Related Upper Limb Disorder (WRULD) - is a general term that concerns the chronic pain that can take place in any part arms, such as elbow, wrist, hands and fingers as well as neck and shoulders. This kind of disorders can also be named as “upper limb work-related musculoskeletal disorders *(UL-WMSDS)*”.

Cumulative trauma disorders can also result from other than work activities that involve repetitive motions or sustained awkward positions such as sports or hobbies. Work and non-work activities may together contribute to cumulative trauma disorders. These disorders can also be aggravated by medical conditions such as diabetes, rheumatoid arthritis, gout, multiple myeloma, thyroid disorders, amyloid disease and pregnancy (Peosh, 1997).

Careful positioning of the body at the video display terminal (VDT) can reduce the likelihood of injuries. In some cases, furniture may have to be readjusted or replaced in order to allow for good working postures. Some of the precautions that should be taken in office works to minimize CTD’s are as follows (Peosh, 1997);

- Wrists should be in a neutral position, that is not flexed or dropped. Bent wrists can lead to Carpal Tunnel Syndrome.
- The least amount of pressure needed when striking the keys should be used.
- Feet should rest on the ground or a foot rest to relieve pressure on the lower back. Dangling legs add pressure to the thighs that could cut off blood flow to the legs.
- The head should face forward and be titled slightly downward (5-30 degrees) in order to put the least demand on the neck and shoulders.
- The forearm should not be raised too much (elbow angle should be almost a right angle, or 70-135 degrees) to avoid neck and shoulder pain.
- The material being worked on should all be near the typewriter or word processor. This will reduce twisting which may damage the back. It will also reduce reaching, which can strain the back and shoulders.
- Supporting the lower back and resting it by leaning back frequently and by supporting the arms.

- Ensuring adequate clearance for thighs and feet by keeping areas under the desk clear, and by using desks or tables that are high enough. Free movement is important for supporting the back and for circulation in the legs.
- Not staying in one working posture continuously. Shift positions so no muscles are tensed in the same position for too long.
- Taking breaks in appropriate frequency during the work.

It is essential to train workers on what postures prevent cumulative trauma disorder, on the importance of taking breaks and exercising, and on how to adjust furniture. Supervisors also need to be aware of these subjects.

Fagarasanu and Kumar (2003) argue that “carpal tunnel syndrome (CTS), the most commonly reported nerve entrapment syndrome, results in the highest number of days lost per case among all work-related illnesses. According to National Centre for Health Statistics of USA, almost half of the carpal tunnel cases resulted in 31 days or more of work loss. CTS is the most common nerve compression and the most common and costly repetitive strain illness n hands. The non-medical costs of a CTS case from compensation settlements and disability average $10,000/hand. This sum is increased by the medical cost and indirect costs that raises it to $20,000–$100,000/hand. U.S. Department of Labour defends that up to 36 per cent of all CTS patients require lifelong medical treatment; the total costs are enormous” (Fagarasanu and Kumar, 2003). Similar statistics have not been done yet by Ministry of Health, Ministry of Labour or universities in Turkey, but it is thought to be parallel or a little worse than that of US statistics. Taking into account the increased muscle activation due to high demanding cognitive tasks, which are present in data entry activities, increasing figures are expected in the future. All these costs represent only a small portion of the total costs that are lost due to the poorly designed keyboard and pointing devices.

Yu-Chuan Lin (2009) describes the carpal tunnel as “a rigid cannular path formed by bony walls on three sides and roofed by a tough transverse carpal ligament. Inside the tunnel are the radial bursa with its invested flexor pollicis longus tendon, the ulnar bursa and its invested flexor digitorum superficialis and profundus tendons, and the median nerve with its artery, and a cellulo adipose layer. The exact pathophysiology of carpal tunnel syndrome is unclear. The median nerve may be directly damaged or secondarily compressed. Secondary compression can be further divided into traumatic and nontraumatic conditions" (Lin Chuan, 2009).

2.2.2. Importance of Keyboard Design to Prevent CTDs

The importance of keyboard design to minimize CTDs is best described by the argument that "from the 37,804 cases of *work-related* CTS reported, 7897 *(21 per cent)* were attributed to repetitive typing or key entry data. There is a loss in productivity before (less typing speed), during, and after (days of hospitalization) the treatment of CTS (Fagarasanu and Kumar, 2003). Also, there is solid scientific prove that intensive usage of keyboard is associated with musculoskeletal problems, commonly known as CTDs, a phenomenon which is widespread among keyboard users. It is stated that more than *50 per cent* of *occupational injuries in offices* result in strain injuries from overuse of the hands, such as during typing. Analysis of cumulative trauma injury (CTI) processes emphasizes that unnatural postures and repetitive movements are the main contributing factors. Keyboard operations very often incorporate incorrect wrist postures and highly repetitive key strokes, which result in a cross national and wide spreading problem (Gilad and Harel, 2000).

Keyboards have been used for over 100 years and were very well known long before the introduction of computer input devices. Typewriters can be admitted as the "forefather" of the keyboards. At the beginning, the refinements were for superior mechanical properties and fewer malfunctions. The next 20–25 years emphasized increasing performance and the last 20–25 years have focused on "typist fatigue", "perceived pain", "muscular strain" and "ergonomics". Nowadays, the computer keyboard is the primary input device for data entry tasks. Although the keyboard is often a nonadjustable device, it is used by nearby all the computer users regardless of age, anthropometric characteristics, gender and performance leading to increased musculoskeletal problems (Fagarasanu and Kumar, 2003).

Ergonomists agree that the conventional model of cumulative trauma injuries and repetitive stress injuries processes emphasize the role of three risk factors:

- Force
- Posture
- Repetition

Keyboard operations are a classical example, where these factors are extremely affecting the physiology of the operators (Gilad and Harel, 2000).

Nelson *et al.* (2000) argue that on average, the carpal tunnel tendon travel for 1 hour of continuous typing ranged from 30 to 59 meters. Additionally, repetitive sliding of tendons within their sheaths will increase the friction that is a major trigger for the disorders of the tendons, their sheaths or adjacent nerves during typing activity (Fagarasanu and Kumar, 2003). This information tells that the vital tendons face great risks while typing.

Clearly though and to be honest, there is a limit to the amount of physical work each employee can perform in a given limited duration without developing musculoskeletal disorders (Lingaard and Caple, 2001). Nearly all tasks, even those respectfully designed by keeping in mind all ergonomics criteria, will actually cause CTSs. Consequently, the main aim of ergonomists is the minimization of CTDs, *not* abating all over. So, a range of alternative keyboards has been developed in an effort to reduce or eliminate adverse effects on operator well being and performance (Swanson, *et al.*, 1997)

NIOSH, US National Institute for Occupational Safety and Health, the Federal agency responsible for conducting research and making recommendations for the prevention of work-related disease and injury which is a participant of the Centers for Disease Control and Prevention os USA, lists the following suggestions while making purchase decision if alternative keyboards are to be used in offices (NIOSH, 1998):

- Determine if the keyboard is compatible with existing hardware and software, and whether it can accommodate other input devices such as mice and trackballs.

- Assess how the keyboard will fit with the workstation. Some alternative keyboards, particularly those with a tented design, must be placed on surfaces that are lower than those required for standard keyboards to achieve proper working posture.
- Evaluate whether the keyboard will affect the user's performance. Does the design make it difficult for the user to see the keys? Does the job require a numeric keypad or specialized keys that may not appear on an alternative keyboard?
- Allow users to try a keyboard on a trial basis before buying it.
- Because one type of keyboard will not be appropriate for all users or tasks, allow users to try different kinds before deciding which to buy, and allow them to retain a conventional keyboard if they wish.
- It may take a few days for a user to become accustomed to an alternative keyboard, and frustration may occur if productivity is affected during this learning phase.
- It can be helpful to involve a specialist who knows about and is experienced in office ergonomics, and also to involve a health professional if a computer user has discomfort or musculoskeletal symptoms.
- Integrate a new alternative keyboard carefully into the work process, ensuring that users are trained in correct use.
- Each workplace should have a comprehensive ergonomics program to protect all workers.

Keyboard usage introduces a wide range of risk factors that are present in such important cumulative and simultaneous levels. For instance, Marklin *et al.* (1999) state that excessive wrist extension or flexion is present in different degrees depending on the type and slope angles of keyboard used, and adding that compared with a neutral wrist posture, ulnar deviation of 10^0 does not increase carpal tunnel pressure. Werner *et al.* (1997) also agree with Marklin by remarking that ulnar deviation occurs directly due to the need to reach the far left or right keys and indirectly as a compensation of the arm abduction.

Fagarasanu *et al.* (2003) list the fundamental risk factors that may cause carpal tunnel syndrome during typing as in Table 2.1.

Table 2.1. Carpal Tunnel Syndrome risk factors during typing

1. Keystroke activation force
2. Proprioceptive feedback
3. Percentage of time typing
4. Typing speed
5. Use of a group of fingers
6. Minimum force needed to activate the key switch
7. Typing force
8. Repetitiveness
9. Keyboard height
10. Awkward postures
11. Distance taken by the wrist during typing

Gilad and Harel (2000) clearly showed that during keying the fingers are in a constant tension, extended closely to their anatomical limit angle. When concluding a set of experiments, they stated that EMG readings are greater when fingers cannot be rested on the keys, due to insufficient actuation force, or are deliberately extended to support their weight. Their conclusions include working protocol and recommendations for keys pressure resistance. The muscle electric activity was detected by surface electrodes attached in pairs to each muscle using Electromyography evaluation equipment. Data records were transferred to DOS files for processing and further analysis.

The laboratory test procedure of Gilad and Harel (2000) was as follows: each subject performed the given experimental typing tasks at his own pace and sitting preference. The procedure started in getting familiar with the text and followed by four similar trials in which the subject types given texts on the different keyboards. The first trial was 5 min long; the other three sequential trials were 10 min long each. Muscular electric activity was sampled 8 times per each 5 min. Data was recorded for each trial separately. Subjective evaluation questionnaires were filled during intermissions. Surface electrodes were attached to the following muscle groups, in the upper limb: *Flexor Carpi Ulnaris*, *Extensor Carpi Ulnaris*, *Deltoid* and *Trapezius*. These muscle groups represent the isometric and active strains developed in the neck and shoulders, arms and hands during the typing acts (Gilad and Harel, 2000). The muscle groups of the upper right limb, which are commonly analysed in occupational biomechanics, from both anterior and posterior aspects can be seen in Figure 2.1.

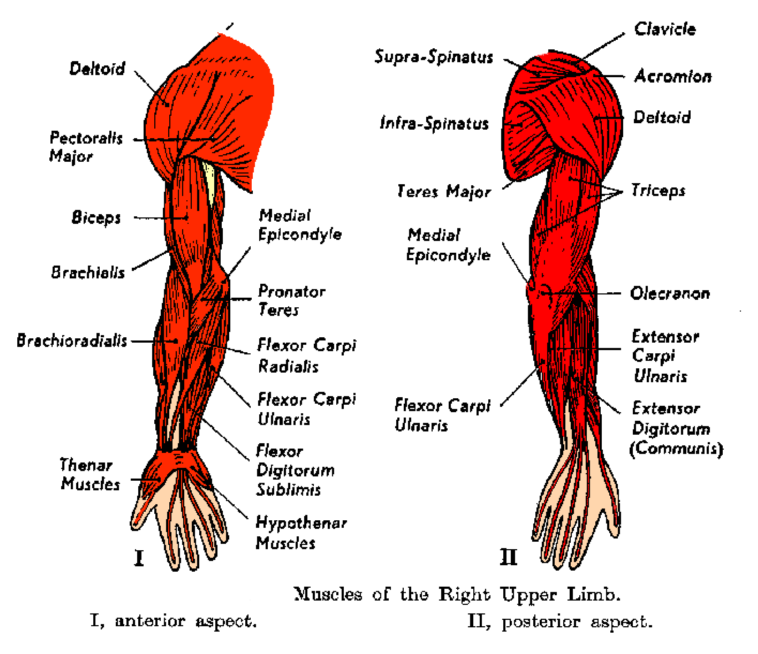


Figure 2.1. The muscle groups of the upper right limb

It should be noted that there are many problems associated with poor postures which have negative long term effects on employee’s body and health. These problematic postures will not only impact how the employee feels but how his/her body looks. If the employee continues to exercise with poor posture, he/she will recruit the wrong muscles and build body disproportionately.

If a body is under the pressure of improper dynamic posture during a movement, or stays in a problematic static posture for a long time, the fatigue level of the affected muscles will increase with a high degree. Swanson *et al.* (1997) present that these problematic postures include forearm pronation close to the anatomical limit, ulnar deviations of 20-40 degrees from neutral, wrist extension and prolonged upper arm/shoulder abduction. Bergamasco *et.al* (1998) quantifiy the limits of some problematic postures during typing activity by saying that "in a task that involves repetitive use of the upper extremity, positions of the arm and hand deemed to be unacceptable are: ulnar deviation > 24^0; radial deviation > 15^0; pronation > 40^0; supination > 57^0; abduction > 67^0; extension > 50^0 and flexion > 45^0 " as shown in Figure 2.2.

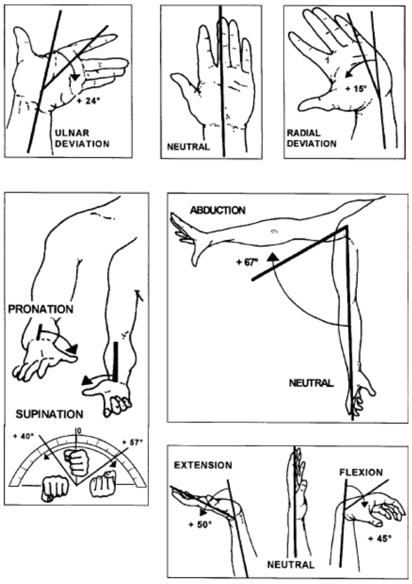


Figure 2.2. Some problematic postures during typing activity

The stated problematic postures should be considered while designing keyboard shape, layout, key stroke strengths, slopes, position on the table and so on. For instance, while designing the slope between the base platform of the keyboard and the table, the wrist extension of target users should be considered. The importance of wrist extension and ulnar deviation of hands during keyboard usage is shown in Figure 2.3.

Figure 2.3. Wrist extension (in left) and ulnar deviation (in right) during typing activity

2.2.3. Basics of Ergonomics Keyboard Design

In order to decrease the problematic postures, ergonomic keyboards are started to be preferred over regular keyboards. Ergonomics keyboards tend to decrease repetitive strain injury on wrists, fingers and hand caused by regular keyboard because of their size and position of keys (Swanson, *et al.*, 1997).

Also, the ergonomic keyboards increase office productivity decreasing the time with short cut keys. It can be argued with no doubt that keyboard is the component of computers which has received the highest attention of ergonomists off late in last decades. They have come up with ergonomically designed keyboards to reduce problems created by standard keyboards. Ergonomists also suggest some rules and tips to use ergonomic keyboard easily and effectively.

Swanson *et al.* (1997) explain that the primary design features of new ergonomics keyboards should include the following fundamental specifics:

(1) 10-15 degrees of horizontal rotation of the right and left keyboard halves in order to reduce ulnar deviation

(2) 25-60 degrees of lateral inclination of the keyboard halves to reduce forearm pronation and abduction of the upper arms

(3) crescent-shaped, rather than parallel, key rows to conform to the anatomical shape of the hand.

2.3. Importance of Ergonomics for Keyboard Layout

Like a great portion of ergonomists and expert keyboard users, Eggers and his friends (2003) defend that the character arrangement on a keyboard can affect a person's comfort and typing performance. Typing activities, lasting long time with an inefficiently designed layout, can cause severe problems in wrists, elbows, fingers or hands, such as carpal tunnel syndrome or repetitive strain injuries, which is the main fundamental reason why ergonomists give importance to keyboards.

The main design problems of QWERTY keyboard, which are also the basic reasons for causing fatigue and discomfort on most of the its users, are the poor shape and the poor key allocation. Because, the standard QWERTY was designed for 2-finger typists (*typewriter users*), it does not efficiently allocate keys to fingers as some fingers are requested to perform much work than others (Dell'Amico *et al.*, 2009). For example, while typing a Turkish text on QWERTY layout, the left pinky finger is used very often due to the location of letter "A", however the letter "J", which is rare in Turkish texts, is located just to the key which is switched by right index finger. In the same manner, as demonstrated by Swanson *et al.* (1997), the left hand and fingers (typically weaker than the right ones) are aimed at handling the most frequently used letters in English.

Fagarasanu and Kumar (2003) also state that typing with a QWERTY keyboard, specifically, is a great risk for carpal tunnel syndrome due to its alphabetical layout. On the basis of their analysis that the following defects exist in the QWERTY design:

- Overloading of the weaker left hand in a right handed person
- Overworking certain fingers and not assigning enough work to others

- Too little typing on the home row
- Fingers are required to execute an excessive amount of jumping back and forth from row to row

The Dvorak keyboard developed in 1932 and patented in 1936 by the educational psychologist August Dvorak, where the vowel letters are placed in the home row is claimed to overcome most of the limitations of the QWERTY keyboard. It has been designed on the basis of how frequent is the use of different letters, including the frequency of two, three, four and five sequences of symbols. However, the Dvorak keyboard has not completely replaced the QWERTY keyboard primarily due to long term adaptation of the QWERTY keyboard. Because, it is very difficult to change the hard-won habits of the users. In addition, the level of enhancement with a Dvorak keyboard has not really justified the time and cost of retraining users. As a result, it can be argued with no doubt that, although Dvorak layout is more effective than the QWERTY one, it was unfortunately never accepted by the general population (Dell'Amico *et al.*, 2009).

Recently, Operations Research and Ergonomics started working together to design "optimal" keyboards by means of quantitative methods, due to the importance of the keyboard layouts for health and productivity of the users. Eggers, *et al.* (2003) considered the problem of assigning characters to keys arranged in a prespecified layout structure. They designed a weighting method based on six performance indicators:

a. Distribution of the work load among all fingers proportionally to their capacity
b. The hits number needed to compose a text
c. Comfort and speed guaranteed when consecutive keys are not hit by the same *hand*
d. Comfort and speed guaranteed when consecutive keys are not hit by the same *fingers*
e. Avoidance of great steps among two different keys
f. Hits direction that should move from the little finger towards the thumb

At the end, they achieved a global score computed by a weighted linear combination of these six scores to model a keyboard layout optimization.

With the help of these six performance indicators, Prajapati *et al.* (2008) argue that an optimal physical 10-finger keyboard layout should have the following qualities:

- Allow for minimum typing effort (minimizing typing fatigue)
- Maximize typing speed
- Reduce typing errors
- Allow easy learning of the touch typing method.

2.4. Modelling the Ergonomics Criteria of n-Finger Keyboard Layouts

In literature, most of the n-finger keyboard optimization applications follow similar ways. First, the performance indicators are defined and modelled mathematically, following the construction of the objective function as a linear combination of predefined performance indicators. For instance, Eggers *et al*. (2003) associate a score for each of the performance criterion; and then compute a global score by a simple weighted linear combination of these scores. They have used six performance criteria for n-finger keyboard optimization with the mathematical formulations given in the following sections.

2.4.1. Accessibility and Load

The strength of all fingers is not the same anatomically; ass can be seen in Figure 2.5. The third row in the figure is the home row (*the row which starts with A-S-D-F-... in standard Turkish Q Layout*). In this Figure; the first column is the Y-H-N keys column, second is the U-J-M column, third is the I-K-Ö column, forth is the O-L-Ç column, fifth is the P-Ş-“.” column. The remaining columns are the columns involving punctuation keys; enter key, backspace key and so on, which are controlled by the right pinky finger during typing. For this reason, the combination character sets assignments should be made on the keys which try to share the total road adequately by all fingers.

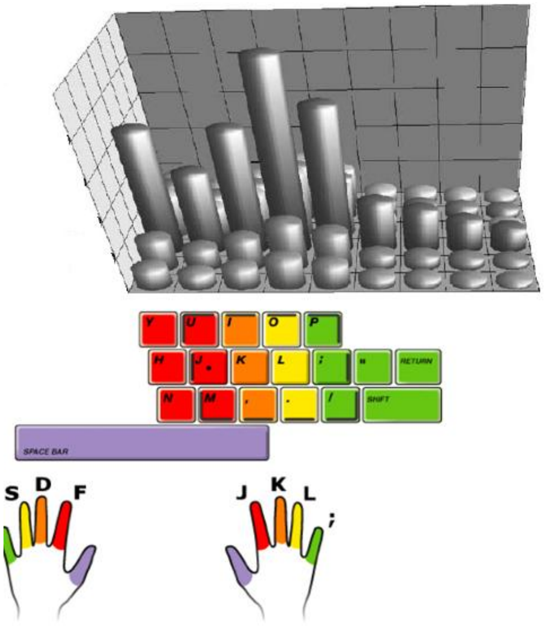


Figure 2.4. Ideal load distribution for the right hand

While achieving this criterion, the fact that some keys are less accessible than others has to be taken into account. The score for this criterion evaluates the variance of the load distribution on a keyboard from the ideal distribution and is given by the following equation

$$v_1 = \sum_{m_i \in \Xi_1^{\mathrm{m}}} (f_{m_i} - f_{m_i}^{opt})^2 \tag{2.1}$$

where Ξ_1^m is the set of all monographs, m_i is one of the element of the layout set for a language, f_{m_i} is the load distribution of fingers on a layout and $f_{m_i}^{opt}$ is the ideal load distribution of fingers (Eggers *et al.*, 2003).

2.4.2. Key Number

This criterion shows the total number of keystrokes to write a text. This criterion is usually used for optimizing the keyboards of the languages which have a great number of characters. In this kind of languages, more than one letter is assigned to a key. The secondary letters of the keys are typed by help of previously stroke another key, such as shift, ctrl and so on. The score v_2 is therefore calculated by dividing the number of characters in a text by the number of keystrokes necessary to produce the text. While it is relatively important for the most general definition, this score does not vary between the different solutions of the optimization model (Eggers *et al.*, 2003).

$$v_2 = \frac{number\ of\ characters\ in\ a\ text}{average\ of\ number\ of\ keystrokes\ to\ write\ a\ text} \tag{2.2}$$

2.4.3. Hand Alternation

In physical keyboards, it is a preferred action to type consecutive letters by different hands. This attitude increases the typing speed. Because, when one of the hands is on the action of typing a letter, the other hand simultaneously positions itself to the next key. Hand alternation is also comfortable for the user from the ergonomics view, due to resting time of muscles groups. Hand alternation score is calculated by summing the frequencies of the digraphs which are typed by the fingers of the same hand. For example, while typing “elma” in Turkish Q layout; the digraphs “el” and “ma” are typed with hand alternation whereas the digraph “lm” is written by the fingers of only right hand. This criterion is defined mathematically as follows:

$$v_3 = \sum_{d_i \in \Xi_3^{\mathrm{d}}} f_{d_i} \tag{2.3}$$

where Ξ_3^d is the set of all digraphs which are typed with same hand, d_i is one of the digraphs of set Ξ_3^d and f_{d_i} is the frequency of digraph d_i (Eggers *et al.*, 2003).

2.4.4. Consecutive Usage of the Same Finger

Like the previous criterion about hand alternation, it is not a preferred action to type two consecutive letters by the same finger of one hand. For instance, while typing "ayhan" in Turkish Q layout, the digraph "yh" is typed by the index finger of the right hand.

While calculating this criterion, it also should be noted that, the distance that has to be taken by the consecutively used finger is important. To understand the situation better, typing the words "ayhan" and "ayna" in Q layout can be analyzed. While typing the digraph "yh" in the word "ayhan", the index finger of the right hand moves about one-key length (from "y" to "h"). However, while typing the digraph "yn" in the word "ayna", the index finger of the right hand moves about two-key length (from "y" to "n"). This criterion can be expressed mathematically as:

$$v_4 = \sum_{d_i \in \Xi_4^{\mathrm{d}}} f_{d_i} * dist(d_i) \tag{2.4}$$

where Ξ_4^d is the set of all digraphs whose letters are typed with the same finger of same hand and $dist(d_i)$ is the distance between the letters of a digraph in a layout. This distance can be calculated by different formulas such as Manhattan distance (Eggers *et al.*, 2003):

$$dist(d_i) = |c_2 - c_1| + |r_2 - r_1| \tag{2.5}$$

where c_2 and c_2 are the respective columns of the two keys which establish the digraph and r_2 and r_1 the corresponding rows. This distance can also be calculated by Pythagorean Theorem as:

$$dist(d_i) = \sqrt{(x_2 - x_{1)}}^2 + (y_2 - y_{1)}}^2} \tag{2.6}$$

where x_2 and x_1 are the respective x-coordinates of the two keys which establish the digraph and y_2 and y_1 the corresponding y-coordinates of the two keys, for an any x-y coordinate system on a layout.

2.4.5. Avoiding Big Steps

During typing, if the same hand but different finger is used for two consecutive hits (*digraph*), then the big steps of hand which lead to awkward postures causing cumulative trauma disorders should be tried to be minimized. To define this criterion, a weight coefficient depending on the two fingers used is assigned to each digraph. These coefficients are the result of the anatomical difference of fingers. For example, two consecutive hits of the same hand whose first hit is made by pinky finger and the second one made by ring finger is a better direction than the reverse one. The heuristic coefficient table used by Eggers *et al.* (2003) is shown on Table 2.2.

Table 2.2. Big step coefficients

	Thumb	Index Finger	Middle Finger	Ring Finger	Little Finger
Thumb	**0**	**0**	**0**	**0**	**0**
Index Finger	**0**	**0**	**5**	**8**	**6**
Middle Finger	**0**	**5**	**0**	**9**	**7**
Ring Finger	**0**	**8**	**9**	**0**	**10**
Little Finger	**0**	**6**	**7**	**10**	**0**

Then the big step criterion can be expressed mathematically as:

$$v_5 = \sum_{d_i \in \Xi_5^{\mathrm{d}}} \kappa(d_i) * f_{d_i} \tag{2.7}$$

where $\kappa(d_i)$ is the weight coefficient. Here, an approximate value of $\kappa(d_i)$ is defined as $\kappa(d_i) = \kappa(u, v)$; u and v representing the first and second finger to type the digraph d_i, respectively. The relevant set Ξ_5^d is therefore the set of digraphs which are typed using the same hand, but not the same finger, and the vertical distance between the two keys is greater than or equal to one row. Also, f_{d_i} is the frequency of digraph $d_{i.}$

2.4.6. Hit Direction

For digraphs typed by one hand only, the preferred hit direction is from the little finger towards the thumb. This is the natural finger movement for most people, which may easily be verified by tapping on the table according to the two possible directions. Ξ_6^d is therefore the set of all digraphs which are produced by using one hand only and whose hit direction is *not* the preferred one. The score is given by:

$$v_6 = \sum_{d_i \in \Xi_6^{\mathrm{d}}} f_{d_i} \tag{2.8}$$

2.4.7. Global Score

The six criteria scores v_j (for $1 \leq j \leq 6$) are combined linearly to form a global score. This global score can be used as an objective function for the physical keyboard optimization problem. Since the composing six elements have different units and different relative importance, all terms are first divided by the related scores of a reference keyboard, $v_{j,ref}$.

Then, these dimensionless terms are summed after multiplied by a relative weight coefficient γ_j. Hence, the global score can be formulated as the following expression:

$$V = \sum_{j=1}^{6} \frac{v_j}{v_{j,ref}} \gamma_j \tag{2.9}$$

These weight coefficients are generally calculated heuristically by ergonomics experts. Eggers *et al.* (2003) used the weight coefficients prepared by two specialized ergonomists by a pair wise comparison method. These coefficients can be seen in Table 2.3.

Table 2.3. Weight coefficients of six criteria in global score, γ_j

Criterion	$\boldsymbol{\gamma_j}$
Load and accessibility	0.45
Key number	0.50
Hand alternation	1.00
Consecutive usage of the same finger	0.80
Avoiding big steps	0.70
Hit direction	0.60

To sum up; typical criteria to evaluate n-finger keyboards (*physical keyboards*) can be listed as posture structure, discomfort produced, keying force, user acceptance, repetitive usage of same finger or same hand, size of the steps of the finger and so on (Dell’Amico *et al.*, 2009). Hence, there are a great number of quality measures while designing n-finger keyboards.

2.5. Modelling the Ergonomics Criteria of Stylus Keyboard Layouts

Contrary to n-finger keyboards, there are fewer quality measures for stylus keyboards. The major task for optimization of stylus keyboards has been trying to minimize the time spent to write a text. Minor tasks can be remarked as to improve the typing accurateness and performance of the user due to multitasking conditions, such as typing while walking (Dell’Amico *et al.*, 2009).

Yanzhi *et al.* (2006) defend that “with the popularity of mobile devices, designing a keyboard for users who can only operate with a pointer or the eyes is a challenging task; however numerical simulations show that the present day keyboards and arrangements are not optimal for such applications. When holding a mobile device, only one hand or one pointer is available for data input. Thus, it is important to minimize the movement time in order to improve performance and minimize potential fatigue”.

3. HUMAN-COMPUTER INTERACTION MODELS FOR STYLUS KEYBOARDS

This chapter briefly describes the human-computer interaction, human motor system and three fundamental human-computer interaction models used while designing or evaluating the performance of stylus keyboard layouts. This chapter stands on the basis of theoretical definitions from Wikipedia, the free encyclopaedia.

3.1. Introduction to Human Computer Interaction

Human–computer interaction (HCI) can be described briefly as the study of interaction between people (users) and computers. HCI is often regarded as the intersection of computer science, behavioural sciences, design and several other fields of study. Interaction between users and computers occurs at the user interface, which includes both software and hardware; for example, characters or objects displayed by software on a personal computer's monitor, input received from users via hardware peripherals such as keyboards and mice, and other user interactions with large-scale computerized systems such as aircraft and power plants (HCI, Wikipedia).

Human motor system creates a response just after an attention signal is somewhat gathered by any of the sensors of the human body. First, sensory processing is in charge to identify the properties of the attention source. After, cognitive processor takes the job with active memory chunks and long term memory, if necessary. Then the motor processing, which is the response of the body to that stimuli, comes.

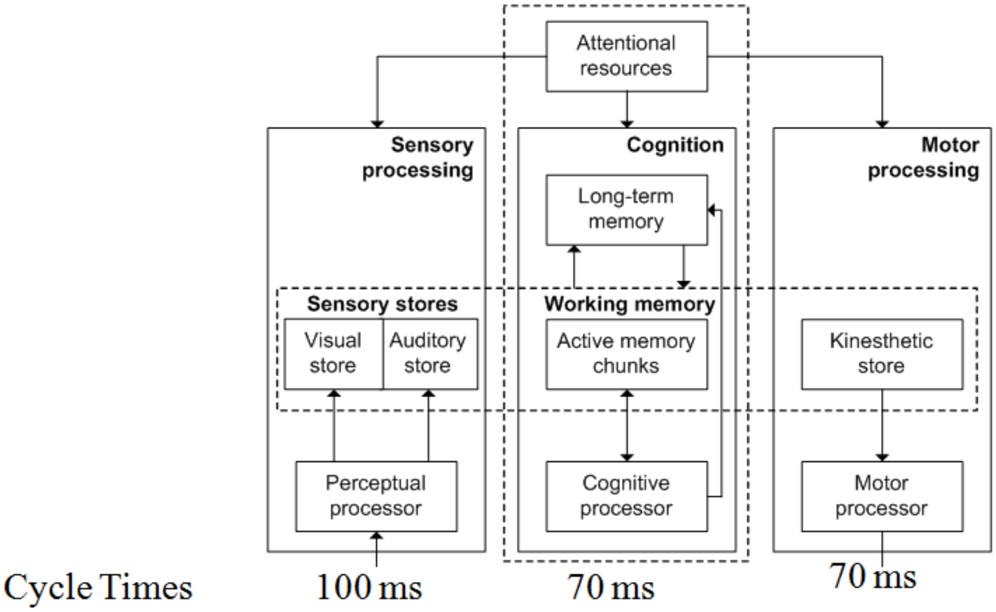


Figure 3.1. Perceive-recognise-act cycle of human motor system

Due to the multidisciplinary nature of HCI, people with different backgrounds contribute to its success. HCI is also sometimes referred to as man–machine interaction (MMI) or computer–human interaction (CHI).

3.2. Basic Goals of Human Computer Interaction

The main goal of HCI is to improve the interactions between users and computers by making computers more usable and receptive to the user's needs. Specifically, HCI is concerned with (HCI, Wikipedia);

- methodologies and processes for designing interfaces (i.e., given a task and a class of users, design the best possible interface within given constraints, optimizing for a desired property such as learning ability or efficiency of use)
- methods for implementing interfaces (e.g. software toolkits and libraries; efficient algorithms)

- techniques for evaluating and comparing interfaces
- developing new interfaces and interaction techniques
- developing descriptive and predictive models and theories of interaction

A long term goal of HCI is to design systems that minimize the barrier between the human's cognitive model of what they want to accomplish and the computer's understanding of the user's task.

Professional practitioners in HCI are usually designers concerned with the practical application of design methodologies to real-world problems. Their work often revolves around designing graphical user interfaces and web interfaces.

Researchers in HCI are interested in developing new design methodologies, experimenting with new hardware devices, prototyping new software systems, exploring new paradigms for interaction, and developing models and theories of interaction.

3.3. Main Design Principles for HCI

When evaluating a current user interface such as stylus keyboard, or designing a new user interface, it is important to keep in mind the following experimental design principles (HCI, Wikipedia);

- Early focus on user(s) and task(s): Establish how many users are needed to perform the task(s) and determine who the appropriate users should be; someone that has never used the interface, and will not use the interface in the future, is most likely not a valid user. In addition, define the task(s) the users will be performing and how often the task(s) need to be performed.
- Empirical measurement: Test the interface early on with real users who come in contact with the interface on an everyday basis. Keep in mind that results may be altered if the performance level of the user is not an accurate depiction of the real

human-computer interaction. Establish quantitative usability specifics such as: the number of users performing the task(s), the time to complete the task(s), and the number of errors made during the task(s).

- Iterative design: After determining the users, tasks, and empirical measurements to include, perform the following four iterative design steps:

1. Design the user interface
2. Test
3. Analyze results
4. Repeat

Consequently, repeat the iterative design process until a sensible, user-friendly interface is created. While designing a stylus keyboard, these steps would be worthwhile both in design and also testing phases.

3.4. Human Performance Models While Using Stylus Keyboard

In literature, the performance of a keyboard user has been modelled by different models of human computer interaction. The researchers often model a keyboard user by some generally accepted models to predict the typing speed of an expert user on that layout. This is indeed a reasonable way, because most of the subjects are familiar with currently in use layouts. They will tend to type faster in QWERTY layout, for instance. Even a new layout which is better than QWERTY may get worse performance indicator values wrongfully.

3.5. Fitts’ Law

3.5.1. Definition

Fitts's law (*often cited as Fitts' law*) is a model of human movement in human-computer interaction and ergonomics which predicts that the time required to rapidly move to a target area is a function of the distance and the size of the target. Fitts's law is used to model the act of pointing, either by physically touching an object with a hand or finger, or virtually, by pointing to an object on a computer display using a pointing device (Fitts’ Law, Wikipedia). Published in 1954, Fitts’s Law is an effective method of modeling the relationship of a very specific, yet common situation in interface design.

3.5.2. Application area

It should be noted that Fitts’ Law can be applied to either touch screen keyboards or to keyboards whose keys are stroke by mouse. Of course, the typing speeds will be very different. Because, hitting an area with motor system controlled finger or finger controlled stylus is easier than pointing it on the computer screen. This situation was also explained by Zhai (2004) as “one important aspect of computer input research is to measure and characterize the performance of various input systems. Given the great potential diversity of input devices, such as the mouse, joystick and touchpad (or different versions of the same type of device), a critical need is to be able to compare and characterize them from a human performance perspective”.

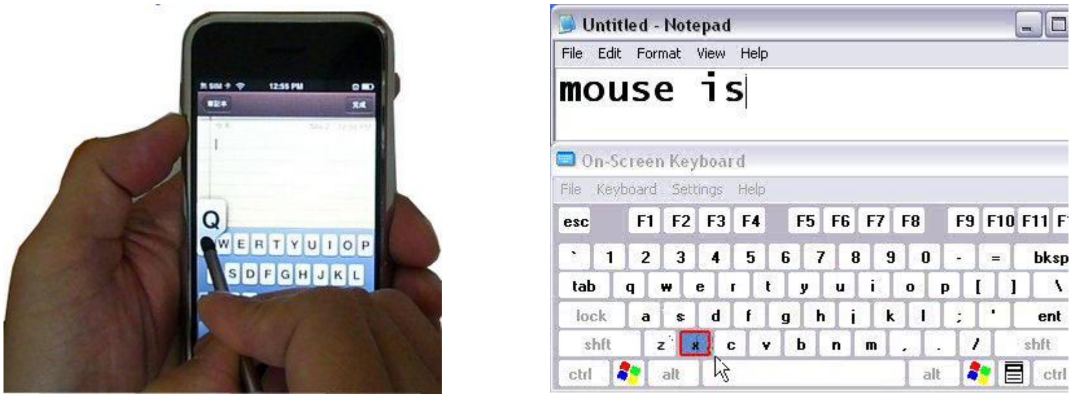


Figure 3.2. Stylus keyboard usage by pen (left) and mouse (right)

3.5.3. Formulation

Fitts tried to fit a regression formula to the data he gathered during his experiments. Also, there are a number of different interpretations of Fitts' Law in literature. These versions of Fitts' formulation have slight differences, but the main model remains the same. All formulations aim to model the relationship between the mean time (MT) to point a target and index of difficulty (ID).

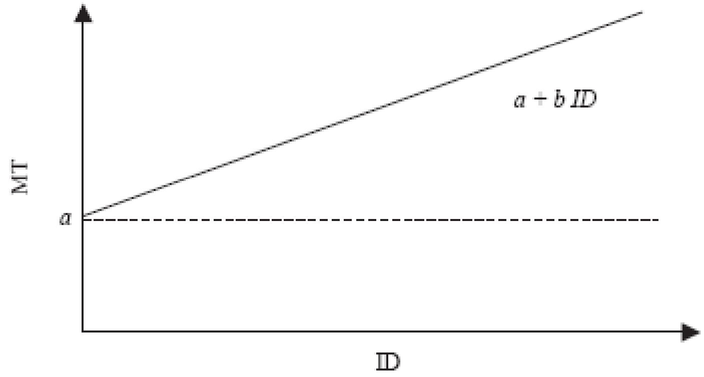


Figure 3.3. Relation of MT with ID in Fitts' Law

Fitts (1954) explains his suggestions in his original paper: “An index of the difficulty of a movement is proposed on the assumption that the average amplitude, the average duration, and the amplitude variability of successive movements are related in a manner suggested by information theory. The basic rationale is that the minimum amount of information required to produce a movement having a particular average amplitude plus or minus a specified tolerance (variable error) is proportional to the logarithm of the ratio of the tolerance to the possible amplitude range is arbitrary and has been set at twice the average amplitude. The average rate of information generated by a series of movements is the average information per movement divided by the time per movement” (Fitts, 1954).

According to Fitts (1954), a movement tasks' difficulty (*ID*, the "index of difficulty") can be quantified using information theory by the metric "bits". Specifically,

$$ID = log_2\left(\frac{2A}{W}\right) \tag{3.1}$$

where A is the distance or amplitude to move and W is the width or tolerance of the region within which the move terminates. Because A and W are both measures of distance, the term within the parentheses is without units. The unit "bits" emerges from the somewhat arbitrary choice of base 2 for the logarithm. From this equation, the time to complete a movement task is predicted using a simple linear equation, where movement time (*MT*) is a linear function of *ID* (MacKenzie, 1995). Derivation of Fitts’ Law can be found in Appendix A.

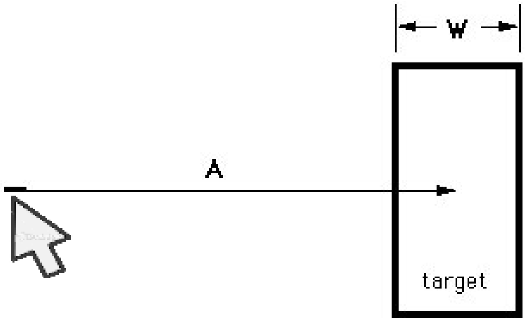


Figure 3.4. Application of Fitts’ Law to point a target on a screen

3.5.4. Fitts' Law and Stylus Keyboards

In literature a number of revised versions of Fitts' original formula, for estimating the typing speeds on for stylus keyboards, have been used (MacKenzie and Soukoreff, 2002), (Zhai *et al.*, 2000) and (MacKenzie and Zhang, 1999).

They defend that according to Fitts' law, the mean time to move the tapping stylus from one key i to another j for a given distance (D_{ij}) and key width (W_j) is

$$t_{ij} = a + b\ log_2\left(\frac{D_{ij} + W_j}{W_j}\right) \tag{3.2}$$

where *a* and *b* are empirically determined coefficients.

There is a wide range of values of the Fitts' law parameters (*a* and *b*) reported in the literature. Index of performance (*IP*) is sometimes used instead of the parameter *b*. For example, *IP (=1/b)* has been reported as low as 4.9 bits per second (bps) in (MacKenzie et al., 1991) and as high as 12 bps in (Fitts, 1954). As the Fitts' law parameters change, the movement speed limit on virtual keyboards change dramatically. Without empirically measuring performance specifically for stylus keyboarding, researchers like MacKenzie and Zhang (1991) tended to use the more conservative estimates of $a = 0$; $b = 1/4.9$ (*second*) based on results from the more general Fitts' reciprocal tapping tasks.

More recently, it was estimated that the values of *a* and *b* in Fitts' law in the context of stylus typing which involves a relatively small range of index of difficulty formed by tightly packed targets, the use of stylus, and visual recognition of a series of intended target letters (but excluding visual search needed by the novice users of stylus keyboards) in varied directions. Their results were $a = 0.083$; $b = 0.127$ s, or $IP = 1/b = 7.9$ bps (Zhai, 2004).

Once the basic Fitts' law parameters are estimated, the average tapping time (per key) on a stylus keyboard (touch screen board) can be obtained by calculating the average of all pairs of keys as

$$\bar{t} = \sum_{i,j \in S} p_{ij} \, . \, t_{ij} \tag{3.3}$$

where S is the key set, usually including the 26 letters in English and the space key, and p_{ij} is the diagraph transition probability from letter i to letter j in a language corpus. If these two equations are combined, then the Fitts' Law applicable to stylus keyboard typing will be as follows:

$$\bar{t} = \sum_{i,j \in S} [a + b . p_{ij} \, . \, log_2 \left(\frac{D_{ij} \, + W_j}{W_j} \right)] \tag{3.4}$$

Zhai and Kristensson (2005) used the above formula with the parameters *a = 0.083* and *b = 0.127* s, thus the formula becomes:

$$\bar{t} = 0.083 + \; 0.127 \, . \sum_{i,j \in S} p_{ij} \, . \, log_2 \left(\frac{D_{ij} \, + W_j}{W_j} \right) \tag{3.5}$$

Fitts (1954) made his experiments in one-dimension surface. He always directly took the width of the target since the movement was always perpendicular to at least one of the edge of the target. The roles of width and height reverse as the approach angle changes from 0° to 90° as depicted in Figure 3.5 (MacKenzie, 1995).

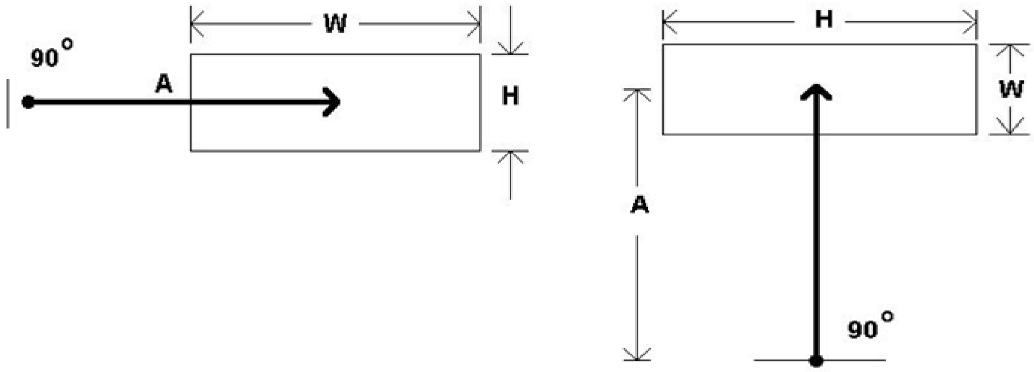


Figure 3.5. Fitts' Law in two dimensions by (MacKenzie, 1995)

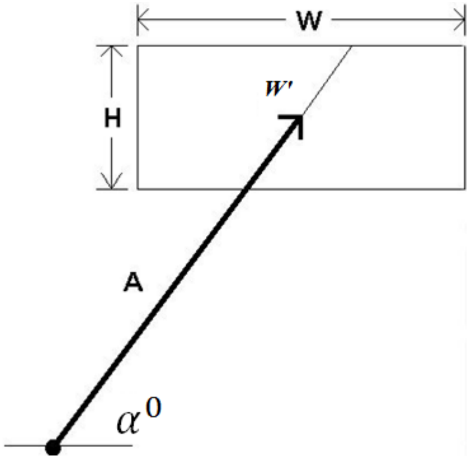


Figure 3.6. Application of Fitts' Law to point a target on a screen by (MacKenzie, 1995)

However, it is not the case while typing on stylus keyboard. The approach angle of stylus continuously changes while tapping target key. So, the researcher should decide whether to take *H*, *W* or *W'* as the width of the target key. *W'*, with no doubt, is the most accurate length since it is the width on the approach angle. However, calculating *W'* takes a huge amount of time and massy calculations. For this reason, when the keys are in square shape, some researchers tend to take *W'* equal to one edge length of a key.

3.6. Hick-Hyman Law

Hick's Law, named after British psychologist William Edmund Hick, or the Hick–Hyman Law (*for the honour of Ray Hyman*), describes the time it takes for a person to make a decision as a result of the possible choices he or she has. The Hick-Hyman Law assesses cognitive information capacity in choice reaction experiments. The amount of time taken to process a certain amount of bits in the Hick-Hyman Law is known as the rate of gain of information. Given n equally probable choices, the average reaction time T required to choose among them is approximately:

$$T = b.log_2(n + 1) \tag{3.6}$$

where b is a constant that can be determined empirically by fitting a line to measured data. Operation of logarithm here expresses depth of "choice tree" hierarchy. The +1 is "because there is uncertainty about whether to respond or not, as well as about which response to make" (Hick's Law, Wiki).

3.7. Comparison of Virtual and Physical Pointing

Hundreds of derivative experiments have been performed since the publication of Fitts's findings. Differences between how well we pointed at objects in real space versus objects on the computer screen was analysed in literature in detail. It was shown that the movement from the starting point to the target area could be divided into two parts (Fitts Law, Wiki):

- the initial high velocity phase
- a deceleration phase

It was discovered that the first phase was only affected by the distance away from the target. Neither the scale of the display nor the size of an object made the user approach more quickly from the start. The phase that actually affects the time to select a smaller object at the same distance is in the deceleration phase. Now, here's the "interesting" part:

"The difference between the virtual and physical display is apparent only in the second movement phase, where visual control of deceleration to the smaller targets in the virtual task took more time than in the physical task."

Figure 3.7. Comparison of virtual and physical pointing by (Fitts Law, Wiki)

Basically, links and buttons on a screen are harder to point out with your mouse than with your finger. And the problem with mice apparently is not in their ability to get to the target, but in our ability to decelerate accurately with them (Fitts Law, Wiki).

As a conclusion, the user that types on a touch screen will type the same text faster than the user that types by using mouse.

4. A SURVEY ON KEYBOARD LAYOUT OPTIMIZATION

Ergonomists and expert users heavily agree on that good keyboard designs can reduce the health risks and pain when these keyboards are used with computer related office works. Also, “typing time” which is representative of speed of performance has been predominantly used to evaluate or design keyboard layouts (Fagarasanu and Kumar, 2003).

After 1930’s, it was started to be accepted by the expert typists that QWERTY layout was a reason of pain and cumulative trauma disorders. It was just after that when the first attempts to re-design this layout had been carried out. First attempts were basically made by shifting a few pairs of letters. Fundamental ergonomic criterion while doing these tries were the anatomic differences of fingers.

With the heightened emphasis of computers in everyday life, it becomes increasingly necessary for an efficient method of data entry. Instead of creating a completely new and possibly unwieldy device to solve this problem, it may be more efficient to merely optimize the tools that are currently in use for faster entry.

During the period of 1930’s to 1960’s, numerous new layouts were proposed for physical keyboards. By the popularity of mobile devices and internet based technology the keyboard layout optimization shifted to stylus keyboards from physical keyboards for different languages, especially for English, German, French, Chinese and Spanish.

In literature, the keyboard layout optimization attempts for Latin languages can be categorized into two main groups as physical keyboards and stylus keyboards.

4.1. Physical Keyboard Layouts

These layouts are commonly the result of the desire to minimize the pain and health risks due to QWERTY layout usage. Since these layouts are physical, usually the devices which plugged these layouts have only one layout. Also, the main characteristic of these layouts is that they are designed for 10 finger typing.

4.1.1. QWERTY Layout

QWERTY layout was initially designed for typewriter users. One of the reason told for QWERTY design was to slow down typing and separate the keys so, that two keys pressed one after another are far away and doesn’t jam the early typewriters. This layout was first developed for English language as seen in Figure 4.1. It also helped salesmen to impress their clients by typing “Type writer” from single row .This resulted in very inefficient design. QWERTY doesn’t make efficient use of home row, it places some of the most used letters on other rows, according to few researches, its home row is used only 32 per cent of the time and not more than 100 English language words can be typed without leaving its home row, whereas this amount is more than 400 for more efficient layouts. Even with all the problems mentioned, once it started getting widely used, it became difficult to change the layout cause of simple economics, as no one was able to justify expense of time and money to convert existing typists to new layout.

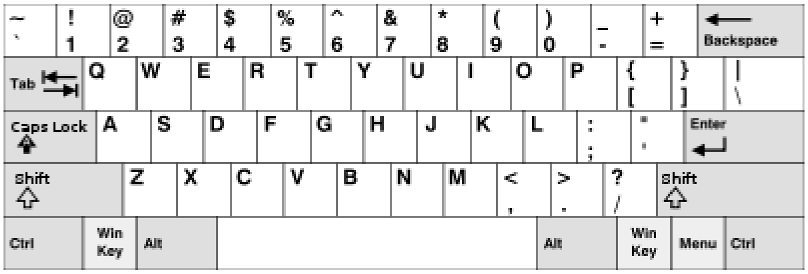


Figure 4.1. QWERTY layout for English speaking countries

4.1.2. Dvorak Layout

It is a physical keyboard layout designed for speed typing. The Dvorak keyboard was designed in the 1930s by August Dvorak, a professor of education, and his brother-in-law, William Dealy. Unlike the traditional QWERTY keyboard, the Dvorak keyboard layout is designed so that the middle row of keys includes the most common letters in English. In addition, common letter combinations are positioned in such a way that they can be typed quickly.

To describe the efficiency of this layout, it can be said that it has been estimated that in an average eight-hour day, a typist's hands travel over 16 miles on a QWERTY keyboard, but only 1 mile on a Dvorak keyboard.

In addition to the standard Dvorak keyboard, there are two additional Dvorak keyboards, a left-handed and right-handed keyboard. These keyboards are designed for people who have only one hand for typing.

From 1931-1935 Dvorak tried to come up with a keyboard layout faster and more efficient than the widely known and used QWERTY. Using letter frequency and letter sequences, the result of their experiments is now known as the Dvorak Simplified Keyboard. Dvorak hypothesized that his keyboard layout could be mastered in approximately 50 per cent of the time it took to master the QWERTY layout, would result in an average of 35 per cent faster typing speeds, would allow for more accurate output and result in the typist to be less fatigued due to their fingers having to move a shorter distance per word. Although they presented a potential successor to the QWERTY keyboard, what Dvorak did not answer was the question: “Is this the most efficient keyboard layout?"

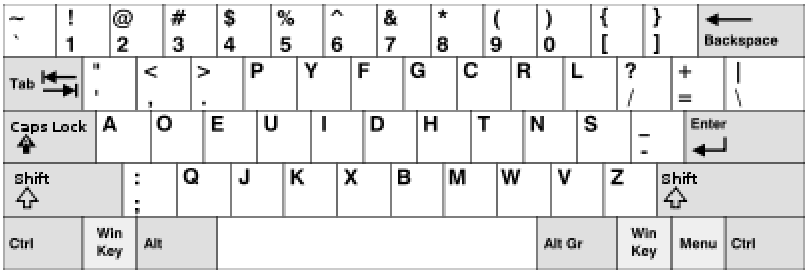


Figure 4.2. Present day version of Dvorak layout

The Dvorak Simplified Keyboard (DSK) is accepted as the best-known alternative to QWERTY layout, and also known as the American Simplified Keyboard, ASK layout. It was named after its inventor, Dr. August Dvorak, not the key order. There are also numerous adaptations for languages other than English, and single-handed variants. Dr. Dvorak's original layout had the numerals rearranged, but the present-day layout has them in numerical order as in Figure 4.2.

4.1.3. F Layout

F layout was designed for Turkish language which uses the Latin alphabet. It was designed in 1955 by İhsan Yener as in Figure 4.3. During its design, the Turkish Language Association (TDK) investigated letter frequencies in Turkish and used this statistical basis to design the Turkish-F keyboard. It provides a balanced distribution of typing effort between the hands – 49 per cent for the left hand and 51 per cent for the right. Besides the Turkish F keyboard, a modified QWERTY keyboard is used on most computers in Turkey. F keyboards are most commonly used by an older generation who learned this layout, because it is no longer taught in schools. Especially, companies and government agencies having old technology still use this layout.

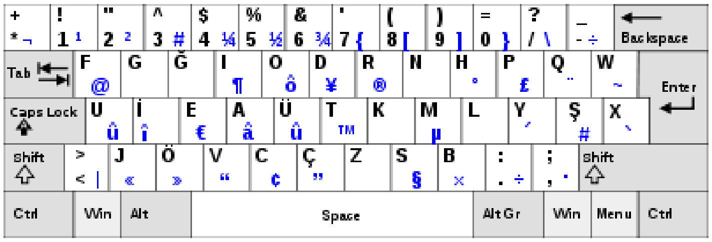


Figure 4.3. Turkish F keyboard layout

It is commonly believed and statistically proved that F layout users type an average Turkish text in shorter time than QWERTY layout users. Turkish students using F layout achieved great successes in worldwide typing competitions.

4.1.4. Colemak Layout

Colemak is another public domain alternative to QWERTY layout that has been designed specifically to be easy to learn for existing QWERTY typists while at the same time being tightly optimized for touch typing and overcoming some of the problems with Dvorak. It was developed by Shai Coleman and the name “Colemak” was inspired by the Linux naming idea. In January 2009, its inventor estimated that it had around 3000 users worldwide. This layout is used especially for expert users. This layout is improved by swapping some letters in pair wise as in Figure 4.4.

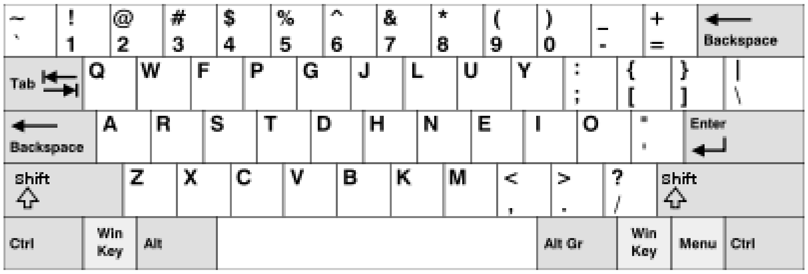


Figure 4.4. The Colemak keyboard layout

Shai Coleman explains the necessity of a new layout by listing the disadvantages of QWERTY layout as follows:

- It places very rare letters in the best positions, so the user’s fingers have to move a lot more
- It suffers from a high same finger ratio that slows down typing and increases strain
- It allows for very long sequences of letters with the same hand (e.g. typing the word "sweaterdresses")
- It was designed to prevent the keys from sticking, without any consideration to ergonomic or efficiency aspects
- It was designed so the word "typewriter" could be typed on the top row to ease demonstrations, for a better marketing technique in 1860’s
- It suffers from an extremely high ratio of home-row-jumping sequences (e.g. typing the word "minimum")

Shai Coleman also argues that Dvorak layout, which was also designed as an alternative to QWERTY, will not solve the problems. He defends that shifting from QWERTY to Dvorak will take huge amount of cost and time, since their character arrangements are very different. He lists the disadvantages of Dvorak layout as:

- The main problem with Dvorak is that it's too difficult and frustrating to learn for existing QWERTY typists because it's so different from QWERTY. Colemak has been designed to be easy to learn.
- Placing 'L' on the QWERTY 'P' position causes excessive strain on the right pinky. Colemak doesn't place frequent letters where the pinkies stretch.
- 'F' is on the QWERTY 'Y' position which is a difficult stretch on normal keyboards.
- 'I' is very frequent but isn't on the home position.
- 'R' is very frequent but isn't on the home row.
- It is significantly lopsided so that the right hand does too much work.

- It's not comfortable to use Ctrl-Z/X/C/V shortcuts with the left hand while holding the mouse with the right hand. Colemak conserves those shortcuts in their QWERTY positions.
- Even though the design principles are sound, the implementation isn't optimal because it was designed without the aid of computers.
- 'L' and 'S' form a frequent same-finger digraph on the right pinky. Same-finger for the pinky is very rare in Colemak. In particular, UNIX commands such as 'ls -l' are very uncomfortable to type.
- Some punctuation (in particular the curly/square brackets) is less comfortable to type on Dvorak. This affects mainly programmers and advanced UNIX users

4.1.5. AZERTY Layout

The AZERTY layout is used in especially French speaking countries such as France, Belgium and some African countries and shown in Figure 4.5. Simple heuristics were used to develop this layout in order to increase the typing speed by using French monographs. Also, easing access of letters and several accented letters, such as é, à and ô, commonly used in French, was another criterion while designing this layout. It differs from the QWERTY layout thus:

- A and Q are swapped,
- Z and W are swapped,
- M is moved to the right of L (where colon/semicolon is on a US keyboard or where the letter “Ş” is on a standard Turkish QWERTY layout),

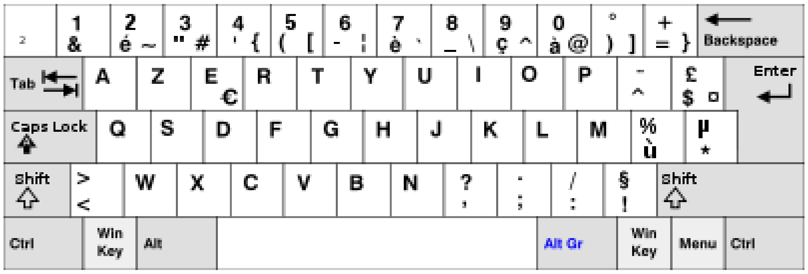


Figure 4.5. The AZERTY keyboard layout for French

4.1.6. Walker's Evolved Keyboard Layout

This layout was just a layout produced by an academic research of Christopher P Walker. He used a simple evaluation algorithm having the stages; initialization, evaluation, selection, reproduction, mutation and computation. The top 1,000 used words in the English language were used to evaluate the individuals. This should optimize the keyboard for the most common used words, which would result in faster typing overall (Walker, 2003).

Event	Points Awarded
Using the Same Finger	-2,00
Using the Same Hand, Different Fingers	1,00
Different Hands	3,00
Using a Key on Home Row	2,00
Using the Pinky Not on Home Row	-1,50
Using the First Finger	0,75
Using the Middle Finger	0.50
Using the Ring Finger	-0,10
Using the Pinky Finger	-0,20

Figure 4.6. Event scorings for Walker’s Evolved Keyboard layout

The main aspect of his application was the existence of a multi criteria decision making problem. He scored the following ergonomics criteria and formulated his objective function as a linear combination of these scores. He calculated the scores of events, shown in Figure 4.6., while typing these top 1,000 used words and finally designed the layout shown in Figure 4.7.

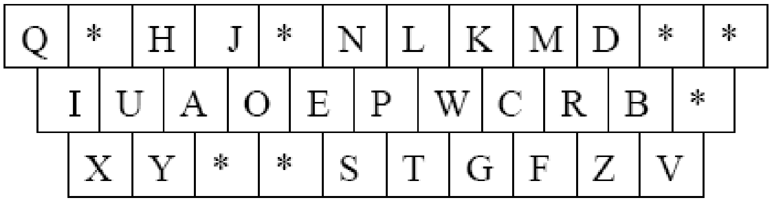


Figure 4.7. Walker’s Evolved Keyboard layout

4.1.7. Other Physical Keyboard Layouts

In literature, there have been a number of physical keyboard layouts, which are usually not used in daily life. For instance, Eggers *et al*. (2003) designed an arrangement for the so-called ergonomic physical keyboard layout (ECP) and claimed that the design is better than the QWERTY keyboard by more than 50 per cent and also slightly better than the Dvorak keyboard by about 1.9 per cent. Prajapati (2008) used these six ergonomics criteria mentioned so far to compare the three existing keyboard layouts (Traditional Layout, Romanized Layout and Devanagari Layout) of Nepali language in the project of One Laptop per Child in Nepal.

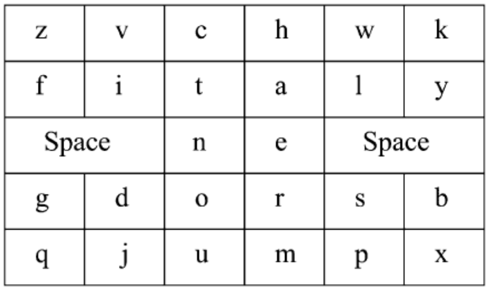
4.2. Stylus Keyboard Layouts

With the impact of revolutions in mobile device technology, the interest of keyboard layout optimization shifted to stylus keyboard layout optimization, instead of physical keyboard layout optimization. For this reason, recent academic works dealing keyboard optimization propose new stylus keyboard layouts.

4.2.1. FITALY Layout

The FITALY keyboard is a USA stylus keyboard patented by Text Ware solutions. It has been designed for English on the basis of the corresponding words frequency (Dell'Amico *et al.*, 2009).

The FITALY keyboard, which is 5 x 6 matrix for the alphabetical part of the keys, has been designed based on the frequencies of the characters in the English language and has been based on the Brown Corpus for the English language. The most frequent characters are located in the middle of the keyboard, as in Figure 4.8. The lower frequency characters are placed towards the corners of the keyboard. The developers of the FITALY layout argue that single-finger typing requires a different keyboard layout since the traditional QWERTY keyboard was designed for typing with ten fingers. Typically, a professional typist maintains fingers on the so-called home keys (the keys "a-s-d-f" for the left hand and "j-k-l"; for the right hand) and typing letters will either be on this home row or involve a move to some adjacent keys, one row below the home row, or one or two rows above. Consequently, there is no significant finger travel.



z	v	c	h	w	k
f	i	t	a	l	y
Space		n	e	Space	
g	d	o	r	s	b
q	j	u	m	p	x

Figure 4.8. The FITALY layout

It is also stated that the situation is quite different on a pen computer or a computer with a touch screen. In these situations (and also on miniature keyboards found in some personal digital assistants), input is done with a single finger, or with an electronic pen or some equivalent device. The same finger has then to travel to successive keys one by one and this ends up involving considerable finger (or pen) travel. They illustrate the issue of hand travel besides finger travel as when typing the word “*transpose*”, the move between the letters “*a”* and “*n”* is too large to be accomplished just by finger travel. It requires a full movement of the hand, which is much less precise than a finger movement and significantly decreases input speed. The same is true for the transition “ns” and also “sp” and “os”. They defend that the QWERTY layout is very inefficient for stylus keyboards due to the reasons mentioned above. It forces the pen to wide left-and-right sweeps like the head of an old dot-matrix printer. This inefficiency comes from having extrapolated such keyboards to a context where the requirements are quite different.

While designing FITALY, it was accepted that the most important characteristic of a single-finger keyboard layout should be the square formed region for the alphabetical part of the keyboard. Assuming an initial position of the pen (or finger) at the centre, this shape results in small travels to the other keys. In addition, key placement was conditioned by frequencies of letters in the English language. These are indicated for each letter (in occurrences per 10,000 letters) in Figure 4.9, based on the Brown Corpus for the English language.

Z 20	V 77	C 230	H 415	W 138	K 49
F 176	I 551	T 701	A 615	L 319	Y 133
(space) 1741		N 550	E 976	(space) 1741	
G 147	D 305	O 590	R 497	S 497	B 110
Q 20	J 20	U 210	M 187	P 150	X 20

Figure 4.9. Letter frequencies based on Brown Corpus for FITALY

As seen in Figure 4.9, the most frequent letters are quite closely grouped in the FITALY keyboard. The six letters- T, A, N, E, O and R - in the centre area represent a combined frequency of 39.3 per cent, and, together with the space keys, of 56.7 per cent. So, more than half of all keystrokes will occur in this area. Then, the next most frequent 4 letters -

I, L, D, and S – were located to the nearest keys to this centre area, representing a combined frequency of 73,4 per cent. Subsequently, next coming most frequent letters – C, H, U and M – were located to the next 4 most appropriate keys. So far located 14 letters with space key represent over 84 per cent of all keystrokes. Remaining 12 letters were assigned to keys according to the frequency of letters in Brown Corpus and the distances of the keys to the centre point. In other words, using these key placements means that more than half of the keys will be one-key-away from the centre keys N and E and 84 per cent of all keys will be two-keys-away at most. In comparison, these distances range up to 5-keys away from the centre keys on the QWERTY keyboard. Similarly, the maximum distance between two letters is 5 on the FITALY keyboard. The difference is most striking on the six most frequent letters - T, A, N, E, O and R -: the largest travel between any two keys is 2 on the FITALY keyboard, compared to 8 for letters “A” and “O” on the QWERTY keyboard.

Figure 4.10. The FITALY layout on PALM screen

However, FITALY layout does not take the relationship between the letters into account. The unique criterion for assignment of a letter to a key is the frequency of the letter in English (*or* monograph frequency) and the distance of the key to the centre of the layout.

4.2.2. OPTI Layout

OPTI layout is one of the optimized virtual keyboard layouts for the English language which was improved by MacKenzie and Zhang (1999), and shown in Figure 4.11. This keyboard layout was optimized for speed by using trial and error method with the help of Fitts' law and character and digraph frequencies in English. Fitts' law gives a function for computing the tapping time given the length of the movement needed and the width of the target thus enabling a researcher to compute a prediction for the upper bound of user performance given the keyboard layout. Trial and error method was needed to generate the keyboard layout. They had improved step by step their layout by swapping the letters (MacKenzie and Zhang, 1999).

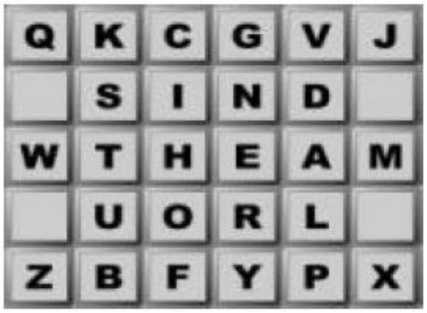
The improved OPTI design (OPTI II) by Zhai et al (2000) involves a mixture of monograph and digraph frequencies: First, 10 most frequent letters are placed in the most centre location of the keyboard. Then, the 10 most frequent digraphs are assigned to the remaining relatively centrally located keys. The placement was all done by trial and error.

Figure 4.11. The Mackenzie’s and Zhang’s OPTI layout

MacKenzie and Zhang (1999) argue that OPTI layout is over 35 per cent faster than QWERTY layout and 5 per cent faster for data entry compared with FITALY. Zhai and Kristensson (2005) agree with them by applying Fitts’s Law to FITALY and OPTI layouts. They defend that OPTI is more efficient than FITALY and QWERTY by ratios of 1.04 and 25.15 per cent, respectively.

OPTI layout has 4 space bars which lie on four parts, which increases the typing speed when thought that blank character is involved over 25 per cent of all digraphs in English. Zhai and friends (2000) later designed a newer version of OPTI layout.

They later made a further improved 5x6 layout which is called OPTI II or Improved OPTI Layout, as shown in Figure 4.12.



Q	K	C	G	V	J
	S	I	N	D	
W	T	H	E	A	M
	U	O	R	L	
Z	B	F	Y	P	X

Figure 4.12. Improved OPTI layout (OPTI II Layout)

4.2.3. Hooke’s Layout

Hooke’s layout is the first stylus keyboard whose letters are not square, but circle. It was designed by dynamic simulation method. Zhai et al. (2000) tell this method as: “Imagine a spring connecting every pair of the 27 keys whose initial positions were randomly placed with spaces between the keys. The elasticity of the springs, when turned on, was proportional to the transitional probability between the two keys so that keys with higher transitional probability would be pulled together with greater force. In addition, there is viscous friction between the circle shaped keys and between the key surface and the table surfaces. The steady state, observed when all keys are pulled together, forms a candidate virtual keyboard design”. This method is depicted in Figure 4.13.

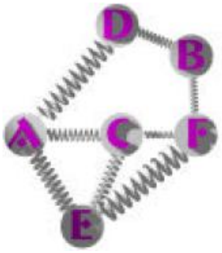


Figure 4.13. Dynamic simulation model for Hooke’s layout

“When not satisfactory, the layout could be ‘stretched’ out to serve as another initial state for the next iteration of the same process. The iteration was repeated until a satisfactory layout is formed. To capture the gist of the spring simulation technique, we call the best design achieved through this method Hooke’s keyboard after Hooke’s Law” (Zhai, *et al.*, 2000). The final Hooke’s layout is shown in Figure 4.14.

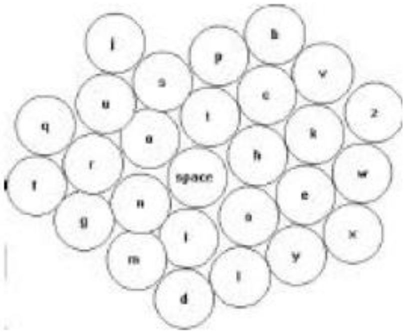


Figure 4.14. The Hooke’s layout

4.2.4. Metropolis Layout

Metropolis layout is well known due to the shape of its keys. The keys in Metropolis layout are lined as a honey comb. The centres of the keys are thought to be in minimum distances in this layout format.

This layout is designed by using Fitts's law and Metropolis algorithm. Zhai *et al.* (2000) tell the stages of Metropolis method as: "The idea of minimizing energy, or tension, in the keyboard layout brought us to explore a better known optimization method - the Metropolis algorithm. The Metropolis algorithm is a Monte Carlo method widely used in searching for the minimum energy state in statistical physics. We define the problem of designing a high performance keyboard is equivalent to searching for the structure of a molecule at a stable low energy state. Applying this approach, we designed and implemented a software system that did a "random walk" in the virtual keyboard design space. In each step of the walk, the algorithm picked a key and moved it in a random direction by a random amount to reach a new configuration".

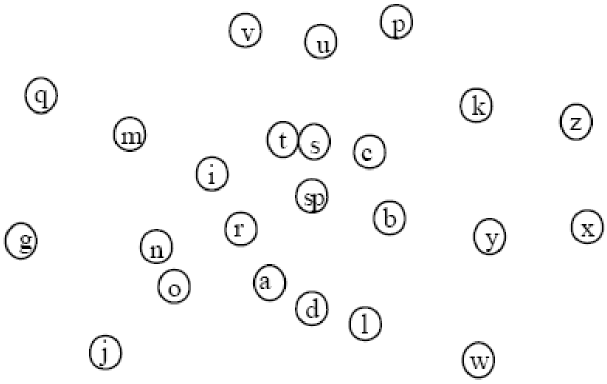


Figure 4.15. Early stages of Metropolis layout

"Again, the initial state where the random walk starts from had a significant impact on the search process. An existing good layout stretched over a larger space was used as an initial state. In addition to the automatic random walk process itself, we also applied interactive "annealing" as commonly used in the Metropolis searching process. The annealing process involved bringing "temperature" through several up and down cycles. When temperature was brought up, the system had a higher probability of moving upwards in energy and jumping out of local minima. When temperature was brought down, the system formed down to a lower energy level. This annealing process was repeated until a sufficiently efficient keyboard layout was found” (Zhai, *et.al*, 2000). Final Metropolis layout is shown in Figure 4.16.

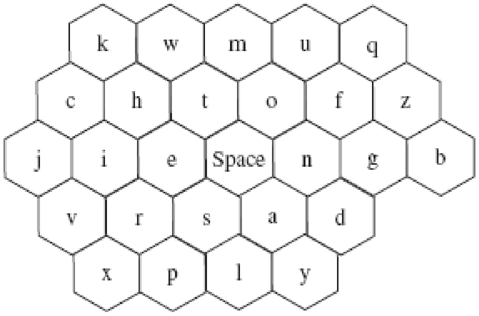


Figure 4.16. The Metropolis layout

To compare the design methods, it should be stated that The Hooke keyboard was based on minimizing the distance between likely characters using a greedy algorithm. The Metropolis keyboard on the other hand uses a random-walk strategy rather than a greedy algorithm.

4.2.5. The Chubon Layout

Chubon layout was also designed by a trial and error method. It does not have a regular standard shape. It seems that this layout is far away to get the attention due to its asymmetric shape as shown in Figure 4.17. Its estimated performance is lower than OPTI and FITALY layouts, but still higher than QWERTY stylus keyboard.

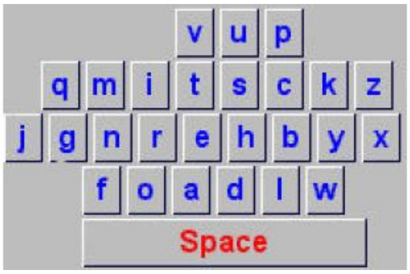
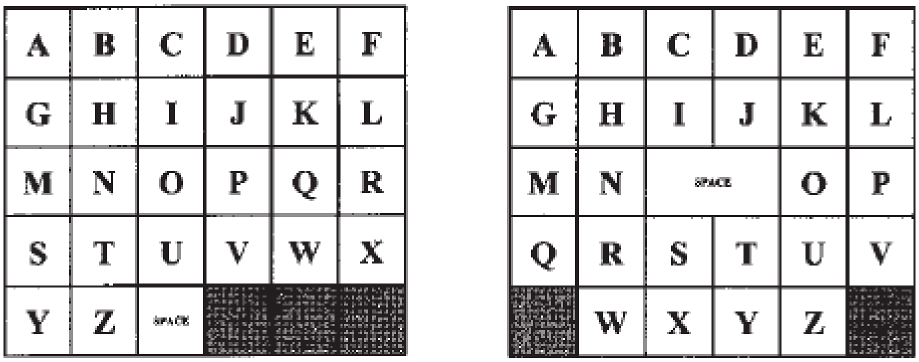


Figure 4.17. The Chubon layout

4.2.6. Other Stylus Keyboard Layouts

There are a few more stylus keyboard layouts which are either only an academic study or not used by device manufacturers. Classical ABC and ABC-centre layouts can be listed in this category. ABC layouts are the natural results of alphabetical order of letters in a language. ABC-centre layout, which has the space bar located in the centre, is a little more effective than ABC layout.



A	B	C	D	E	F
G	H	I	J	K	L
M	N	O	P	Q	R
S	T	U	V	W	X
Y	Z	SPACE	▨	▨	▨

A	B	C	D	E	F
G	H	I	J	K	L
M	N	SPACE		O	P
Q	R	S	T	U	V
▨	W	X	Y	Z	▨

Figure 4.18. Classical ABC and ABC-centre layouts

Yanzhi *et al*. (2000) proposed two stylus keyboard layouts, one for square shape and one for longitudinal shape, called YLAROF (Figure 4.19.) and RANI (Figure 4.20.), respectively.

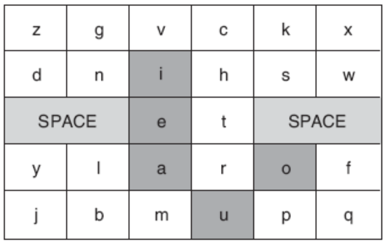


Figure 4.19. The YLAROF layout

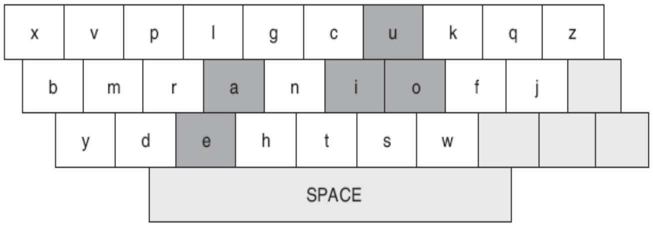


Figure 4.20. The RANI layout

Zhai and Kristensson (2005) designed another stylus keyboard, named ATOMIK layout, by using the connectivity index of the letters in most frequent letters of a series of English corpus.

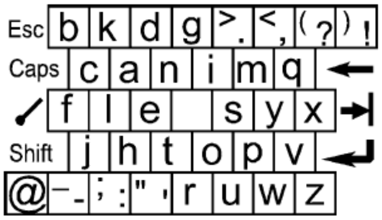


Figure 4.21. The ATOMIK layout

4.2.7. Comparison of Stylus Keyboard Layouts

In literature, there exist a number of studies that compare stylus keyboard layouts. The comparisons generally are based on either subject tests or computer aided simulations which predict the expected values of expert users.

To speak roughly, it is tested that square shaped layouts are more effective in speed than longitudinal layouts. Also, the layouts designed just for stylus typing (*OPTI*, *FITALY*...) are more effective than the revised 10-finger layouts (*QWERTY*, *Dvorak*...).

Researchers used wpm (word per minute) or cps (character per second) units while comparing the layouts for typing speeds. It is commonly accepted that while transforming cps to wpm, the words are involving 5 characters including blanks. Hence, cps is multiplied by 12 to transform to wpm.

Yanzhi *et al*. (2006) first tried to estimate the typing speeds of expert users by the help of Fitts' law. The results were published in wpm units.

Table 4.1. Expected typing speeds for expert users (wpm) by (Yanzhi, 2006)

RANI	40.11
YLAROF	43.44
QWERTY	31.14
Dvorak	28.61
FITALY	42.61
Metropolis	42.79

Yanzhi *et al*. (2000), made also a speed test in laboratory conditions using 20 subjects. They tried to compare the average typing speed of these subjects using RANI, YLAROF, QWERTY, Dvorak, FITALY and Metropolis stylus keyboard layouts. The subjects typed the same texts on every layout and the average completion times are recorded. The texts were in English and randomly selected from different areas including history, business, art, news, politics, religion and world.

According to them, YLAROF is the lowest recorded times with Metropolis and Fitaly being second and third, respectively. Another interesting point is that Dvorak has lower typing speeds when compared with QWERTY. Hence, QWERTY can be said to be more effective than Dvorak in stylus keyboard typing.

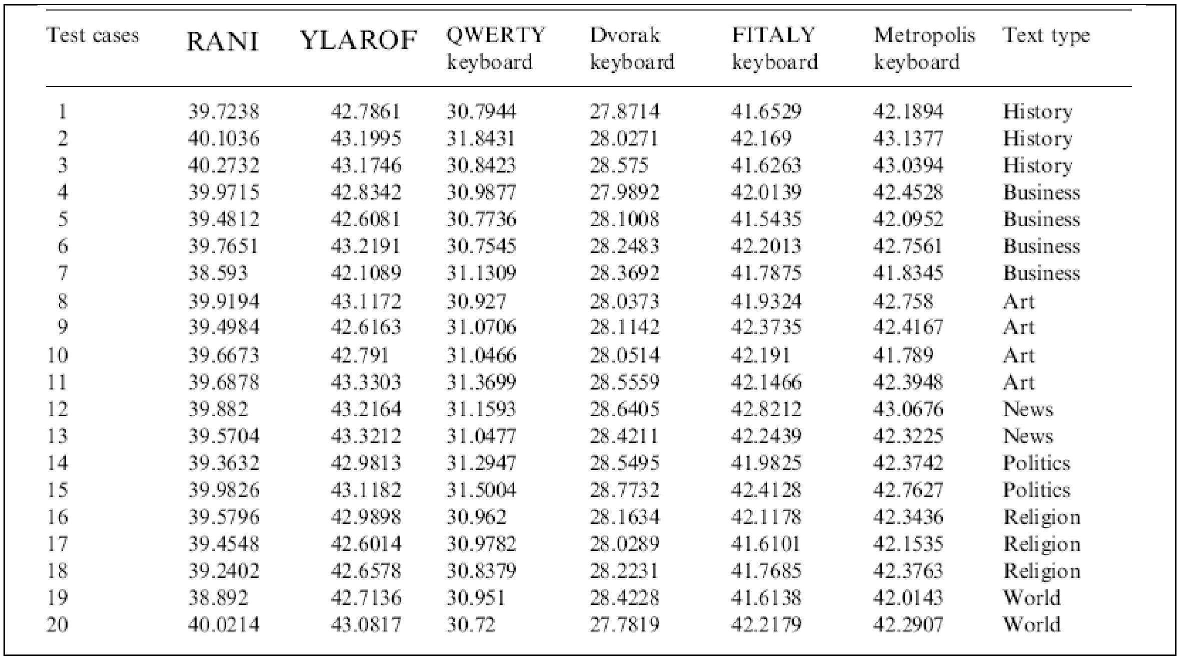
MacKenzie and Zhang compared (1999) OPTI layout with QWERTY. They tried to plot the learning curve of OPTI layout users. The method was as follows: “The experiment was a 2 x 20 within-subjects factorial design. The two factors were:

- Keyboard layout {QWERTY, OPTI}
- Session {20 sessions}

Each session lasted about 45 minutes and was divided into two 20-22 minute periods. One of the two layouts was assigned in each half-session period in alternating order from session to session. The order of the conditions was balanced between participants to reduce interactions.

Each half-session contained several blocks of trials. The number of blocks for each half-session period was controlled such that as many blocks as possible were collected within the allotted time. Therefore, in the early sessions, fewer blocks (5 to 6) were administered than in later sessions (9 to 11). A five-minute break was allowed between the two half-sessions.

Table 4.2. Typing speed records of subjects (Yanzhi)



Test cases	RANI	YLAROF	QWERTY keyboard	Dvorak keyboard	FITALY keyboard	Metropolis keyboard	Text type
1	39.7238	42.7861	30.7944	27.8714	41.6529	42.1894	History
2	40.1036	43.1995	31.8431	28.0271	42.169	43.1377	History
3	40.2732	43.1746	30.8423	28.575	41.6263	43.0394	History
4	39.9715	42.8342	30.9877	27.9892	42.0139	42.4528	Business
5	39.4812	42.6081	30.7736	28.1008	41.5435	42.0952	Business
6	39.7651	43.2191	30.7545	28.2483	42.2013	42.7561	Business
7	38.593	42.1089	31.1309	28.3692	41.7875	41.8345	Business
8	39.9194	43.1172	30.927	28.0373	41.9324	42.758	Art
9	39.4984	42.6163	31.0706	28.1142	42.3735	42.4167	Art
10	39.6673	42.791	31.0466	28.0514	42.191	41.789	Art
11	39.6878	43.3303	31.3699	28.5559	42.1466	42.3948	Art
12	39.882	43.2164	31.1593	28.6405	42.8212	43.0676	News
13	39.5704	43.3212	31.0477	28.4211	42.2439	42.3225	News
14	39.3632	42.9813	31.2947	28.5495	41.9825	42.3742	Politics
15	39.9826	43.1182	31.5004	28.7732	42.4128	42.7627	Politics
16	39.5796	42.9898	30.962	28.1634	42.1178	42.3436	Religion
17	39.4548	42.6014	30.9782	28.0289	41.6101	42.1535	Religion
18	39.2402	42.6578	30.8379	28.2231	41.7685	42.3763	Religion
19	38.892	42.7136	30.951	28.4228	41.6138	42.0143	World
20	40.0214	43.0817	30.72	27.7819	42.2179	42.2907	World

Each block contained 10 text phrases of about 25 characters each. These 10 phrases were randomly selected from a source file of 70 phrases. Phrases were not repeated within blocks but repeats were allowed from block to block. The phrases were chosen to be representative of English and easy to remember" (MacKenzie and Zhang, 1999).

The results above were as expected. That is, the OPTI layout faired poorly initially (17 wpm) in comparison with the QWERTY layout (28 wpm). With practice, however, the OPTI layout eventually out-performed the QWERTY layout.

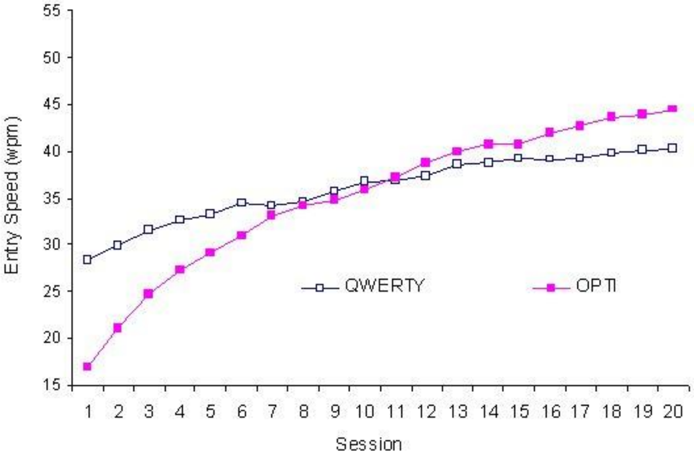


Figure 4.22. Typing speed by session and layout (MacKenzie and Zhang, 1999)

They had also analysed the error rates for both layouts session by session. They defined the error as any difference typed by the subject from the given character. The error rates ranged from 2.07 per cent for OPTI and 3.21 per cent for QWERTY on the first session to 4.18 per cent for OPTI and 4.84 per cent for QWERTY on the 20th session.

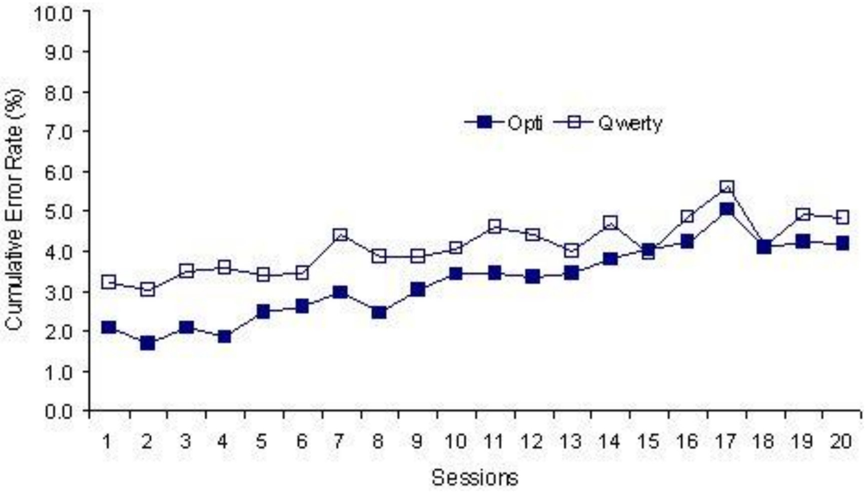


Figure 4.23. Error rates by session and layout (MacKenzie and Zhang)

Zhai and Kristensson (2005) also estimated the expected top user's speed in *wpm* by using again Fitts' law with the parameters $a = 0.083$ and $b = 0.127$ s. They concluded that Metropolis is the fastest stylus layout among all as shown on Table 4.3.

Table 4.3. Expected typing speeds for expert users, wpm (Zhai and Kristensson, 2005)

Keyboard	**Typing speed (*wpm*)**
OPTI	42.8
QWERTY	34.2
ATOMIK	45.3
FITALY	41.2
Metropolis	46.6

5. QUADRATIC ASSIGNMENT PROBLEM (QAP)

In this chapter; quadratic assignment problem (QAP) which is one of the solution methods for stylus keyboard layout problem, is described in detail.

5.1. Introduction

“What is the optimal way to wire a computer backboard? How are the locations of clinics within a hospital decided? Where to locate warehouses of nationwide logistics firms? What possible linkages could there be between these problems? Most would agree that at first glance, they are seemingly unrelated beyond the fact that all are decision problems. One might even propose that such decisions are made arbitrarily. However, it is the solution to these and countless other problems that contains the key to their correlation. They are all modelled by one of the most challenging problems in combinatorial optimization. This problem has been a focus of researchers for over four decades; it is known as the Quadratic Assignment Problem” (Commander, 2005).

C. Commander (2005) depicts the history and motivation of the Quadratic Assignment Problem (QAP) in his survey by saying that: “QAP was originally introduced in 1957 by Tjalling C. Koopmans and Martin Beckman who were trying to model a facilities location problem. Since then, it has been among the most studied problems in all of combinatorial optimization. Many scientists including mathematicians, computer scientists, operations research analysts, and economists have used the QAP to model a variety of optimization problems. For over four decades, scientists have been studying the QAP, and have made significant discoveries in the study of assignment problems. Over the years, the QAP has been used to model such things as hospital, computer backboard design, scheduling problems, and of course, location problems.

In other words, the quadratic assignment problem can be seen as a facility location problem. For example, assume that there are n facilities and n locations in a real life case. The pair wise flows between facilities and the distances between each pair of locations are known. The problem can be defined as assigning a facility to exactly one location such that the distance times flow is minimized (Vliet, 2009).

5.2. Brief Mathematical Definition

Mathematically, the problem can be formulated by defining two $n\ x\ n$ matrices: a flow matrix F whose $(i,j)-th$ element represents the flow from facility i to facility j; and a distance matrix D whose $(i,j)-th$ element represents the distance between location i and location j. An assignment is represented by the vector p which is a permutation of the numbers $\{1,2,3,\dots..,n\}$. So, $p(i)$ is the location to which facility i is assigned, and likewise $p(j)$ is the location to which facility j is assigned. Let $\boldsymbol{\pi}$ be the set of all possible permutations that can be created by the numbers $\{1,2,3,\dots..,n\}$. With these definitions, the Quadratic Assignment Problem can be written as

$$\min_{p\in\pi}\sum_{i=1}^{n}\sum_{j=1}^{n}f_{i,j}*d_{p(i),p(j)} \tag{5.1}$$

In other words, the QAP is trying to find a permutation p of the set of facilities and locations which minimizes the objective function $F*D$ where $F=[f_{ij}]\ and\ D=[d_{p(i),p(j)}]$.

5.3. Differences between Quadratic Assignment and Linear Assignment

A commonly used intuitive introduction to the linear assignment (LAP) as used in literature, involves the assignment of n people to n jobs. For each job assignment, there is a related cost, c_{ij}, of assigning person i to job j. The objective is to assign each person to one and only one job in such a manner that minimizes the sum of each assignment cost, i.e., the total cost.

Mathematically, the Linear Assignment Problem can be formulated as follows:

$$\min_{\pi \epsilon S_n} \sum_{i=1}^{n} c_{i,\pi(i)} \tag{5.2}$$

where S_n is the set of all permutations of $\{1,2,3, \ldots.,n\}$. and $j = \pi\ (i)$ is the job assignment of person i. Notice that each set of assignments is a permutation of a set of n integers; hence, there are n! distinct ways in which n jobs can be assigned to n people. Quadratic Assignment Problem is more complicated generalization of the Linear Assignment Problem, having Flow Matrix like LAP; but also distance matrix which LAP does not (Commander, 2005).

As with the LAP, there are n! permutations from which to choose the optimal assignment. However, the reader should be made aware that there is a key difference between these two problems which makes the QAP considerably more difficult to solve. Unlike the LAP in which the assignment of job j to person i was made independently of the assignments of the other employees, with the QAP the assignments are not independent. That is, when considering an assignment of person i to location k, one must consider the assignments of all other people who have some nonzero affinity for person i (Commander, 2005).

The quadratic assignment problem (QAP) is an interesting "combinatorial optimization" problem. Koopmans and Beckmann introduced this problem in 1957 as an economic location problem. But nowadays, the QAP has also a lot of applications in other fields and there are many real life problems which can be modelled by QAP's. Moreover, many other combinatorial problems, such as the travelling salesman problem (TSP), the bin-packing problem, and the max-clique problem can be formulated as a QAP (Loiola, 2007). Furthermore, Vliet (2009) states that the QAP is one of the great challenges in combinatorial optimization. This is because the problem is very hard to solve and also hard to approximate.

Since its first formulation, the QAP has been drawing researchers' attention worldwide, not only because of its practical and theoretical importance, but also because of its complexity. The QAP is one of the most difficult combinatorial optimization problems. In general, instances of size $n > 30$ cannot be solved in reasonable (polynomial) time by even super parallel processors. For ordinary home or school computers, n decreases till ten even nine. Sahni and Gonzales (1976) showed that QAP is NP-hard and that, unless $P = NP$, it is not possible to find an f-approximation algorithm, for a constant f (Sahni and Gonzales, 1976). Such results are valid even when flows and distances appear as symmetric coefficient matrices. Due to its high computation complexity, the QAP has been chosen as the first major test application for the GRIBB project (great international branch-and-bound search). This project is seeking to establish a software library for solving a large class of parallel search problems by the use of numerous computers around the world accessed via Internet (Loiola, 2007).

5.4. QAP Formulations

5.4.1. Integer Linear Programming Formulations (IP *or* ILP)

First, we present the QAP as a Boolean program (*0-1 program*) followed by a linear programming problem, where the binary constraints are relaxed. The Boolean formulation was initially proposed by Koopmans and Beckmann in 1957 and later used in several works (Loiola, 2007).

We consider f_{ij} the flow between facilities i and j, and d_{kp} the distance between locations k and p. It is our goal to calculate:

$$\min \sum_{i,j=1}^{n} \sum_{k,p=1}^{n} f_{i,j} * d_{k,p} * x_{i,k} * x_{j,p} \tag{5.3}$$

$$s.t.\ for\ all\ j;\ \sum_{i=1}^{n} x_{i,j} = 1 \qquad 1 \le j \le n, \tag{5.4}$$

$$for\ all\ i;\ \sum_{j=1}^{n} x_{i,j} = 1 \qquad 1 \le i \le n, \tag{5.5}$$

$$x_{i,j} \in \{0,1\} \qquad 1 \le i,j \le n \tag{5.6}$$

5.4.2. Mixed Integer Linear Programming (MILP) Formulations

The QAP, as a mixed integer programming formulation, is found in the literature in different forms. All of them replace the quadratic terms by linear terms. For example, Lawler used n^4 variables, to linearize the quadratic terms of Integer Linear Programming formulation as follows (Loiola, 2007);

$$c_{i,j,k,p} = f_{i,j} * d_{k,p} \tag{5.7}$$

$$y_{i,j,k,p} = x_{i,k} * x_{j,p} \tag{5.8}$$

$$1 \le i,j,k,p \le n \tag{5.9}$$

In general, QAP linearizations based on MILP models present a huge number of variables and constraints, a fact that makes this approach unpopular. However, these linearizations, together with some constraint relaxations, lead to the achievement of much improved lower bounds for the optimal solution, by using branch-and-bound algorithm for example.

5.4.3. Formulations by Permutations

Taking a simple approach, the pair wise allocation of facility costs to adjacent locations is proportional to flows and to distances between them. The QAP formulation that arises from this proportionality and uses the permutation concept can be found in numerous articles, especially in the recent ones (Loiola, 2007).

Let S_n be the set of all permutations with n elements and $\pi \in S_n$. Consider $f_{i,j}$ the flows between facilities i and j and $d_{\pi(i),\pi(j)}$ the distances between locations $\pi(i) and\ \pi(j)$. If each permutation π represents an allocation of facilities to locations, the problem expression becomes:

$$\min_{\pi \in S_n} \sum_{i,j=1}^{n} f_{i,j} * d_{\pi(i),\pi(j)} \tag{5.10}$$

This formulation is equivalent to the first one presented in Integer Linear Programming formulation; since the constraints in ILP formulation involving $\boldsymbol{x_{i,j}}$ define the permutation matrices $X = [x_{i,j}]$ related to S_n elements as in formulation by permutation (Loiola, 2007).

Here,

$$for\ all\ 1 \leq i,j \leq n; \quad x_{i,j} = \begin{cases} 1, & if\ \pi(i) = j; \\ 0, & if\ \pi(i) \neq j. \end{cases} \tag{5.11}$$

5.4.4. Trace Formulation

“This formulation is supported by linear algebra and exploits the trace function (the sum of the matrix main diagonal elements) in order to determine QAP lower bounds for the cost. This approach allows for the application of spectral theory, which makes possible the use of semi-definite programming to the QAP. Edwards stated the trace formulation as follows where again F is the flow matrix of facilities and D is the distance matrix of the locations” (Loiola, 2007).

$$\min_{X \in S_n} \; tr(F * X * D * X^t) \tag{5.12}$$

5.4.5. Graph Formulation

Let us consider two undirected weighted complete graphs, the first one having its edges associated to flows and the second one, to distances. The QAP can be thought as the problem of finding an optimal allocation of the vertices of one graph on those of the other. In this formulation the solution costs are given as the sum of products of corresponding edge weights. This formulation is the least used one among all formulation types of QAP since the computational difficulties and limited application areas (Loiola, 2007).

5.5. QAP Related Problems

5.5.1. A Brief List of QAP Related Problems:

The most classical QAP-related problem is, obviously, the Linear Assignment Problem (LAP), which is polynomial and easily solved by the Hungarian method. Several different presentations of this problem can be found in the literature. A wide range of QAP theoretical studies involve several related quadratic problems. Loiola listed these problems as follows (Loiola, 2007):

- Linear Assignment Problem (LAP)
- The three-index assignment problem (3-dimensional AP or 3AP)
- The quadratic bottleneck assignment problem (QBAP)
- The biquadratic assignment problem (BiQAP)
- The quadratic 3-dimensional assignment problem (Q3AP)
- The quadratic semi-assignment problem (QSAP)
- The multiobjective QAP (mQAP)

5.5.2. The Quadratic Bottleneck Assignment Problem (QBAP)

It is considered that QBAP is a variation of QAP with applications to backboard wiring. In that work, a placement algorithm was presented for the optimal connection of n elements in individual locations so that the length wire needed to connect two elements is minimized. The basic claim of the paper is: “the optimal weighted-wire-length equals the least among the maximum-wire-length norms” (Loiola, 2007).

This concept arises from the principle that it may be better to minimize the largest cost in a problem, than to minimize the overall cost.

The QBAP general program is obtained from the QAP formulation by substituting the maximum operation in the objective function for the sums, which suggests the term bottleneck function:

$$\min_{\pi \in S_n} \max\left\{f_{i,j} * d_{\pi(i),\pi(j)} \quad : 1 \leq i,j \leq n\right\} \tag{5.13}$$

5.6. Solution Methods of Quadratic Assignment Problems

The solution methods for QAP can be classified into two groups as follows;

- Exact Algorithms
- Heuristic Algorithms

5.6.1. Exact Algorithms

Exact algorithms do guarantee the global optimum solution of quadratic assignment problems. Unfortunately, these methods generally try all combinations in the solution set. Hence, they need a great amount of time, sometimes even years.

There are three main exact methods used to find the global optimal solution for a given QAP:

- dynamic programming
- cutting plane techniques
- branch and bound procedures

Research has shown that branch and bound procedure is the most successful among exact algorithms for solving QAP. Even still, despite the huge revolutions in computer and processor technology, due to the overwhelming complexity of QAP, most problems with their sizes greater than $n = 30$ remain nearly intractable by exact algorithms. Since branch and bound procedures are generally the most helpful for solving QAPs, many educational and commercial software packages utilizes from branch-and-bound algorithm directly or a slightly different form of it (Yongzong and Hazohao, 2006).

In recent years, procedures that combine branch-and-bound techniques with parallel implementation are being widely used. Due to them, the best results for the QAP are being achieved. Yet, it is important to observe that the success for the instances of bigger sizes is also related to the hardware technological improvements (Loiola, 2007).

In typical branch and bound (B&B) algorithms for QAP, a heuristic procedure is used to generate a suboptimal, but suitable, initial feasible solution. Then at any node of the tree, some bounding methods are used to find a "bound" on the best possible solution that can be expected from any descendent of that node, and the "bound" is compared with the objective value of the initial feasible solution. If the initial solution is better than what we can ever expect from any solution resulting from that node, then it is safe to stop branching from that node. In other words, we can discard that part of the tree from further consideration. What is happening is that an optimal permutation is being constructed iteratively, one element at a time.

Branch and bound techniques have evolved greatly over the past 40 years, starting with Gilmore who in 1962 solved a QAP of size $n = 8$, to the solution of nug30, a QAP of size $n = 30$ in 2000 (Loiola, 2007).

5.6.2. Metaheuristics

A metaheuristic is a heuristic method for solving a very general class of computational problems by combining user-given procedures - usually heuristics themselves - in the hope of obtaining a more efficient or more robust procedure. The name combines the Greek prefix "meta" ("beyond", here in the sense of "higher level") and "heuristic" (from ευρισκειν, heuriskein, "to find") (Metaheuristics, Wiki)

Metaheuristics are generally applied to problems for which there is no satisfactory problem-specific algorithm; or when it is not practical to implement such a method. Most commonly used metaheuristics are targeted to combinatorial optimization problems, but of course can handle any problem that can be recast in that form.

Before the end of the 1980s, most of the proposed heuristic methods for combinatorial optimization problems were specific and dedicated to a given problem. After that period, this paradigm has changed. More general techniques have appeared, known as metaheuristics. They are characterized by the definition of a priori strategies adapted to the problem structure. Several of these techniques are based on some form of simulation of a natural process studied within another field of knowledge-metaphors (Loiola, 2007).

There have been several metaheuristics to find a near optimum solution of QAP. Following section briefly describes and gives the core of three of these metaheuristics, which are largely used to solve the QAP model of stylus keyboard layouts.

Simulated Annealing: Simulated Annealing (SA) is a generic probabilistic metaheuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global minimum of a given function in a large search space. It is often used when the search space is discrete (e.g., all tours that visit a given set of cities). For certain problems, simulated annealing may be more effective than exhaustive enumeration - provided that the goal is merely to find an acceptably good solution in a fixed amount of time, rather than the best possible solution.

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one.

By analogy with this physical process, each step of the SA algorithm replaces the current solution by a random "nearby" solution, chosen with a probability that depends on the difference between the corresponding function values and on a global parameter T (called the temperature), that is gradually decreased during the process.

In the simulated annealing (SA) method, each point *s* of the search space is analogous to a state of some physical system, and the function *E*(*s*) to be minimized is analogous to the internal energy of the system in that state. The goal is to bring the system, from an arbitrary *initial state*, to a state with the minimum possible energy (Simulated Annealing, Wiki).

Ant Colony Optimization: The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. It is a population-based metaheuristic that can be used to find approximate solutions to difficult optimization problems.

In ACO, a set of software agents called artificial ants search for good solutions to a given optimization problem. To apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph. The artificial ants incrementally build solutions by moving on the graph. The solution construction process is stochastic and is biased by a pheromone model, that is, a set of parameters associated with graph components (either nodes or edges) whose values are modified at runtime by the ants (Ant Colony Op., Scholarpedia).

Table 5.1. Steps of Ant Colony Optimization

1. Choose the initial population of individuals
2. Evaluate the fitness of each individual in that population
3. Repeat on this generation until termination: (time limit, sufficient fitness achieved, etc.)
3.1. Select the best-fit individuals for reproduction
3.2. Breed new individuals through crossover and mutation operations to give birth to offspring
3.3. Evaluate the individual fitness of new individuals
3.4. Replace least-fit population with new individuals

Genetic Algorithm: A genetic algorithm (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms (EA) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover.

Genetic algorithms are implemented in a computer simulation in which a population of abstract representations (called chromosomes or the genotype of the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves toward better solutions. Traditionally, solutions are represented in binary as strings of {0,1}, but other encodings are also possible. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached (Genetic algorithm, Wiki).

Genetic algorithms find application in bioinformatics, phylogenetics, computational science, engineering, economics, chemistry, manufacturing, mathematics, physics and other fields.

Other Metaheuristics: There are a number of metaheuristics other than previously described three. However, these ones were rarely used in stylus keyboard layout optimization. Hence, only a list of them is given here as follows:

- Variable Neighbourhood Search Algorithm
- Local Search
- A Greedy Randomized Adaptive Search Procedure (GRASP)

- Scatter Search
- Tabu Search
- Neural Networks
- Iterated Local Search
- Evolutionary Computing

6. MODELLING STYLUS KEYBOARD LAYOUT FOR TURKISH LANGUAGE

This chapter is the first chapter for the application of theoretical knowledge explained in previous chapters for designing optimum stylus keyboard layout for Turkish language. The main purpose of this chapter is modelling the problem and calculation of the parameters, which are necessary to design optimum stylus keyboard layouts, both in longitudinal shape and square shape.

6.1. Basic Properties of Turkish Language

Turkish language has over 82 million native speakers all over the world. It is the official language of the Republic of Turkey and Turkish Republic of Northern Cyprus. Also, dialects of Turkish language are spoken largely in Central Asia and Caucasian countries. Its speakers are located predominantly in Turkey and Cyprus, with smaller groups in Iraq, Greece, Bulgaria, the Republic of Macedonia, Kosovo, Albania and other parts of Eastern Europe. Turkish is also spoken by several million immigrants in Western Europe, particularly in Germany.

The 2005 edition of *Güncel Türkçe Sözlük*, the official dictionary of the Turkish language published by Turkish Language Association, contains 104,481 words, of which about 86 per cent are Turkish and 14 per cent are of foreign origin.

The vowels of the Turkish language are, in their alphabetical order; *a, e, ı, i, o, ö, u,* and *ü*. There are no diphthongs in Turkish; when two vowels come together, which occurs rarely and only with loanwords, each vowel retains its individual sound. There are 21 consonants; b, c, ç, d, f, g, ğ, h, j, k, l, m, n, p, r, s, ş, t, v, y and z. Among them, “ğ” never occurs at the beginning of a word or a syllable, but always follows a vowel.

6.2. Keyboard Shape Models of the Problem

Stylus keyboard layout will be designed for both the longitudinal shape and square shape. Since the *D* matrix changes according to the shape of the keyboard, each of these two different shaped keyboards should be designed separately.

6.2.1. Longitudinal Shape Model

Standard longitudinal stylus keyboards for Turkish Language mostly have the following properties:

- Having three rows for letters
- Top row made up of 12 keys for letters
- Home (or centrum row) has 11 keys for letters
- Bottom row involves 9 keys for letters

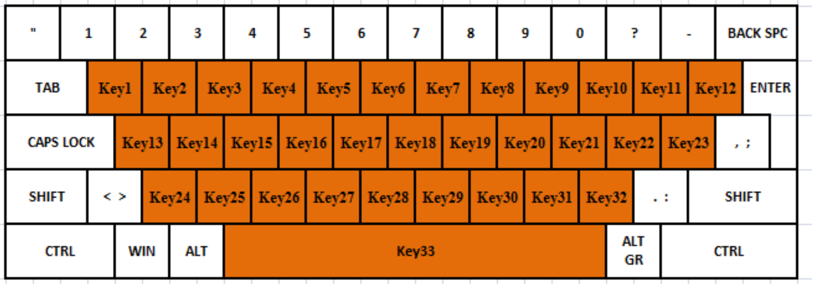


Figure 6.1. Longitudinal shape keyboard model of the problem

Hence, there have been 33 Keys (32 keys for letters including q, w and x; also Space Bar key) in a standard longitudinal shape keyboard. In spite of the fact that there have been a great number of different stylus keyboards in shape; for the consistency of the study, some assumptions are made about the generalization and the standardization of these shapes. The main properties of the model keyboard shape are as follows;

- The home row is "indented" just a half key right according to the top row.
- Also, the bottom row is indented just a half key to the right according to the home row.
- All keys, except the space bar key, are standard in shape, being just a square with an edge of 30 units.
- Space bar is designed in seven-key length.
- Neighbouring keys have an interval of two units, from both the lateral neighbouring and up-down neighbouring. These intervals are called "dead pixels" in literature and these dead pixels are useful to reduce the error rate, but increase the typing speed a little.
- Since the distances should be measured from a "point", not a region, the stylus is assumed to be taping just on the centre of the key. Due to this fact, all of the distances are calculated upon the centre point of the keys.
- For learning curve of the users and logic, blank character is assigned and fixed to space bar initially.

6.2.2. Square Shape Model

It had become a common strategy to optimize the stylus keyboard in a square shape, so as to minimize the distances between the keys. This strategy will force the stylus to move in a narrower central region. OPTI or FITALY stylus keyboard layouts can be given as an example of this shape keyboards.

This is also logic for mobile device manufacturers. Because, it is easy to install or embed a square shaped stylus keyboard to the device. Due to these reasons, a square shaped stylus keyboard will also be tried to design for Turkish language.

The square shaped keyboard model has the following properties:

- Having six columns and six rows.
- All 36 keys are square.
- All keys 36 keys being just a square with an edge of 30 units.
- Neighbouring keys have an interval of two units, from both the lateral neighbouring and up-down neighbouring.
- The first three keys of the bottom row are assigned to q, w and x; and the next three keys are *empty*, meaning that any character other than letters or blank characters, such as punctuation marks or function keys can be assigned later by the manufacturer. Hence, the sixth row will never take consideration in model from now and on.
- Since the remaining 30 keys are identical, there is not any constraint about the assignment of blank character. In other words, blank character can be assigned any of the 30 keys by the model.

Key1	Key2	Key3	Key4	Key5	Key6
Key7	Key8	Key9	Key10	Key11	Key12
Key13	Key14	Key15	Key16	Key17	Key18
Key19	Key20	Key21	Key22	Key23	Key24
Key25	Key26	Key27	Key28	Key29	Key30
q	w	x	*	*	*

Figure 6.2. Square shape keyboard model of the problem

6.3. Modelling the Problem as a QAP

To make a mathematical model of a real life equipment design problem, it will be worthwhile thinking the usage characteristics and phases of the equipment which is being designed.

Stylus keyboards are used by a pointer like a finger, stylus, pencil, mouse pointer, etc. to tab the keys during writing. From now on, the pointer will be named as *stylus* for consistency.

The stylus is assumed to be automatically positioned to the first letter of the first digraph in the start of the typing action, and also after any punctuation mark. In other words, all punctuation marks will be ignored. Also, function keys such as Shift and Ctrl are ignored, hence the typed texts will accepted as *case-insensitive.*

The quality criterion of the keyboard layout optimization will be the average distance taken by the stylus per digraph. To quantify this criterion, total distance taken by the stylus should be divided by the number of digraphs that are typed during typing a text. Someone may formulate this criterion by plotting the exact route on the keyboard with a probabilistic method.

Instead of plotting the exact route, someone may also formulate this criterion by the segments of the total route, as done in this work. For example, while typing “*Özge gel*” in stylus keyboard; total distance taken by the stylus can be calculated by summing up the plotted route as

$$\text{“ö} \rightarrow z \rightarrow g \rightarrow e \rightarrow \_ \rightarrow g \rightarrow e \rightarrow l\text{”} \quad \textit{or}$$

$$|\text{ö}z| + |zg| + 2 * |ge| + |e\_| + |\_g| + |el|$$

where $|ij|$ figures the distance between letters of the digraph “ij”. Here, $|ge|$ is multiplied by two, since the digraph “ge” is observed twice in “Ö$zge\ gel$”.

So, our criterion can be expressed in terms of digraphs in any text as follows:

$$\min_{Layout} \sum_{\substack{All\ digrap\ h\ types \\ in\ the\ text}} \{n_{digrap\ h} * |digraph|\} \tag{6.1}$$

where $n_{digrap\ h}$ denotes the number of a digraph observed in a text and $|digraph|$ denotes the distance between the letters of that digraph in that layout. Since the direction of the criterion is minimization; than the last expression above states to find the *layout* which minimizes "*the sum of the number of digraphs multiplied by the distance between the letters of that digraph*".

When looked carefully, it can be easily noted that this expression is just the object function of QAP, which was stated before as:

$$\min_{\pi \in S_n} \sum_{i,j=1}^{n} f_{i,j} * d_{\pi(i),\pi(j)} \tag{6.2}$$

If the criterion expression stated above is applied to Turkish language in general, *not to a specific text*, then the layout which minimizes an average text written in Turkish will be designed.

So, let L_{30} be the set of all layouts which can be formed by 30 letters (29 letters plus blank character) and 30 keys, and let $\pi \in L_{30}$. Consider $f_{i,j}$ the general digraph frequency of the *digraph "ij"* in Turkish language and $d_{\pi(i),\pi(j)}$ the distances between keys $\pi(i)\ and\ \pi(j)$. If each permutation π represents an assignment of letters to keys, the problem expression of our model in permutation style becomes as;

$$\min_{\pi \in L_{30}} \sum_{i=1}^{30} \sum_{j=1}^{30} f_{i,j} * d_{\pi(i),\pi(j)} \tag{6.3}$$

$$i,j \in \{a,b,c,ç,d,e,f,g,ğ,h,ı,i,j,k,l,m,n,o,ö,p,r,s,ş,t,u,ü,v,y,z,blank\}$$

Integer linear programming model for optimum stylus keyboard layout for Turkish language can also be formulated as follows;

$$\min \sum_{i=1}^{30} \sum_{j=1}^{30} \sum_{k=1}^{30} \sum_{p=1}^{30} f_{i,j} * d_{k,p} * x_{i,k} * x_{j,p} \tag{6.4}$$

$$s.t.\ for\ all\ j;\ \sum_{i=1}^{30} x_{i,j} = 1 \qquad 1 \leq j \leq 30 \tag{6.5}$$

$$for\ all\ i;\ \sum_{j=1}^{30} x_{i,j} = 1 \qquad 1 \leq i \leq 30 \tag{6.6}$$

$$x_{i,j} \in \{0,1\} \qquad 1 \leq i,j \leq 30 \tag{6.7}$$

The first constraint in the integer linear programming model defines that a letter can be assigned to one and only one key. With a similar manner, the second constraint defines that a key can be assigned to one and only one letter. The decision variables $\boldsymbol{x_{i,j}}$ are Binary variables:

$$x_{i,j} = \begin{cases} 1; & if\ key\ i\ is\ assigned\ to\ letter\ j \\ 0; & else \end{cases} \tag{6.8}$$

6.4. Generating the Distance Matrices

6.4.1. Generating the Distance Matrix of Longitudinal Keyboard

The distance matrix, *which also can be named as Road Matrix or shortly Matrix D*, is the matrix made up of the elements $[d_{i,j}]$; where $d_{i,j}$ is the distance between keys i and j on our keyboard models.

For generating the **D** matrix, the longitudinal shape keyboard model is launched on an x-y coordinate system, where the $(\mathbf{0},\mathbf{0})$ point is selected as the top-left corner of the keyboard.

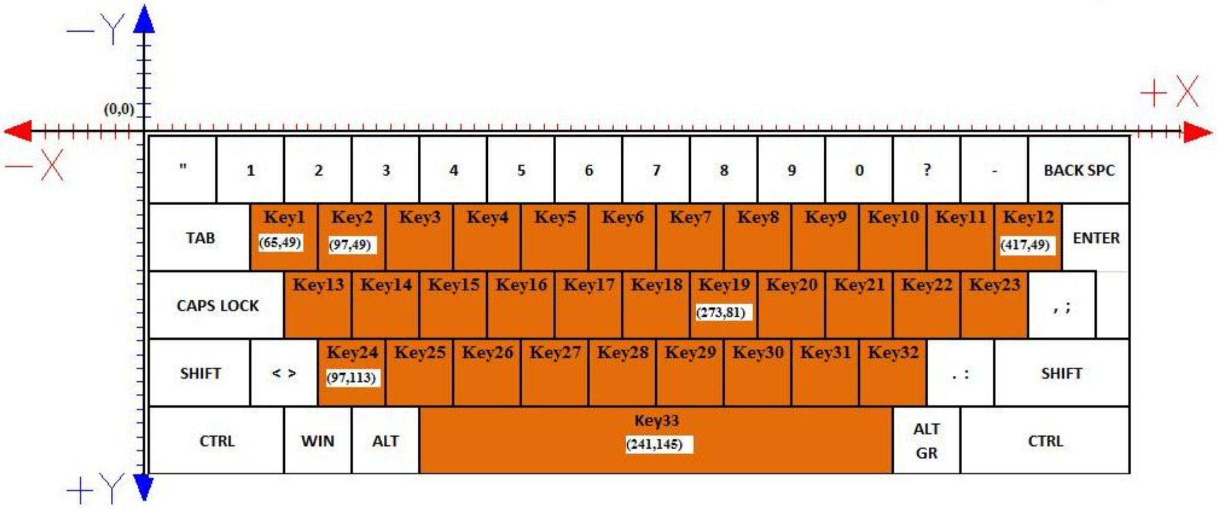


Figure 6.3. Longitudinal shape keyboard model on x-y coordinate system

As seen in Figure 6.3., the top-left corner of the main frame of the keyboard (*the corner of "é" key in standard Q layout*) is arbitrarily chosen to be the (0,0) origin of the x-y coordinate system, where the right hand side is accepted as +x direction and downward is +y direction. By this analytical description of the keyboard shape, the $(\boldsymbol{x},\boldsymbol{y})$ coordinates of the midpoint of each of 33 keys (one of those is the space key) is calculated, as listed in Table C.2 in Appendix C.

The distance between key i, with centre coordinates $(\boldsymbol{x_i}, \boldsymbol{y_i})$ and key j, with center coordinates $(\boldsymbol{x_j}, \boldsymbol{y_j})$, calculated by Pythagorean Theorem as;

$$d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \tag{6.9}$$

Resulting 33 by 33 Distance Matrix (D) is shown in Table B.1 of Appendix B.

6.4.2. Generating the Distance Matrix of Square Keyboard

For generating the D matrix, the square shape keyboard model is launched on an x-y coordinate system, where the $(0,0)$ point is selected as the top-left corner of the keyboard. As seen in Figure 6.4., the top-left corner of the main frame of the keyboard is supposed to be the (0,0) origin of the x-y coordinate system, where the right hand side is accepted as +x direction and downward is +y direction. By this analytical description of the keyboard shape, the (x, y) coordinates of the midpoint of each of 30 keys is calculated, as listed in Table C.3 in Appendix C.

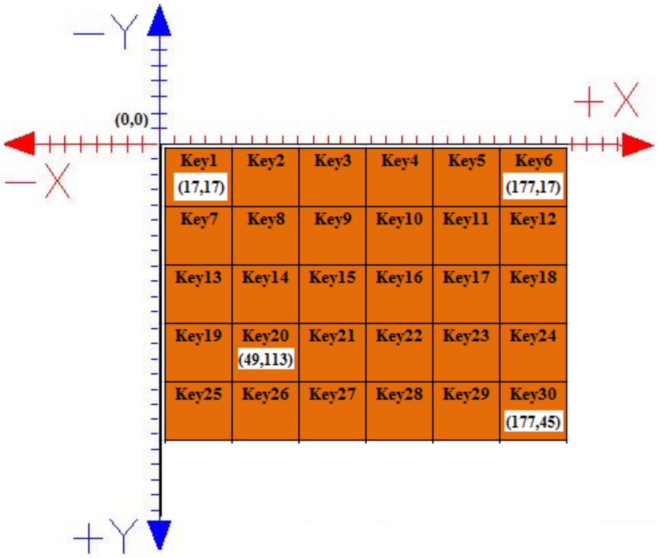


Figure 6.4. Square shape keyboard model on x-y coordinate system

With the same manner in longitudinal shape keyboard, the distance between key i, whose coordinates are (x_i, y_i) and key j, whose coordinates are $\left(x_j, y_j\right)$, calculated by Pythagorean Theorem as;

$$d = \sqrt{(x_i - x_{j)}{}^2 + (y_i - y_{j)}{}^2} \tag{6.10}$$

Resulting 30 by 30 Distance Matrix (D) is shown in Table B.2 of Appendix B.

6.5. Generating the Flow Matrix

The flow matrix, *which also can be named as Load Matrix or Cost Matrix or shortly Matrix* F, is the matrix made up of the elements$[f_{i,j}]$; where $f_{i,j}$ is the frequency of the digraph "i, j".

29 letters in Turkish alphabet including the blank character generate $30x30 = 900$ different digraph (different pair wise permutations of 30 characters) types whose set can be defined as

$$Set\ of\ Turkish\ Digraphs = \{"aa", "ab", "ac", \ldots, "zz", "z\_", "\_a", "\_b", \ldots, "\_z", "\_\_"\}$$

where “_” refers to blank character.

A detailed statistical analysis has been carried out to quantify the relationship between letters in Turkish language. To explain with other words, the frequency table for these 900 digraphs were created by random sampling method.

The Turkish e-texts that have been analysed during creating F matrix is categorized into 5 groups, namely;

1. *Daily articles*: 50 different daily different political opinion newspaper articles written by randomly selected authors. Distribution among the newspaper is made with respect to their nationwide circulation (Total: 203,991 digraphs).
2. *Magazines*: 50 different nationwide magazines of various core topics, ranging from medicine, education, business, finance, literature, sports and science (Total: 717,163 digraphs).
3. *Turkish classics summary*: 66 famous Turkish classics' summary. Some of these classics were written in Ottoman Turkish and translated to today's Turkish. Main purpose for selecting these old classics is the necessity for the old words' weight in overall digraph frequencies (Total: 330,512 digraphs)
4. *World classics' translation summary*: Summaries of 107 globally well known classics' Turkish translations. Main purpose of adding these summaries is the inclination of foreign originated words' weight in overall digraph frequencies (Total: 519,162 digraphs).
5. *Random articles*: 34 scientific or academic articles on different topics such as medicine, history, sociology, psychology and engineering (Total: 426,015 digraphs)

Detailed list of the sources of the analyzed documents while generating the digraph frequency table can be found in Table B.3. through Table B.7 of Appendix B.

In overall; 2,196,843 digraphs which make over 4,000 pages were analysed by the help of Microsoft Excel macros to generate the 30 by 30 flow matrix $\boldsymbol{F}$, as shown in Table B.8. of Appendix B. The F Matrix depicts the number of a digraph among 1,000 digraphs. For instance, in a Turkish text having 1,000 digraphs; the digraph "*me*" will be seen 6.38443 times and the digraph "*tk*" 0.38636 times. It should be noted that digraph $\boldsymbol{f_{i,j}}$ does not have the same frequency with $\boldsymbol{f_{j,i}}$ as can be illustrated by the digraph "*rd*" with the frequency of 3.52432 whereas "*dr*" with the frequency of 0.16033.

During statistical analysis of this stage, case sensitivity was off, meaning that all upper cases were transformed to lower cases, such as "England" transformed to "england".

However, all punctuation characters were treated as an infringement of digraph analysis. This is because, while using keyboard, punctuation marks are generally created with a combination of some keys (Ctrl, Alt, Shift ...). For example, the text “*Ali, gel!*” has 5 digraphs; “*al”, “li”, “_g”, “ge”* and “*el”.*

From this part, unless otherwise stated, the frequencies expressed are “frequency in one thousand”. For instance, if frequency of a digraph is said to be 4.54213, then it should be understood that, mentioned digraph’s frequency is “*4.54213 in one thousand digraphs*”.

6.6. Some Interesting Features of Turkish Language

In this section, some interesting features that are learned from the generated digraph frequency table are presented. These features are not directly related to the design problem and listed for attention of interested people.

- Most frequent digraphs are “n_” and “e_” (with 20.90581 and 18.07441 frequencies, respectively), where “_” denotes the blank character.
- Most frequent digraphs whose both characters are letters:”ar” and “la” (with 17.05680 and 15.87727 frequencies, respectively)
- 98 digraphs have 0 frequency (such as “cç”,”ğj”,”jh”,”sj” ..etc)
- Sum of the frequencies of 285 least frequent digraphs is less than 1.
- The letter which is the most observed as the first letter of Turkish words is “b”.
- The letter which is the most observed as the last letter of Turkish words is “n”.

Relative frequencies of letters with blank character in Turish language are shown in Table B.9 in Appendix B.

7. INITIALLY FIXED ASSIGNMENTS

In this chapter, initial assignments of foreign characters to distal keys and blank character to space bar are explained in detail.

7.1. Introduction

After modelling the problem and generating the flow matrix and distance matrices for both longitudinal and square shape keyboard models, the problem of designing optimum stylus keyboard layout for Turkish language that minimizes the distance taken by the stylus is ready to be solved.

7.2. Assignment of Blank Character

7.2.1. Longitudinal Type Keyboard

Standard virtual keyboards designed for Turkish language involve 32 keys for letters and a relatively long space bar. It was previously stated that in our new longitudinal design, relatively long key (space bar) should be assigned to “blank character” due to three reasons;

- Users are accustomed to space bar for blank character for a century, since even typewriters have the same property.
- Like almost every language, blank character is used more than any other letter in Turkish language. Most used character should be assigned to most remarkable key (which is space bar) from the view of Ergonomics and for a better learning curve of users of new design.

- Most computer systems do not allow the users to change the assignment of space bar with a new character.

7.2.2. Square Type Keyboard

Since all of the keys of square shape keyboard model are identical, there will not be any initial assignment for blank character. Just the solution of the model will assign the blank character and hence, the space bar will be the key to which the blank character is assigned.

To increase the typing speeds of users, space bar can be visually differentiated, with a different colour for example.

7.3. Assignment of Foreign Letters

7.3.1. Longitudinal Type Keyboard

Turkish language has 29 letters for both upper and lower case which can be listed as

$$\{a, b, c, \text{ç}, d, e, f, g, \text{ğ}, h, \text{ı}, i, j, k, l, m, n, o, \text{ö}, p, r, s, \text{ş}, t, u, \text{ü}, v, y, z\}$$

These letters do not change into another format in the middle or end of the word, like the case as in Arabic language. The letters “q”, “w” and “x” are not used in Turkish words directly, but our new longitudinal design should also involve these 3 letters due to the following reasons;

- Turkish users may sometimes need these 3 letters during using virtual keyboards, while typing some foreign words for example.
- Virtual keyboards are largely used in mobile devices connected to Internet. While typing internet addresses, these 3 letters, especially “w” are mandatory.

- People whose mother language is different from Turkish may use this new design, such as tourists or businessmen visiting Turkey.

In early stages of this work, “q, w and x” are thought to be assigned to the most distal keys. In order to achieve this, “Centralization Index” is used.

Centralization Index, ρ; for any key on any keyboard type, is the sum of the distances from that key to each of the other keys. This can also be computed by summing the elements of the row corresponding to that key in D matrix. To express ρ mathematically, the Centralization Index of key *i* on a *keyboard type* is:

$$\rho(i, keyboard\ type) = \sum_{j=1}^{n} d_{i,j} \tag{7.1}$$

where *n* is the number of keys in that keyboard type model. For longitudinal type keyboard, ρ for all keys are calculated as follows:

$$\rho(i, longitudinal\ keyboard) = \sum_{j=1}^{33} d_{i,j} \qquad for\ 1 \leq i \leq 33 \tag{7.2}$$

As seen on Table B.10 in Appendix B, the most distal keys were Key23, Key1 and Key12. To depict clearly Key23, Key1 and Key12 are the keys assigned to letters “i”, “q” and “ü” on a standard Turkish QWERTY Layout.

Assigning three foreign letters to different sides of a keyboard is not logic for learning curve of the users. The user should remember where the foreign letters in a short time; otherwise he would be confused to during typing which foreign letters were in the right side and the one in the left side. So, it is decided to locate these 3 letters to distal keys as a group.
There have been three distal three-key-blocks option:

a. “Key12-Key23-Key11” group with a total centralization index of 16,968.27

b. “Key1-Key2-Key13” group with a total centralization index of 16,212.56

c. “Key10-Key11-Key12” group with a total centralization index of 15,871.19

The most distal block among three is the “Key12-Key23-Key11” group. Fortunately, F Layout also involves “q” and “w” letters on Key11 and Key12, respectively. To ease the shift of F Layout users to our new longitudinal design by increasing the similarities between new design and F Layout; we assigned “q” to Key11, “w” to Key12 and “x” to Key23.

Up to this point of the longitudinal design, four characters are assigned to four keys and these are the fixed assignments before the start of the solution phase, shown in Table 7.1.

Table 7.1. Fixed assignments of longitudinal shape before solution phase

Key	**Character**
Key11	q
Key12	w
Key23	x
Key33	Blank

Flow Matrix does not cover “q, w, x” letters, since there is hardly ever relationship between these letters and Turkish letters. Due to this fact, these three letters will take no attention during any stage of the solution. Also, the keys assigned to these three letters (Key11, Key12 and Key23) are removed from the Distance Matrix and will take no attention from now and on. So, the revised Distance Matrix is 30 by 30 just as the Flow Matrix.

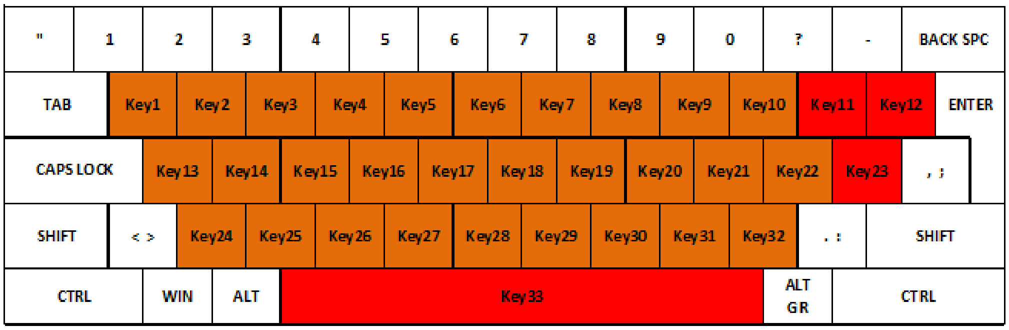


Figure 7.1. Initially fixed keys on longitudinal keyboard

Hence, the problem is now to assign 30 characters to 30 keys in a way that minimizes the distance taken by the stylus with the constraint that Key33 should be assigned to "blank character".

7.3.2. Square Type Keyboard

As stated in chapter six, the bottom row of square keyboard model is initially reserved for the three foreign letters and function keys. During the generation of D matrix of square keyboard model, these 6 keys were ignored. For this reason, there is no initial assignment for square type keyboard.

7.4. Solution of the QAP Model

To sum up, longitudinal keyboard model has 30 letters to be assigned to 30 keys with the constraint that blank character should be assigned to space bar key. Assigning 29 letters to 29 keys results in $29! = 8{,}841{,}761{,}993{,}739\ x\ 10^{18}$ different layout combinations.

Also, square keyboard model has 30 letters to be assigned to 30 keys without any initial assignment constraint. Assigning 30 letters to 30 keys result with $30! = 265.252.859.812.191\ x\ 10^{18}$ different layout combinations.

Trying all of these layouts that minimizes the $[F * D]_{30x30}$ even with the help of serial connected power CPUs will take a processing time of months or years. For this reason, this quadratic assignment problem can only be solved with the help of a heuristic or metaheuristic method.

8. A DECOMPOSITION-BASED HEURISTIC FOR STYLUS KEYBOARDS

As mentioned in Quadratic Assignment Problem chapter, there is no exact algorithm which solves problems for $n \geq 30$ in a reasonable time. For this reason a new deterministic, greedy, constructive, decomposition based heuristic, named *Optimum Connected Optimal Blocks (OCOB*), to solve stylus keyboard layout problem is developed in this chapter.

8.1. Optimum Connected Optimal Blocks Heuristic (OCOB)

8.1.1. Introduction

The main idea behind OCOB heuristic can be summarized with the concept from system analysis and management sciences: "After the whole system is partitioned into smaller units, if these small units are optimized and then connected with each other in the best way; then the whole system will be near optimum". This general problem solution approach is named as "decomposition" in literature. Decomposition refers to the process by which a complex problem or system is broken down into parts that are easier to conceive, understand, program, and maintain.

8.1.2. Background of OCOB Heuristic

In literature, most of the solution techniques on quadratic assignment problems are designed as an exact solution or as a metaheuristic. Loiola (2007) denotes, in his literature survey on QAP between the years 1999 and 2005, that about 30 papers deal with exact methods, more than 90 are dedicated to metaheuristic methods, while just less than 20 are written for heuristics methods; a natural consequence of the NP-hardness of the problem.

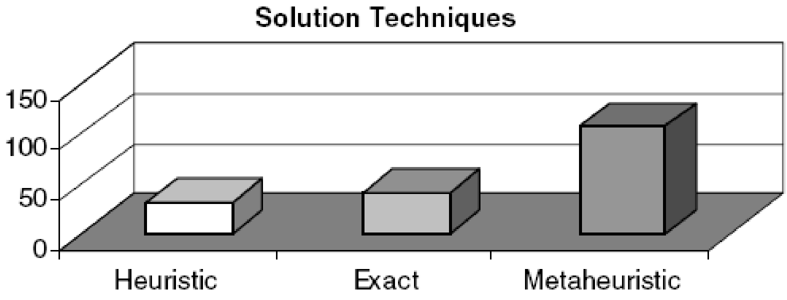


Figure 8.1. Papers categorized according to solution techniques on QAP (1999-2005)

The main idea while developing this heuristic is the “Relax-and-Fix” approach. This approach is generally applied when the solution set is so large that enumerating all candidate solutions is impossible due to time inadequacy. This approach is largely used for improving heuristics for flow sheet decomposition for scheduling, solving one-stage one-machine lot-scheduling, multi-product multi-level production planning.

During the literature study of this thesis, no heuristics designed for quadratic assignment problem based on relax-and-fix approach has been confronted with.

The main issue and decision point of relax-and-fix approach based OCOB heuristic is the selection of keys and letters to be optimized in phases; or in other words, which letters and keys to match pair wise in a phase. The answer depends on the bottleneck function. This function can be obtained by adding a “max” operation before the term $f * d$ in QAP formula:

$$\min_{\pi \in S_n} \ \max\left\{f_{i,j} * d_{\pi(i),\pi(j)} \quad : 1 \leq i,j \leq n\right\} \tag{8.1}$$

The philosophy in bottleneck function is “trial of minimization of the maximum costs in a cost minimization problem will result in an acceptable solution”. To achieve this; letters which are highly related with other letters should be assigned to keys which are relatively close to other keys.

8.1.3. Algorithm of OCOB Heuristic

In OCOB, first the keys are sorted ascending order by centralization index and the letters are sorted in descending order by connection index. Then, these keys and letters are grouped in equal numbers to be matched. Letters in a group are assigned to keys in the corresponding group in each phase while all of the assignments made in the previous phases are fixed.

In nearly all heuristics and metaheuristics, there have been some decision points such as initial values of decision variables, stopping criteria, maximum iteration or node number, relative gap and so on. OCOB has also a decision point: how many letters or keys should the groups have? The answer depends on the technical properties of the computer on which the solution codes will run as well as the time limit of the designer.

Algorithm of OCOB is;

Step#1: Sort the keys in ascending order by centralization index and the letters in descending order by connection index.

Step#2: Set *n*, the maximum number of letters that can be assigned optimally to keys in a run in a reasonable amount of time.

Step#3: Fix the assignments of letters to keys that are initially made and should *not* be re-assigned (*if there is any).*

Step#4: Solve the first unassigned *n* letters and *n* keys QAP problem optimally with revised distance and flow matrixes which are "(*n + fixed assignments) by (n + fixed assignments)*"

Step#5: Fix the assignments made in Step#4 for the rest of the solution

Step#6: Repeat Step#4 through Step#5 until all letters are assigned to all keys.

Step#7: Repeat Step#4 through Step#6 for *n=n-1, n=n-2, n=n-3* and *n=n-4*

Step#8: Take the layout having minimum cost function value among designed 5 layouts.

8.1.4. Flow Chart of OCOB Heuristic

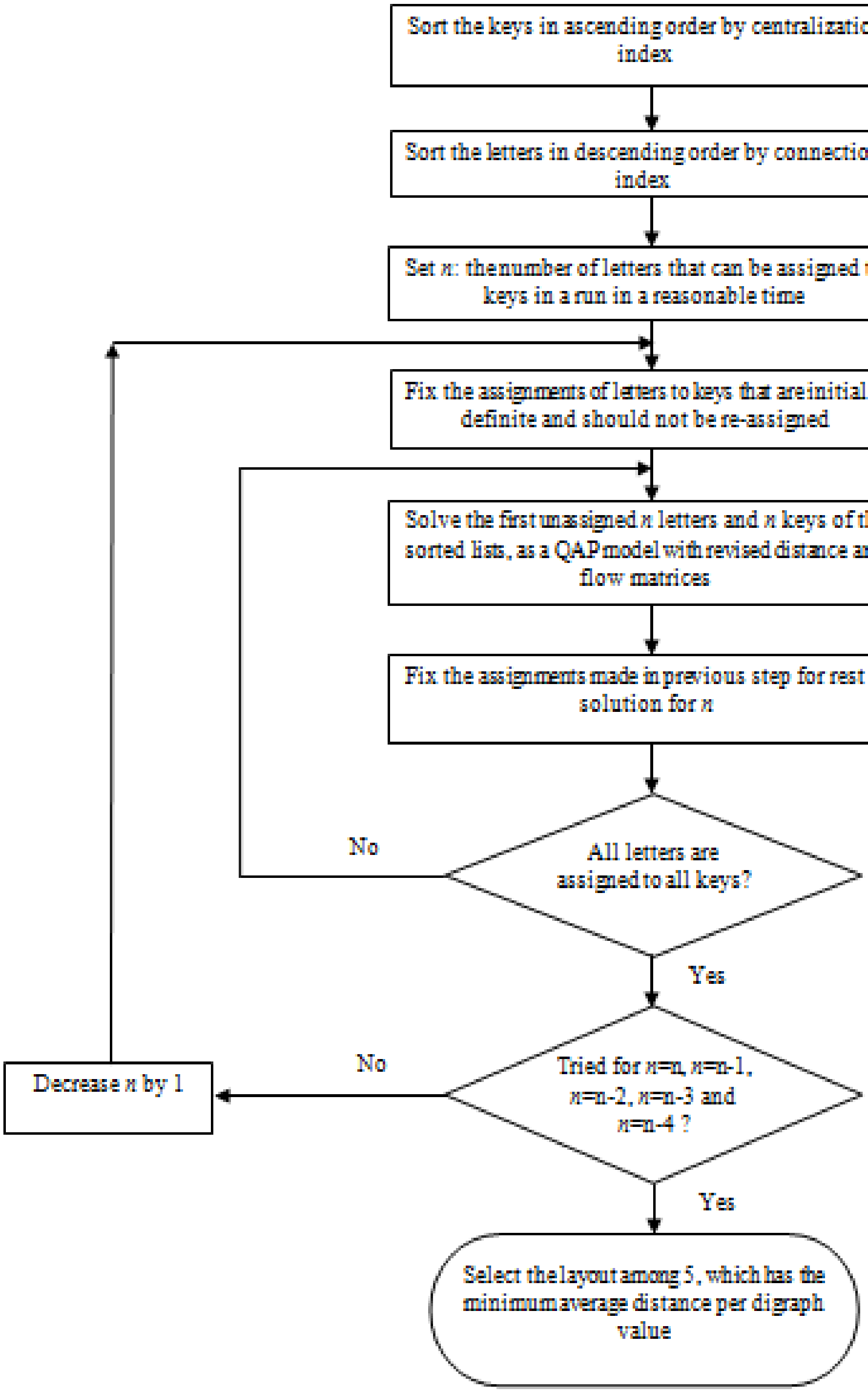


Figure 8.2. Flow chart of OCOB Heuristic

8.1.5. Centralization Index of Keys

Centralization Index, denoted by $\boldsymbol{\rho}$; for any key on any keyboard type, is the sum of the distances from that key to each of the other keys. This can also be computed by summing the elements of the row corresponding to that key in D matrix. To express $\boldsymbol{\rho}$ mathematically, the Centralization Index of key *i* on a *keyboard type* is:

$$\rho(i, keyboard\ type) = \sum_{j=1}^{n} d_{i,j} \tag{8.2}$$

where *n* is the number of keys in that keyboard type model and $\boldsymbol{d_{ij}}$ the distance between key $\boldsymbol{i}$ and key $\boldsymbol{j}$. Shortly, this index is used to determine the most central keys.

Centralization Index is a kind of distance rating, which is used to determine the relative location of a key to other keys of a given keyboard model. In order to apply OCOB, the keys are sorted in ascending order (*most central to least central*) according to their centralization index.

8.1.6. Connection Index of Letters

Connection Index, denoted by $\boldsymbol{\tau}$; for any letter, is the sum of the frequencies of the digraphs involving that letter. This index, for example, for letter *´i´* is computed by summing the elements of the row *i* and column *i* and subtracting the intersecting element frequency ($f_{i,i}$) in *n*x*n F* (frequency) matrix. To express $\boldsymbol{\tau}$ mathematically, the *Connection Index* of letter *i* is:

$$\tau(i) = \sum_{j=1}^{n} f_{i,j} + \sum_{j=1}^{n} f_{j,i} - f_{i,i} \tag{8.3}$$

where $\boldsymbol{f_{i,j}}$ is the digraph frequency of the digraph ***"ij"*** and *n* is the number of letters. The reason for the subtraction of $\boldsymbol{f_{i,i}}$ is the repetitive summation of this term in the previous two terms (i. e., to avoid the addition of $f_{i,i}$ twice to the summation since it repeats itself in row *i* and column *i*). That is, all diagonal elements of *F* matrix are subtracted in calculating *Connection Index* values. Connection Index determines a letter´s connectivity with all other letters. It is used to determine the priority of a letter´s entrance to the solution phases of the OCOB heuristic.

In order to apply OCOB, the letters should be sorted in descending order according to their connection index. Connection index is a kind of relation index of a letter with other letters.

8.2. Sum of the Frequencies of Assigned Letters

To analyse the improvement of the objective function through the phases, a function for "Sum of the frequencies of the assigned letters" is defined. To express mathematically, we define a function, say $\boldsymbol{\varphi}$, equals to the summation of the frequencies of the digraphs that can be formed by the letters that are assigned and fixed until the end of a phase.

$$\varphi = \sum_{i,j \epsilon S} f_{i,j} \tag{8.4}$$

where S is the set of digraphs formed by the assigned and fixed letters.

For example; let 12 letters, say $a, e, i, n, r, l, ı, k, d, t, m,$ and $blank,$ be assigned and fixed till the end of a phase. Then there will be $12x12 = 144$ different digraph types which can be formed by these 12 letters. By the values in Table B.8 in Appendix B;

$$\varphi = \sum_{i,j} f_{i,j} = 559.71837$$

$$\mathrm{i, j} \in \{\mathrm{a, e, i, n, r, l, ı, k, d, t, m, blank}\}$$

This connotes that the $12x12 = 144$ digraphs formed by these 12 letters account for 55.97 per cent of all digraphs of an average Turkish text.

8.3. Change of the Average Distance throughout the Phases

The average distance taken by the stylus, at the end of any phase, say $\boldsymbol{\psi}$, can be calculated with the following formula:

$$\psi = \frac{\sum_{i,j=1}^{n} f_{i,j} * d_{p(i),p(j)}}{\sum_{i,j=1}^{n} f_{i,j}} \tag{8.5}$$

$$\psi = \frac{\sum_{i,j=1}^{n} f_{i,j} * d_{p(i),p(j)}}{\varphi} \tag{8.6}$$

where n is the number of letters fixed to keys until the end of the phase and $p(i)$ is the key to which the letter *i* is assigned.

For example; let 12 letters, say $a, e, i, n, r, l, ı, k, d, t, m,$ and $blank,$ be assigned and fixed till the end of a phase. Then there will be $12x12 = 144$ different digraph types which can be formed by these 12 letters. Also, let the objective function value, ***z***, of the quadratic assignment problem with the revised (*rows and columns of other letters deleted*) 12 by 12 digraph frequency and distance matrixes be $62{,}354.6627$. Hence,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{29{,}876.0146}{559.7184} = 53.37687\ units$$

This result can be divided by *30* to transform the unit to *key-length* since all D matrices are generated according to the model where one edge of the square shaped keys was 30 units. So,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{53.3769}{30} = 1.7792\ key - lengths$$

This value expresses that, if a Turkish text involving only these 12 letters were typed, the average distance taken by the stylus would be 1.7792 key-lengths.

This average distance will continuously increase throughout the phases with no doubt until the end of the solution, because every new assignment is worse than the last assignment made just before it, due to the centralization index list of the keys. But the increment in the average distance will continuously decrease, with a positive but decreasing slope of the tangent of the curve.

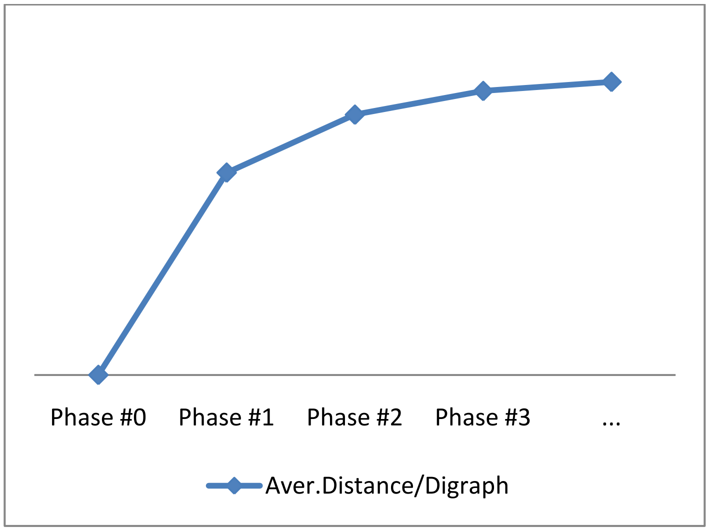


Figure 8.3. The change of Aver.Dist. per Digraph taken by the stylus through the phases

9. DESIGNING OPTIMUM LONGITUDINAL STYLUS KEYBOARD LAYOUT

In this chapter, the detailed work to design the optimum longitudinal stylus keyboard layout for Turkish language using the OCOB heuristic is explained.

9.1. Introduction

In the previous chapters, the design of optimum stylus keyboard layout for Turkish language has been modelled but could not be solved, due to its NP-hardness. It is now possible to solve this model by using OCOB heuristic.

Since all distance matrices are generated according to a keyboard model having square keys whose one edge is 30 units, the objective function of quadratic assignment problem is divided by 30 to transform the unit of the average distance per digraph to *key-length*. Also, the frequently used term “*revised matrices*” means “deleting the rows and columns of a matrices corresponding to the keys or letters that are *not* in attention in that phase.”

9.2. Technical Background of Computer Solutions

The computer solution of OCOB phases for the design of stylus keyboard layout problem is coded in The General Algebraic Modeling System (GAMS) which is a high-level modeling system for mathematical programming and optimization. It consists of a language compiler and a stable of integrated high-performance solvers.

The codes of OCOB is a reconstruction and adaptation of a sample for MIQCP type in GAMS Model Library contributed by Burkard R.E. from QAPLIB- a Quadratic Assignment Problem Library (Burkard, 1997). The codes were tested by using the data sets from the QAPLIB (QAPLIB Library).

The solver used to solve the phases of OCOB is CPLEX which is a powerful Linear Programming (LP), Mixed-Integer Programming (MIP), Quadratically Constraint Programming (QCP) and second order cone programs, and Mixed-Integer Quadratically Constraint Programming (MIQCP) using branch-and-cut or Barrier algorithms.

The phases of OCOB in GAMS using Cplex were run in Bogazici University Industrial Engineering Graduate Computer Laboratory due to licence of GAMS. The computer used during all phases of OCOB was same and had an Intel Core 2 Duo processor and a RAM of 4 GB.

OptCR option of Cplex was set equal to 0. OptCR is a relative optimality criterion for a MIP problem. The OptCR option asks Cplex to stop when

$$\frac{(|BP - BF|)}{(10^{-10} + |BF|)} < OptCR \tag{9.1}$$

where BF is the objective function value of the current best integer solution while BP is the best possible integer solution. Similarly OptCA is the absolute optimality criterion for a MIP problem in Cplex.

Both of the OptCA and OptCR were set equal to 0.00 to force GAMS to try all of the combinations of binary decision variables.

9.3. Step#1: Sorting the Keys and Letters

The first step of OCOB starts with sorting the keys in ascending order by *Centralization Index* and the letters in descending order by *Connection Index*.

9.3.1. Sorting the Keys

While sorting the keys, as mentioned in Chapter 7; Key11, Key12 and Key23 will take no attention throughout the phases. 30 keys (*including the Space Bar key-Key33*) are sorted in ascending order upon their centralization indices as shown in Table B.10 in Appendix B.

9.3.2. Sorting the Letters

Before the generation of the digraph frequency table (*F* Matrix) as stated in Chapter 7, the foreign letters “q, w and x” were assigned to Key11, Key12 and Key23, respectively. After this evocation, the remaining 30 letters (*including the blank character*) are sorted in descending order upon their connection indices as shown in Table B.9 in Appendix B.

Hereafter, both the centralization and connection indices will not be used; they were just the tools to sort the keys and letters within each other.

9.4. Step#2: Setting *n*

After the completion of sorting the letters and keys, next step is the grouping of letters and keys to be matched within. Since the keys and the letters are sorted, the group creation depends only on the size of the group – hence the parameter *n*.

The effect of the *n* to the solution is as follows:

- *If n increases*, the amount of time to reach the optimum solution increases. Of course, if n is taken equal to the number of letters or keys, then the objective function value found will be the global optimum solution. Unfortunately, it is practically infeasible to develop heuristic and metaheuristic methods that can solve such large problems at once in a reasonable time due to the limitations of today´s computer technology For instance, with an Intel Core 2 Duo Processor and a RAM of 4 GB, the amount of time to reach the optimum solution is approximately 5 minutes for $n = 9$; 42 minutes for $n = 10$; 15 hours for $n = 11$ and over 200 hours for $n = 12$ using branch-and-cut exact method. By regression, it is estimated that solution time will reach up to 3.000 hours for $n = 13$. These processing times change according to data set depending on the cut off speed of exact method used.

- Roughly speaking, *if n decreases*, the gap between the global optimum and the solution found in this heuristic will increase. Reduced n would increase the risk of missing global optimum. If $n = 1$, there is no need to use any exact method or computer program since the sorted list of letters will directly be assigned to sorted list of keys one-to-one.

These two antagonist trends will compromise with each other with a pay off between quality of the result and the amount of time needed.

In order to make a good choice of n, the maximum number of letters to be assigned to keys throughout a phase, some speed tests were made. The data set were taken from the QAPLIB- A Quadratic Assignment Problem Library (QAPLIB Library).

Data sets for n = 8, n = 9, n = 10 and n = 11 were run on GAMS. There were no CPU time data sets for running OCOB for these n values in the in the used computers At Bogazici University Graduate Laboratory (Intel Core 2 Due Processor and a RAM of 4 GB or similar). So, the data sets were generated by fixing some assignments initially. For instance, assignment of first facility in Had12 data set was made to third location initially and fixed in order to generate a data set for *n* = 11. GAMS code could find the proven optimal solution.

Then, “chr12a” data set from Christofides was tested. The computer program on GAMS using Cplex could find the optimum solution in over 32 hours. Next, “nug12” from Nugent and “tai12a” from Taillard were tried, GAMS was able to find the optimum solution in 95 hours.

By the help of regression method, it is estimated that, GAMS can find optimal solution for *n* = 12 in 270 hours.

As a result, *n* is set equal to 12 for the stylus keyboard layout problem during phases in GAMS.

9.5. Step#3: Initially Fixed Assignments

The initially fixed assignments for the longitudinal keyboard were explained in detail in Section 7.3.1., and in Table 7.1. Since the letters *q*, *w*, and *x* as well as the keys *Key11*, *Key12* and *Key23* were omitted for the rest of the solution, the only fixed assignment is the assignment of blank character to Key33 (*which is space bar*). This assignment is due to ergonomics principles and should not be re-assigned during any phase of the solution.

Table 9.1. Fixed assignments for the longitudinal keyboard before solution phases

Assigned Key	Assigned Letter
Key33 (space bar)	blank

According to OCOB algorithm, Step#4 to Step#6 will be run for $n = 8,\ n = 9,\ n = 10,\ n = 11\ and\ n = 12.$

9.6. Step#4 to Step#6 for *n* = 8

Table B.11 shows the letters and keys in which phase to be assigned. In each phase, the initial assignments and also the assignments made in previous phases will be fixed. The revised *D* matrix for a phase will include only the keys which are being optimized in that phase and also the keys that are fixed before that phase. Rows and columns for other keys will be deleted. With the same manner, the revised *F* matrix for a phase will include only the letters which are being optimized in that phase and also the letters that are fixed before that phase. Rows and columns for other letters will be deleted.

9.6.1. Phase 1 for *n* = 8

The first unassigned 8 keys in the sorted list are {Key18, Key17, Key19, Key28, Key6, Key29, Key7, and Key16}. These keys are shown in Figure 9.1. Also, the first unassigned 8 letters are {a, e, i, n, r, l, ı and k}. The revised *D* matrix for Phase 1 will be 9 by 9 and is shown in Table 9.2 and the revised *F* matrix for Phase 1 will be 9 by 9 and is shown in Table 9.3.

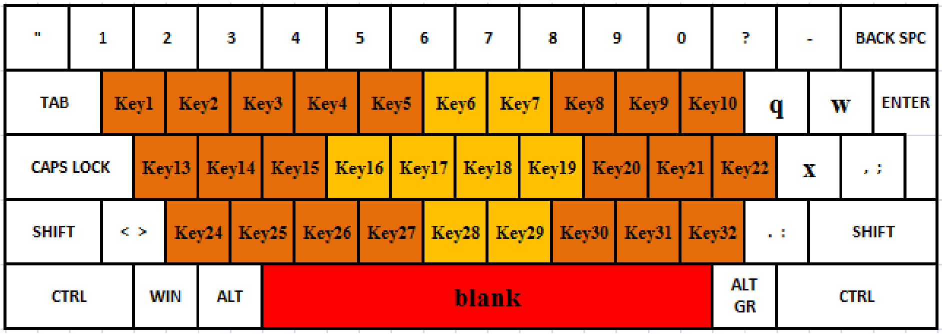


Figure 9.1. The keys to be assigned on Longitudinal Keyboard in Phase 1 for *n*=8

Table 9.2. Revised D matrix for longitudinal keyboard of Phase 1 for *n* = 8

	Key6	**Key7**	**Key16**	**Key17**	**Key18**	**Key19**	**Key28**	**Key29**	**Key33**
Key6	0.00	32.00	57.69	35.78	35.78	57.69	64.00	71.55	97.32
Key7	32.00	0.00	86.16	57.69	35.78	35.78	71.55	64.00	97.32
Key16	57.69	86.16	0.00	32.00	64.00	96.00	57.69	86.16	90.51
Key17	35.78	57.69	32.00	0.00	32.00	64.00	35.78	57.69	71.55
Key18	35.78	35.78	64.00	32.00	0.00	32.00	35.78	35.78	64.00
Key19	57.69	35.78	96.00	64.00	32.00	0.00	57.69	35.78	71.55
Key28	64.00	71.55	57.69	35.78	35.78	57.69	0.00	32.00	35.78
Key29	71.55	64.00	86.16	57.69	35.78	35.78	32.00	0.00	35.78
Key33	97.32	97.32	90.51	71.55	64.00	71.55	35.78	35.78	0.00

Table 9.3. Revised F matrix of Phase 1 for *n* = 8

	a	**e**	**ı**	**i**	**k**	**l**	**n**	**r**	**blank**
a	0.25226	0.02523	0.00581	0.53510	6.77751	7.25800	14.57494	17.05680	14.92655
e	0.09995	0.04302	0.00090	0.05989	5.63788	5.66063	10.59926	14.78841	18.07441
ı	0.00857	0.00092	0.01348	0.00349	2.50640	3.27661	10.62558	4.47752	8.75614
i	0.26587	0.10889	0.00000	0.10092	3.57825	7.77489	13.97641	10.35877	12.91463
k	7.24097	3.03882	2.17621	4.70968	0.35223	3.32630	0.14252	0.48860	8.03385
l	15.87727	13.21510	4.33392	7.09082	0.82879	1.78677	0.11042	0.00687	2.74906
n	4.16983	5.36326	5.38872	5.89959	0.50233	3.81887	0.27794	0.67244	20.90581
r	7.68159	5.06491	5.27850	7.25476	2.12022	3.24139	0.29131	0.08786	10.39171
blank	9.42801	5.14363	0.20857	7.96840	11.00211	0.52716	1.79343	1.35755	0.72292

Hence, the integer linear programming form of QAP model of Phase 1 will be:

$$\min \sum_{i=1}^{9} \sum_{j=1}^{9} \sum_{k=1}^{9} \sum_{p=1}^{9} f_{i,j} * d_{k,p} * x_{i,k} * x_{j,p} \tag{9.2}$$

$$\sum_{i=1}^{9} x_{i,j} = 1 \qquad 1 \leq j \leq 9 \tag{9.3}$$

$$\sum_{j=1}^{9} x_{i,j} = 1 \qquad 1 \leq i \leq 9 \tag{9.4}$$

$$x_{9,9} = 1 \tag{9.5}$$

The φ value of Phase 1 for $n = 8$ is;

$$\varphi = \sum_{i,j} f_{i,j} = 397.1949$$

This connotes that the 81 digraphs formed by these 9 letters account for 39.72 per cent of all digraphs in an average Turkish text.

GAMS/Cplex reached the proven optimal solution of 20,429.5944 for the Phase 1 in 26 seconds after 467,025 iterations and 80,570 nodes. The assignments of the optimal solution are listed by GAMS/Cplex as follows:

Table 9.4. Assignments after Phase 1 for *n* = 8

Key	**Letter**
Key6	e
Key7	l
Key16	ı
Key17	r
Key18	a
Key19	k
Key28	n
Key29	i
Key33	blank

$$\psi = \frac{z}{\varphi} = \frac{20{,}429.5944}{397.19491\ x\ 30} = 1.7145\ (key\ length/digraph)$$

This value expresses that, if a Turkish text involving only these 9 letters were typed; the average distance taken by the stylus would be 1.7145 key-lengths.

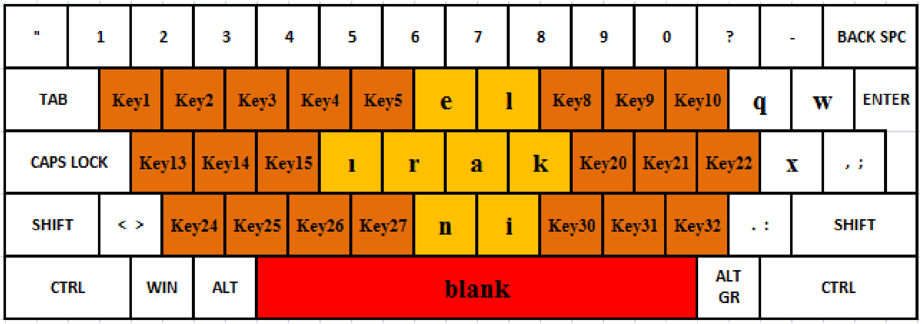


Figure 9.2. Assignments till the end of Phase 1 for *n*=8 for long. keyboard

9.6.2. Phase 2 for *n* = 8

The first unassigned 8 keys in the sorted list are {Key27, Key5, Key30, Key8, Key20, Key26, Key4, and Key15}. These keys are shown in Figure 9.3. Also, the first unassigned 8 letters are {d, t, m, s, y, u, o and b}. The *D* matrix for Phase 2 will be 17 by 17 and is revised from Table C.1 in Appendix C and the *F* matrix for Phase 2 will be 17 by 17 and is revised from Table B.8 in Appendix B.

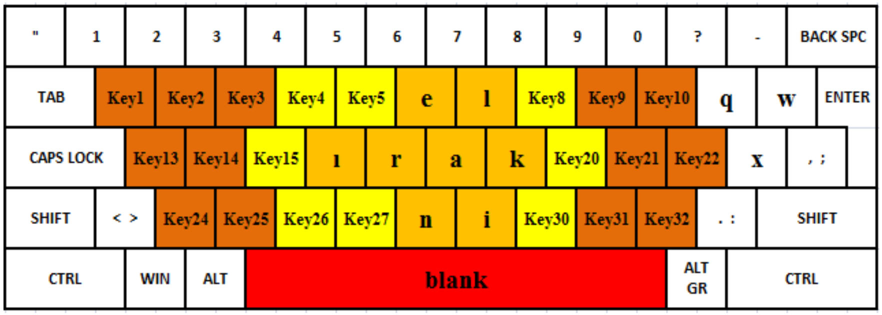


Figure 9.3. The keys to be assigned on Longitudinal Keyboard in Phase 2 for *n*=8

Hence, the integer linear programming form of QAP model of Phase 2 is shown on Table B.22.

The φ value of Phase 2 for $n = 8$ is;

$$\varphi = \sum_{i,j} f_{i,j} = 770.7061$$

This connotes that the 289 digraphs formed by these 17 letters account for 77.07 per cent of all digraphs in an average Turkish text.

GAMS/Cplex reached the proven optimal solution of 47,770.5513 for the Phase 2 in 13 seconds after 213,416 iterations and 44,062 nodes. The assignments of the optimal solution are listed by GAMS/Cplex as shown in Table 9.5.

Table 9.5. Assignments after Phase 2 for n = 8

Key4	s
Key5	t
Key6	e
Key7	l
Key8	m
Key15	o
Key16	ı
Key17	r
Key18	a
Key19	k
Key20	u
Key26	y
Key27	d
Key28	n
Key29	i
Key30	b
Key33	blank

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{47,770.5513}{770.7061 \; x \; 30} = 2.0661 \; (key\ length/digraph)$$

This value expresses that, if a Turkish text involving only these 17 letters were typed; the average distance taken by the stylus would be 2.0661 key-lengths.

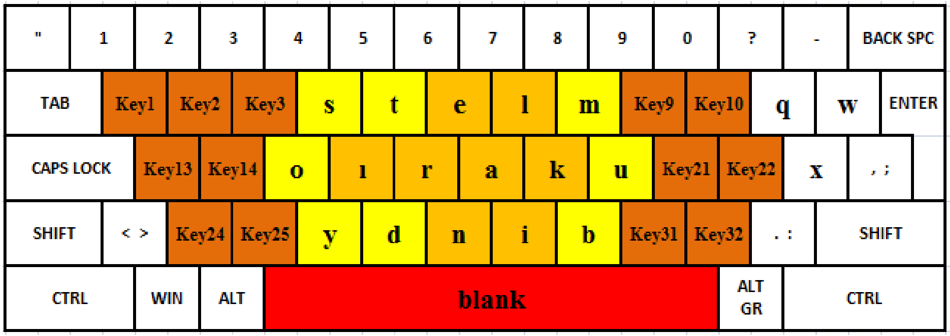


Figure 9.4. Assignments till the end of Phase 2 for *n*=8 for long. keyboard

9.6.3. Phase 3 for *n* = 8

The first unassigned 8 keys in the sorted list are {Key9, Key31, Key21, Key3, Key25, Key14, Key10, and Key32}. These keys are shown in Figure 9.5. Also, the first unassigned 8 letters are {ü, ş, z, g, v, ç, h and ğ}. The *D* matrix for Phase 3 will be 25 by 25 and is revised from Table C.1 in Appendix C and the *F* matrix for Phase 3 will be 25 by 25 and is revised from Table B.8 in Appendix B.

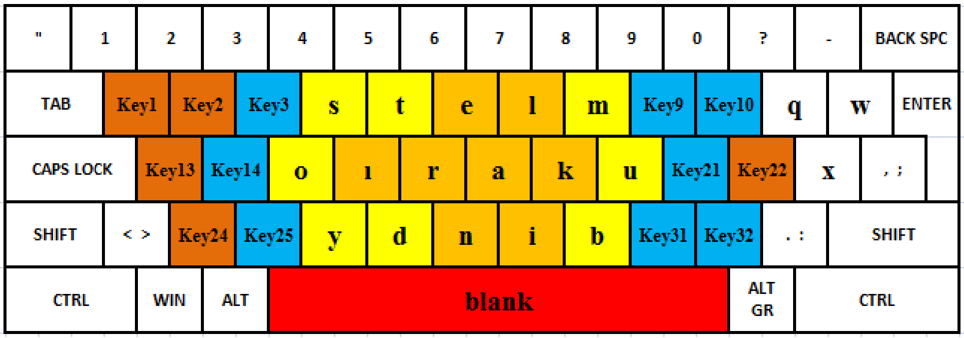


Figure 9.5. The keys to be assigned on Longitudinal Keyboard in Phase 3 for *n*=8

Hence, the integer linear programming form of QAP model of Phase 3 is shown on Table B.23.

The φ value of Phase 3 for $n = 8$ is;

$$\varphi = \sum_{i,j} f_{i,j} = 945.3562$$

This connotes that the 625 digraphs formed by these 25 letters account for 94.54 per cent of all digraphs in an average Turkish text. GAMS/Cplex reached the proven optimal solution of 67,369.158984 for the Phase 3 in 6 seconds after 49,948 iterations and 12,452 nodes. The assignments of the optimal solution are listed by GAMS/Cplex as in Table 9.6.

Table 9.6. Assignments after Phase 3 for n = 8

Key3	v
Key4	s
Key5	t
Key6	e
Key7	l
Key8	m
Key9	ş
Key10	z
Key14	ğ
Key15	o
Key16	ı
Key17	r
Key18	a
Key19	k
Key20	u
Key21	ü
Key25	ç
Key26	y
Key27	d
Key28	n
Key29	i
Key30	b
Key31	g
Key32	h
Key33	blank

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{67{,}369.1590}{945.3563\ x\ 30} = 2.3754\ (key\ length/digraph)$$

This value expresses that, if a Turkish text involving only these 25 letters were typed; the average distance taken by the stylus would be 2.3754 key-lengths.

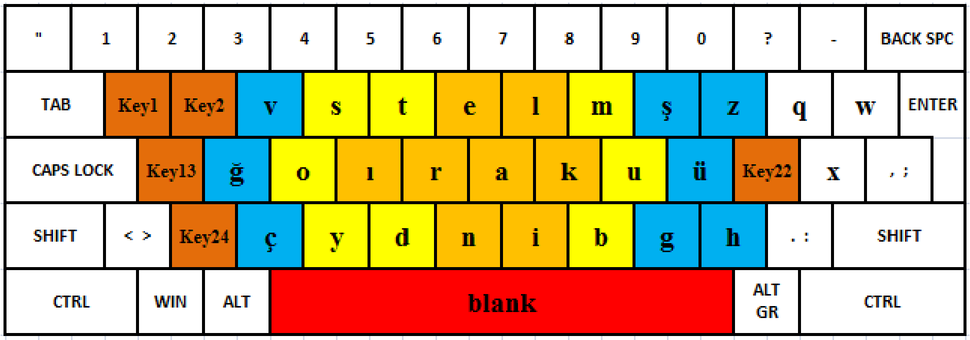


Figure 9.6. Assignments till the end of Phase 3 for *n*=8 for long. keyboard

9.6.4. Phase 4 for *n* = 8

Since there are 5 unassigned keys and $5 < n$, all unassigned keys are relaxed in Phase 4, which are {Key1, Key2, Key13, Key22 and Key24}. These keys are shown in Figure 9.7. Also, since there are 5 unassigned letters and $5 < n$, all unassigned letters are relaxed in Phase 4, which are {j, p, f, c and ö}. The *D* matrix for Phase 4 will be 30 by 30 and is revised from Table C.1 in Appendix C and the *F* matrix for Phase 4 will be 30 by 30 and is revised from Table B.8 in Appendix B.

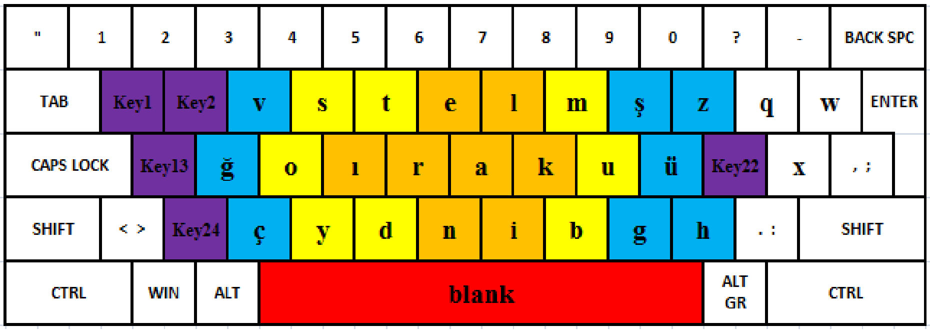


Figure 9.7. The keys to be assigned on Longitudinal Keyboard in Phase 4 for *n*=8

Hence, the integer linear programming form of QAP model of Phase 4 is shown on Table B.24.

GAMS/Cplex reached the proven optimal solution of 75,215.5595 for the Phase 4 in one second after 31 iterations and 0 nodes. The assignments of the optimal solution are listed by GAMS/Cplex as in Table 9.7.

Table 9.7. Assignments after Phase 4 for n = 8

Key1	j
Key2	p
Key3	v
Key4	s
Key5	t
Key6	e
Key7	l
Key8	m
Key9	ş
Key10	z
Key13	f
Key14	ğ
Key15	o
Key16	ı
Key17	r
Key18	a
Key19	k
Key20	u
Key21	ü
Key22	c
Key24	ö
Key25	ç
Key26	y
Key27	d
Key28	n
Key29	i
Key30	b
Key31	g
Key32	h
Key33	blank

The φ value of Phase 4 for n = 8 is;

$$\varphi = \sum_{i,j} f_{i,j} = 1{,}000.0000$$

This connotes that the 900 digraphs formed by these 30 letters account for 100 per cent of all digraphs in an average Turkish text (*since all letters of Turkish alphabet are assigned*).

Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{75{,}215.5595}{1{,}000.0000\ x\ 30} = 2.5072\ (key\ length/digraph)$$

This value expresses that, if a Turkish text was typed; the average distance taken by the stylus would be 2.5072 key-lengths.

Consequently, the optimal longitudinal stylus keyboard layout is designed for *n* = 8 with the average distance of 2.5072 key-lengths per digraph taken by the stylus, as shown in Figure 9.8.

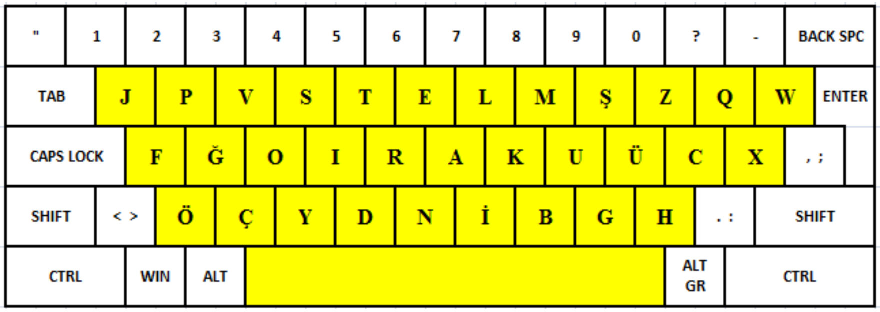


Figure 9.8. Optimal longitudinal stylus keyboard layout designed for *n* = 8

9.7. Summary of Phases for *n* = 9

Table B.12 in Appendix B shows the keys and letters in which phase to be assigned. Revised D matrices for phases are generated from Table C.1 in Appendix C and F matrices from Table B.8 in Appendix B.

The φ value of Phase 1 for n = 9 is;

$$\varphi = \sum_{i,j} f_{i,j} = 462.2371$$

GAMS/Cplex reached the proven optimal solution of 24,006.9422 for the Phase 1 in 297 seconds after 5,046,077 iterations and 725,270 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{24{,}006.9422}{462.2370\ x\ 30} = 1.7312\ (key\ length/digraph)$$

The φ value of Phase 2 for n = 9 is;

$$\varphi = \sum_{i,j} f_{i,j} = 824.3561$$

GAMS/Cplex reached the proven optimal solution of 52,768.3824 for the Phase 2 in 181 seconds after 2,671,393 iterations, 450,823 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{52{,}768.3824}{824.3561\ x\ 30} = 2.1337\ (key\ length/digraph)$$

The φ value of Phase 3 for n = 9 is;

$$\varphi = \sum_{i,j} f_{i,j} = 990.4223$$

GAMS/Cplex reached the proven optimal solution of 73,194.7165 for the Phase 3 in 6 seconds after 59,492 iterations and 13,931 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{73{,}194.7165}{990.4223\ x\ 30} = 2.4634\ (key\ length/digraph)$$

The φ value of Phase 4 for $n = 9$ is;

$$\varphi = \sum_{i,j} f_{i,j} = 1{,}000.0000$$

GAMS/Cplex reached the proven optimal solution of 74,785.4784 for the Phase 4 in 1 second after 2 iterations and 0 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{74{,}785.4784}{1{,}000.0000\ x\ 30} = 2.4928\ (key\ length/digraph)$$

Consequently, the optimal longitudinal stylus keyboard layout is designed for $n = 9$, with the average distance of 2.4928 key-lengths per digraph taken by the stylus, as shown in Figure 9.9.

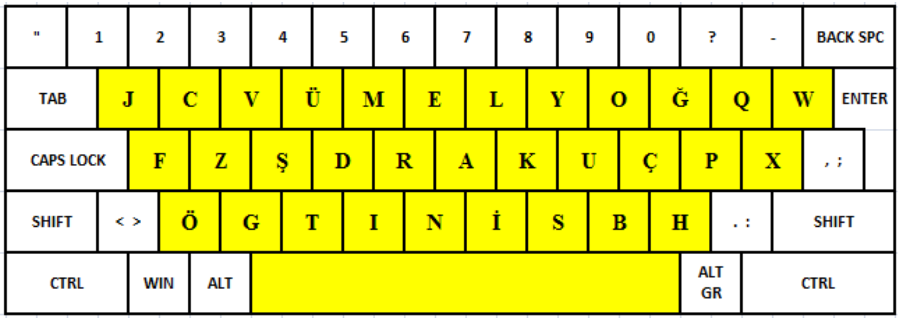


Figure 9.9. Optimal longitudinal stylus keyboard layout designed for $n = 9$

9.8. Summary of Phases for *n* = 10

Table B.13 in Appendix B shows the keys and letters in which phase to be assigned. Revised D matrices for phases are generated from Table C.1 in Appendix C and F matrices from Table B.8 in Appendix B.

The φ value of Phase 1 for *n* = 10 is;

$$\varphi = \sum_{i,j} f_{i,j} = 509.3843$$

GAMS/Cplex reached the proven optimal solution of 26,806.340769 for the Phase 1 in 3,691 seconds after 56,386,596 iterations and 7,242,570 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{26{,}806.3408}{509.3843\ x\ 30} = 1.7542\ (key\ length/digraph)$$

The φ value of Phase 2 for *n* = 10 is;

$$\varphi = \sum_{i,j} f_{i,j} = 867.9224$$

GAMS/Cplex reached the proven optimal solution of 57,817.7963 for the Phase 2 in 1,290 seconds after 15,478,866 iterations and 2,570,710 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{57{,}817.7963}{867.9224\ x\ 30} = 2.2205\ (key\ length/digraph)$$

The φ value of Phase 3 for *n* = 10 is;

$$\varphi = \sum_{i,j} f_{i,j} = 1{,}000.0000$$

GAMS/Cplex reached the proven optimal solution of 74,441.3102 for the Phase 3 in 1 second after 2,946 iterations and 839 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{74{,}441.3102}{1{,}000.0000\ x\ 30} = 2.4814\ (key\ length/digraph)$$

Consequently, the optimal longitudinal stylus keyboard layout is designed for *n* = 10, with the average distance of 2.4814 key-lengths per digraph taken by the stylus.

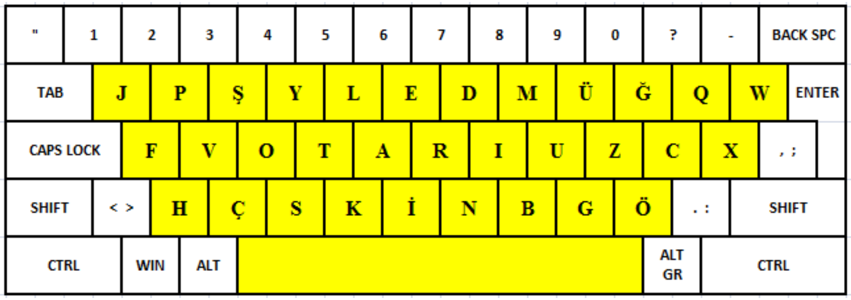


Figure 9.10. Optimal longitudinal stylus keyboard layout designed for *n* = 10

9.9. Summary of Phases for *n* = 11

Table B.14 in Appendix B shows the keys and letters in which phase to be assigned. Revised D matrices for phases are generated from Table C.1 in Appendix C and F matrices from Table B.8 in Appendix B.

The φ value of Phase 1 for *n* = 11 is;

$$\varphi = \sum_{i,j} f_{i,j} = 559.7184$$

GAMS/Cplex reached the proven optimal solution of 29,876.0146 for the Phase 1 in 51,387 seconds after 726,176,425 iterations and 79,445,452 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{29{,}876.0146}{559.7184 x\ 30} = 1.7792\ (key\ length/digraph)$$

The φ value of Phase 2 for $n = 11$ is;

$$\varphi = \sum_{i,j} f_{i,j} = 908.0566$$

GAMS/Cplex reached the proven optimal solution of 62,354.6627 for the Phase 2 in 13,177 seconds after 141,165,316 iterations and 22,554,754 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{62{,}354.6626}{908.0566\ x\ 30} = 2.2889\ (key\ length/digraph)$$

The φ value of Phase 3 for $n = 11$ is;

$$\varphi = \sum_{i,j} f_{i,j} = 1{,}000.0000$$

GAMS/Cplex reached the proven optimal solution of 74,920.7806 for the Phase 3 in less than 1 second after 787 iterations and 228 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{74{,}920.7806}{1{,}000.0000\ x\ 30} = 2.4974\ (key\ length/digraph)$$

Consequently, the optimal longitudinal stylus keyboard layout is designed for n =11, with the average distance of **2.4974** $(key\ length/digraph)$ taken by the stylus.

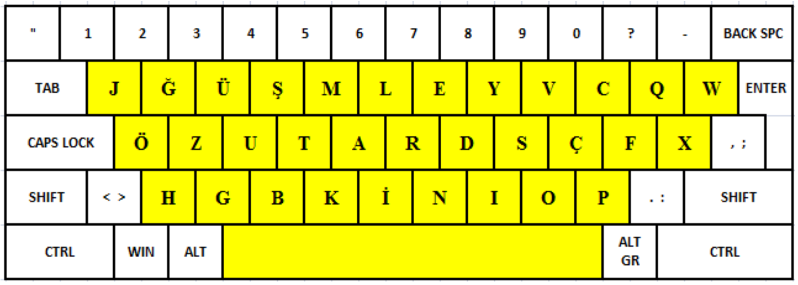


Figure 9.11. Optimal longitudinal stylus keyboard layout designed for n = 11

9.10. Summary of Phases for *n* = 12

Table B.15 in Appendix B shows the keys and letters in which phase to be assigned. Revised D matrices for phases are generated from Table C.1 in Appendix C and F matrices from Table B.8 in Appendix B.

The φ value of Phase 1 for n = 12 is;

$$\varphi = \sum_{i,j} f_{i,j} = 606.5202$$

GAMS/Cplex reached the proven optimal solution of 33,451.695720 for the Phase 1 in 557,487 seconds after 949,853,124 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{33{,}451.6957}{606.5202x\ 30} = 1.8385\ (key\ length/digraph)$$

The φ value of Phase 2 for n = 12 is;

$$\varphi = \sum_{i,j} f_{i,j} = 945.3562$$

GAMS/Cplex reached the proven optimal solution of 66,317.0646 for the Phase 2 in 78,293 seconds after 884,832,780 iterations and 142,546,501 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{66{,}317.0647}{945.3563x\ 30} = 2.3384\ (key\ length/digraph)$$

The φ value of Phase 3 for n = 12 is;

$$\varphi = \sum_{i,j} f_{i,j} = 1{,}000.0000$$

GAMS/Cplex reached the proven optimal solution of 73,953.6145 for the Phase 3 in less than 1 second after 27 iterations and 0 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{73{,}953.6145}{1{,}000.0000\ x\ 30} = 2.4651\ (key\ length/digraph)$$

Consequently, the optimal longitudinal stylus keyboard layout is designed for n = 12, with the average distance of 2.4651 ($key\ length/digraph$) taken by the stylus.

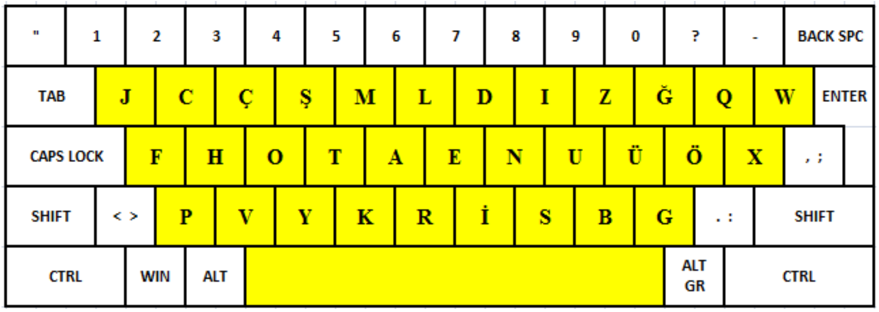


Figure 9.12. Optimal longitudinal stylus keyboard layout designed for n = 12

9.11. Selecting the Optimal Layout

Totally five layouts were designed according to OCOB heuristic. The last step of the algorithm is the selection of the layout which has the minimum average distance per digraph value. So, the layout which was designed when $n = 12$ is chosen as the optimum longitudinal stylus keyboard layout for Turkish language having average distance taken by the stylus per digraph value of 2.4651 $(key\ length/digraph)$, shown in Figure 9.12.

9.12. Analysing the Effect of *n*

Basically, as mentioned in Section 9.4; solution time increases extremely even if *n* increases by one. So, it is worth now to analyse the effect of *n* on the improvement of objective function, *average distance taken by the stylus per digraph.*

Since the most centralized keys were assigned to letters in early phases and centralization deteriorates phase by phase, the average distance per digraph increases phase by phase. However, this increment slows down in every phase which plots a graphic having a positive but decreasing first derivative.

If the cumulative density function of the frequencies of the digraphs, which are sorted in ascending order by their centralization rating, converges to 100 percent rapidly; than the slope of this graph will decrease more quickly.

The improvement in solution is generally less than six per thousand when *n* increases by one; however the solution CPU time increases by more than 13 times as shown in Figure 9.13.

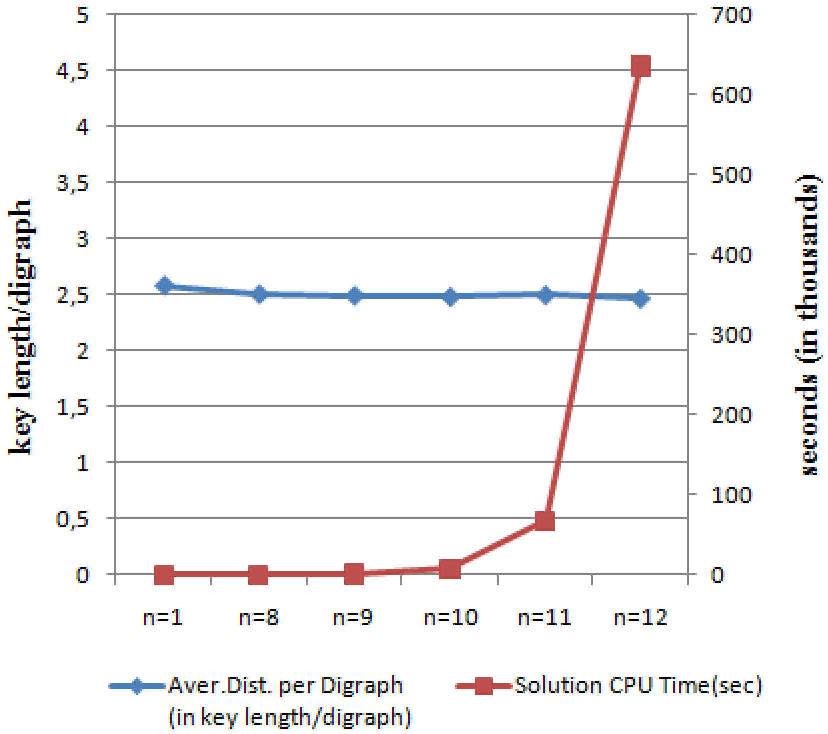


Figure 9.13. Solution improvement and solution CPU time versus *n* for long. keyboard

According to Figure 9.1, solution does not decrease much as *n* increases, especially for the values greater than 10 or 11.

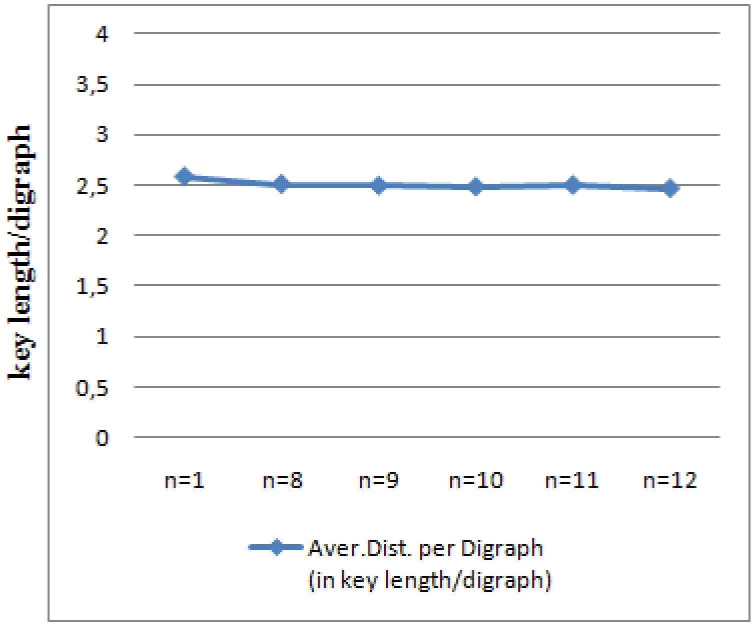


Figure 9.14. Solution improvement versus *n* for long. keyboard

So, it can be argued that “*there is a general trend that objective function decreases as n increases, but this trend does not follow a smooth line. In other words, there is no guarantee that solution for ‘n+1’ is less than solution for ‘n’. Also, after a point, the change in objective function is very little and can be omitted due to high solution CPU time for stylus keyboard optimization problems.*”

10. DESIGNING OPTIMUM SQUARE STYLUS KEYBOARD LAYOUT

In this chapter, the work to design the optimum square stylus keyboard layout for Turkish language using the OCOB heuristic is explained.

10.1. Introduction

In the previous chapter, the design of optimum stylus keyboard layout for Turkish language has been designed for longitudinal keyboard. Since, the application of OCOB heuristic is explained during designing longitudinal keyboard in detail; only the core and results of the stages will be discussed for square layout design.

10.2. Step#1: Sorting the Keys and Letters

The first step of OCOB starts with sorting the keys in ascending order by centralization ratings and the letters in descending order by connection ratings.

10.2.1. Sorting the Keys

30 keys are sorted in ascending order upon their centralization indexes. Since the square keyboard model is symmetric, there will be keys which have equal centralization indexes. In this case, sorting is made randomly. The sorted list of keys of square keyboard can be seen in Table B.16 in Appendix B.

10.2.2. Sorting the Letters

Since the alternation in keyboard shape type does not affect the sorting of letters, the sorted list of letters for square layout optimization will be the same with the longitudinal layout optimization. So, the sorted list of letters according to connection index is shown in Table B.9 in Appendix B. Hereafter, both the centralization and connection indexes will not be used; they were just the tools to sort the keys and letters within each other.

10.3. Step#2: Setting *n*

Since, the quadratic assignment problem model, computer codes and hardware environment is unchanged; n is set equal to the value, which is used during longitudinal keyboard optimization. As a result, n is set equal to 12 for the square stylus keyboard layout problem during phases in GAMS.

10.4. Step#3: Initially Fixed Assignments

As stated in Chapter 7 in detail, all of the keys of square shape keyboard model are identical. Hence, there will not be any initial assignment for blank character. Just the solution of the model will assign the blank character and hence, the space bar will be the key to which the blank character is assigned.

According to OCOB algorithm, Step#4 to Step#6 will be run for $n = 8,\ n = 9,\ n = 10,\ n = 11\ and\ n = 12$.

10.5. Summary of Phases for *n* = 8

Table B.17 in Appendix B shows the keys and letters in which phase to be assigned. Revised D matrices for phases are generated from Table B.2 in Appendix B and F matrices from Table B.8 in Appendix B.

The φ value of Phase 1 for *n* = 8 is;

$$\varphi = \sum_{i,j} f_{i,j} = 334.7322$$

GAMS/Cplex reached the proven optimal solution of 14,740.4600 for the Phase 1 in 27 seconds after 518,824 iterations and 80,638 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{14{,}740.4600}{334.7322x\ 30} = 1.4679\ (key\ length/digraph)$$

The φ value of Phase 2 for *n* = 8 is;

$$\varphi = \sum_{i,j} f_{i,j} = 728.2434$$

GAMS/Cplex reached the proven optimal solution of 38,588.4465 for the Phase 2 in 21 seconds after 417,322 iterations and 68,964 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{38{,}588.4465}{728.2434x\ 30} = 1.7663\ (key\ length/digraph)$$

The φ value of Phase 3 for *n* = 8 is;

$$\varphi = \sum_{i,j} f_{i,j} \; = 926.9962$$

GAMS/Cplex reached the proven optimal solution of 57,545.0205 for the Phase 3 in 2 seconds after 23,893 iterations and 5,415 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{57{,}545.0205}{926.9961x\ 30} = 2.0692\ (key\ length/digraph)$$

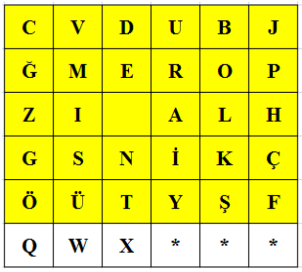
The φ value of Phase 4 for n = 8 is;

$$\varphi = \sum_{i,j} f_{i,j} \; = 1{,}000.0000$$

GAMS/Cplex reached the proven optimal solution of 64,402.7963 for the Phase 4 in less than 1 second after 67 iterations and 11 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{64{,}402.7963}{1{,}000.0000x\ 30} = 2.1468\ (key\ length/digraph)$$

Consequently, the optimal square stylus keyboard layout is designed for n = 8, with the average distance of 2.1468 $(key\ length/digraph)$ taken by the stylus, as shown in Figure 10.1.



C	V	D	U	B	J
Ğ	M	E	R	O	P
Z	I		A	L	H
G	S	N	İ	K	Ç
Ö	Ü	T	Y	Ş	F
Q	W	X	*	*	*

Figure 10.1. Optimal square stylus keyboard layout designed for n = 8

10.6. Summary of Phases for *n* = 9

Table B.18 in Appendix B shows the keys and letters in which phase to be assigned. Revised D matrices for phases are generated from Table B.2 in Appendix B and F matrices from Table B.8 in Appendix B.

The φ value of Phase 1 for *n* = 9 is;

$$\varphi = \sum_{i,j} f_{i,j} = 397.1949$$

GAMS/Cplex reached the proven optimal solution of 18,093.9282 for the Phase 1 in less than 319 second after 5,553,726 iterations and 725,700 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{18{,}093.9282}{397.1949x\ 30} = 1.5185\ (key\ length/digraph)$$

The φ value of Phase 2 for *n* = 9 is;

$$\varphi = \sum_{i,j} f_{i,j} = 795.6116$$

GAMS/Cplex reached the proven optimal solution of 45,438.1095 for the Phase 2 in 264 second after 4,091,907 iterations and 666,369 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{45{,}438.1095}{795.6116x\ 30} = 1.9037\ (key\ length/digraph)$$

The φ value of Phase 3 for *n* = 9 is;

$$\varphi = \sum_{i,j} f_{i,j} = 976.1370$$

GAMS/Cplex reached the proven optimal solution of 60,519.4945 for the Phase 3 in 3 second after 17,854 iterations and 4,069 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{60{,}519.4945}{976.1370x\ 30} = 2.0666\ (key\ length/digraph)$$

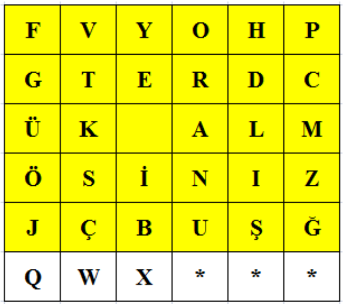
The φ value of Phase 4 for n = 9 is;

$$\varphi = \sum_{i,j} f_{i,j} = 1{,}000.0000$$

GAMS/Cplex reached the proven optimal solution of 62,964.5663 for the Phase 4 in less than 1 second after 12 iterations and 0 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{62{,}964.5663}{1{,}000.0000x\ 30} = 2.0988\ (key\ length/digraph)$$

Consequently, the optimal square stylus keyboard layout is designed for n = 9, with the average distance of 2.0988 $(key\ length/digraph)$ taken by the stylus, as shown in Figure 10.2.



F	**V**	**Y**	**O**	**H**	**P**
G	**T**	**E**	**R**	**D**	**C**
Ü	**K**		**A**	**L**	**M**
Ö	**S**	**İ**	**N**	**I**	**Z**
J	**Ç**	**B**	**U**	**Ş**	**Ğ**
Q	**W**	**X**	*	*	*

Figure 10.2. Optimal square stylus keyboard layout designed for n = 9

10.7. Summary of Phases for *n* = 10

Table B.19 in Appendix B shows the keys and letters in which phase to be assigned. Revised D matrices for phases are generated from Table B.2 in Appendix B and F matrices from Table B.8 in Appendix B.

The φ value of Phase 1 for $n = 10$ is;

$$\varphi = \sum_{i,j} f_{i,j} = 462.2371$$

GAMS/Cplex reached the proven optimal solution of 22,066.3344 for the Phase 1 in 3,934 second after 61,235,641 iterations and 7,254,883 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{22{,}066.3344}{462.2371x\ 30} = 1.5913\ (key\ length/digraph)$$

The φ value of Phase 2 for $n = 10$ is;

$$\varphi = \sum_{i,j} f_{i,j} = 846.5460$$

GAMS/Cplex reached the proven optimal solution of 49,495.9891 for the Phase 2 in 2,232 second after 28,942,446 iterations and 4,374,842 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{49{,}495.9891}{846.5460x\ 30} = 1.9489\ (key\ length/digraph)$$

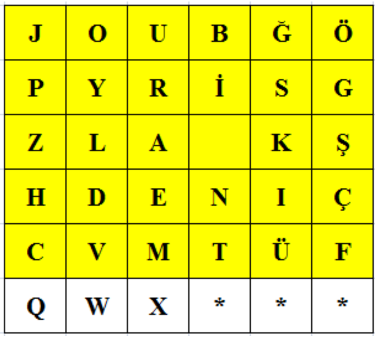
The φ value of Phase 3 for $n = 10$ is;

$$\varphi = \sum_{i,j} f_{i,j} = 1{,}000.0000$$

GAMS/Cplex reached the proven optimal solution of 63,141.6951 for the Phase 3 in 9 seconds after 69,548 iterations and 15,571 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{63{,}141.6951}{1{,}000.000x\ 30} = 2.1047\ (key\ length/digraph)$$

Consequently, the optimal square stylus keyboard layout is designed for n = 10, with the average distance of $2.1047\ (key\ length/digraph)$ taken by the stylus, as shown in Figure 10.3.



J	**O**	**U**	**B**	**Ğ**	**Ö**
P	**Y**	**R**	**İ**	**S**	**G**
Z	**L**	**A**		**K**	**Ş**
H	**D**	**E**	**N**	**I**	**Ç**
C	**V**	**M**	**T**	**Ü**	**F**
Q	**W**	**X**	*	*	*

Figure 10.3. Optimal square stylus keyboard layout designed for *n* = 10

10.8. Summary of Phases for *n* = 11

Table B.20 in Appendix B shows the keys and letters in which phase to be assigned. Revised D matrices for phases are generated from Table B.2 in Appendix B and F matrices from Table B.8 in Appendix B.

The φ value of Phase 1 for n = 11 is;

$$\varphi = \sum_{i,j} f_{i,j} \ = 509.3843$$

GAMS/Cplex reached the proven optimal solution of 24,761.4348 for the Phase 1 in 51,042 seconds after 679,195,974 iterations and 79,585,811 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{24{,}761.4348}{509.3843x\ 30} = 1.6204\ (key\ length/digraph)$$

The *φ* value of Phase 1 for *n* = 11 is;

$$\varphi = \sum_{i,j} f_{i,j} \ = 888.7202$$

GAMS/Cplex reached the proven optimal solution of 52,587.5988 for the Phase 2 in 15,444 seconds after 152,955,742 iterations and 24,015,644 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{52{,}587.5988}{888.7202x\ 30} = 1.9724\ (key\ length/digraph)$$

The *φ* value of Phase 3 for ı = 11 is;

$$\varphi = \sum_{i,j} f_{i,j} \ = 1{,}000.0000$$

GAMS/Cplex reached the proven optimal solution of 62,773.3477 for the Phase 3 in less than 1 second after 1,234 iterations and 301 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{62{,}773.3477}{1{,}000.0000x\ 30} = 2.0924\ (key\ length/digraph)$$

Consequently, the optimal square stylus keyboard layout is designed for n = 11, with the average distance of $2.0924\ (key\ length/digraph)$ taken by the stylus, as shown in Figure 10.4.

C	**V**	**Y**	**O**	**Z**	**J**
Ö	**D**	**E**	**L**	**M**	**P**
G	**N**		**A**	**K**	**Ü**
Ç	**I**	**İ**	**R**	**T**	**H**
Ğ	**B**	**S**	**U**	**Ş**	**F**
Q	**W**	**X**	*	*	*

Figure 10.4. Optimal square stylus keyboard layout designed for *n* = 11

10.9. Summary of Phases for *n* = 12

Table B.21 in Appendix B shows the keys and letters in which phase to be assigned. Revised D matrices for phases are generated from Table B.2 in Appendix B and F matrices from Table B.8 in Appendix B.

The φ value of Phase 1 for n = 12 is;

$$\varphi = \sum_{i,j} f_{i,j} = 559{,}7184$$

GAMS/Cplex reached the proven optimal solution of 28,072.7775 for the Phase 1 in 777,889 seconds after 952,999,494 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{28{,}072.7775}{559{,}7184x\ 30} = 1.6718\ (key\ length/digraph)$$

The φ value of Phase 2 for n = 12 is;

$$\varphi = \sum_{i,j} f_{i,j} = 926{,}9962$$

GAMS/Cplex reached the proven optimal solution of 55,968.1928 for the Phase 2 in 118,212 seconds after 1,369,475,027 iterations, 211,440,089 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{55{,}968.1928}{926{,}992x\ 30} = 2.0125\ (key\ length/digraph)$$

The φ value of Phase 3 for n = 12 is;

$$\varphi = \sum_{i,j} f_{i,j} = 1{,}000.0000$$

GAMS/Cplex reached the proven optimal solution of 62,717.5530 for the Phase 3 in less than 1 second after 1,369,475,027 iterations and 211,440,089 nodes. Also,

$$\boldsymbol{\psi} = \frac{\boldsymbol{z}}{\boldsymbol{\varphi}} = \frac{62{,}717.5530}{1{,}000.0000x\ 30} = 2.0906\ (key\ length/digraph)$$

Consequently, the optimal square stylus keyboard layout is designed for n = 12, with the average distance of $2.0906\ (key\ length/digraph)$ taken by the stylus, as shown in Figure 10.5.

Ğ	B	S	U	Ş	F
C	I	İ	R	T	P
Z	N		A	K	Ü
G	D	E	L	M	H
Ö	V	Y	O	Ç	J
Q	W	X	*	*	*

Figure 10.5. Optimal square stylus keyboard layout designed for n = 12

10.10. Step#8: Selecting the Optimal Layout

Totally five layouts were designed according to OCOB heuristic. The last step of the algorithm is the selection of the layout which has the minimum average distance per digraph value. So, the layout which was designed when $n = 12$ is chosen as the optimum square stylus keyboard layout for Turkish language having average distance taken by the stylus per digraph value of 2.0906 ($key\ length/digraph$), shown in Figure 10.5.

10.11. Analysing the Effect of *n*

Basically, as mentioned in Section 9.4; solution time increases extremely even if n increases by one. So, it is worth now to analyse the effect of n on the improvement of objective function, *average distance taken by the stylus per digraph.*

Since the most centralized keys were assigned to letters in early phases and centralization deteriorates phase by phase, the average distance per digraph increases phase by phase. However, this increment slows down in every phase which plots a graphic having a positive but decreasing first derivative.

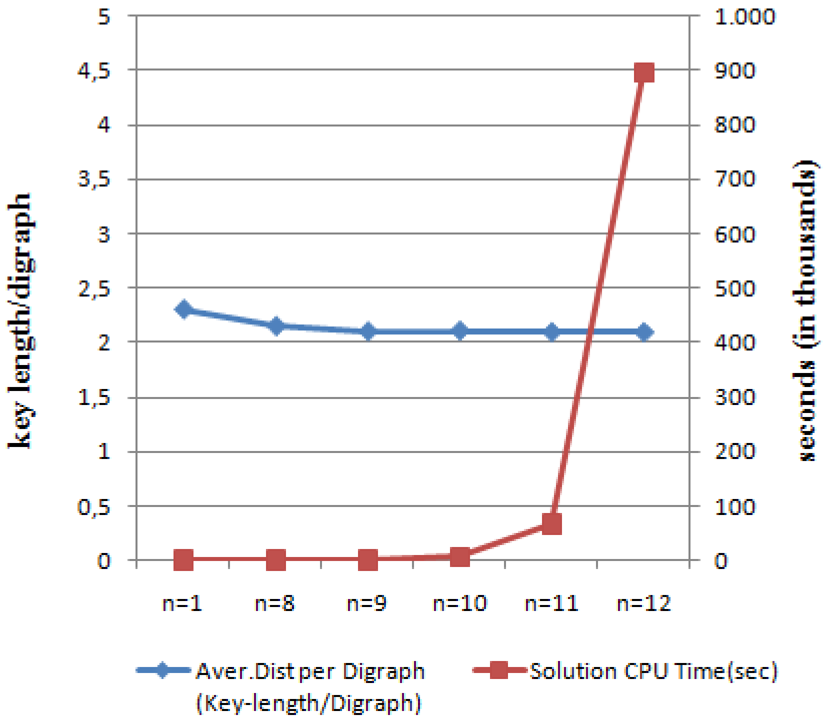


Figure 10.6. Solution improvement and solution CPU time versus *n* for square keyboard

The improvement in solution is generally less than six per thousand when *n* increases by one; however the solution CPU time increases by more than 13 times.

According to Figure 10.7, solution does not improve much as *n* increases, especially for the values greater than 10 or 11.

So, it can be argued that "*there is a general trend that objective function decreases as n increases, but this trend does not follow a smooth line. In other words, there is no guarantee that solution for 'n+1' is less than solution for 'n'. Also, after a point, the change in objective function is very little and can be omitted due to high solution CPU time for stylus keyboard optimization problems.*"

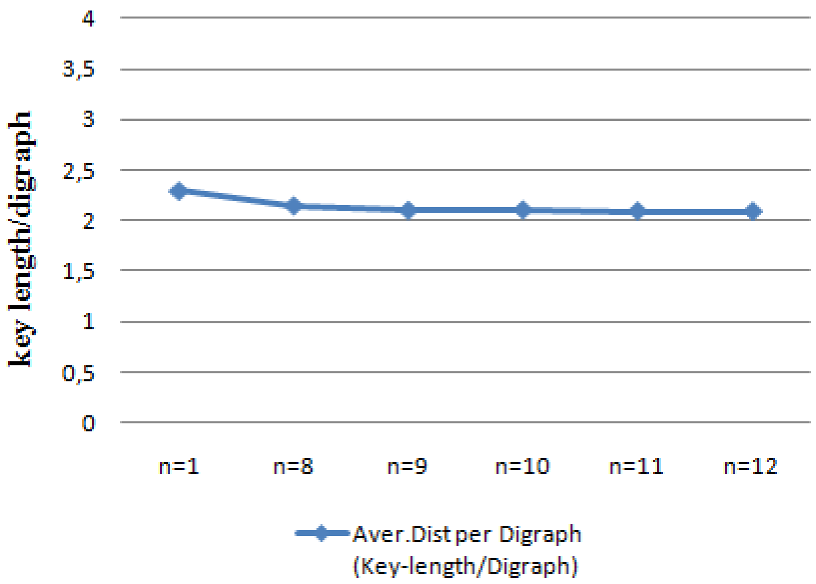


Figure 10.7. Solution improvement versus *n* for square keyboard optimization

11. COMPARISON OF LAYOUTS BY COMPUTER AIDED SIMULATION

In this chapter, stylus typing is simulated by the help of computer in order to compare the new generated longitudinal and square layouts. The distance per digraph taken by a simulated single finger robot is analysed by using randomly selected Turkish e-texts.

11.1. Introduction

The next step after designing stage of the layouts is the comparison of these layouts with the layouts which are currently being used for Turkish language. Test methods with their numerical performance criterion are:

- Computer aided tests (Simulation): Evaluation by using Turkish e-texts measuring the *average distance per digraph taken by the stylus*.
- Expert user' typing speed: Comparing the layouts with Fitts' Law
- Subject tests: Evaluation of the layouts by plotting learning curves of subjects.

This chapter describes the first of the comparison tests: computer aided simulations

11.2. Expected Values of Average Distances per Digraph

With the statistically generated flow matrix (F) shown in Table B.8 in Appendix B, the expected values of the average distance taken by the stylus for any layout can be computed using the following formula:

$$\frac{\sum_{i,j=1}^{n} f_{i,j} * d_{\pi(i),\pi(j)}}{\sum_{i,j=1}^{n} f_{i,j}} \tag{11.1}$$

where i,j ∈ {a, b, c, ç, ...,z, blank} and **π** defines key assignment of letters in any layout. Shortly, we should multiply the *F* matrix of Turkish language with the *D* matrix of the layout that we want to analyse. This multiplication gives the expected value of average distance per digraph taken by the stylus on that layout.

Currently used stylus keyboards for Turkish language are the QWERTY and the F layout which are shown in the Figures 11.1 and 11.2.

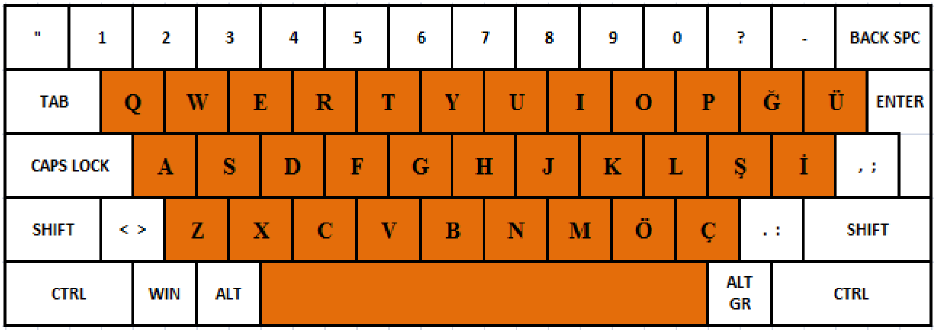


Figure 11.1. Modified version of QWERTY layout for Turkish language

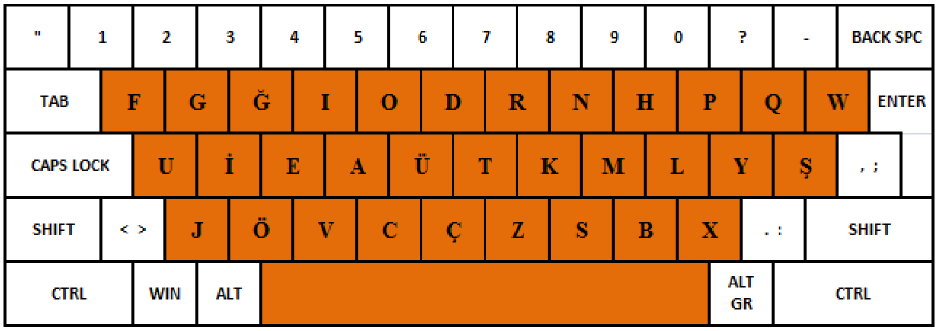


Figure 11.2. F layout designed specifically for Turkish language

The expected average distance per digraph when the defined formulas applied are computed as in Table 11.1.

Table 11.1. Expected values of average distance per digraph for four layouts

Layout	**Average Distance per Digraph (in Key-length)**
QWERTY	4.1906
F	4.0848
New Longitudinal	2.4651
New Square	2.0906

This connotes that if a Turkish text, in adequate length, is written by using the stylus keyboards with the layout QWERTY, F, New Longitudinal Layout and New Square Layout; then the average distance taken by the stylus will be 4.1906, 4.0848, 2.4651 and 2.0906 key-lengths, respectively. Adequate length comes from the "power of large numbers" theory in statistics. This theory implies that the statistical value of a sample converges to expected values if the sample size gets larger. This theory is generally illustrated by the ratio of tail or head when tossing a coin 4 times, 10 times and 1,000 times.

Of course, when a random Turkish e-text of small size is tested, the measured average distance per digraph may be far away from these expected values. To overcome this situation, random e-texts involving more than 1,000 digraphs will be used.

Theoretically, in the view of average distance to be taken by the stylus while typing a Turkish text, the New Longitudinal layout is 41.2 per cent and the New Square layout is 50.1 per cent more efficient than QWERTY layout. Also, the New Longitudinal layout is 39.7 per cent and the New Square layout is 48.8 per cent more efficient than F layout. Yet, F layout is slightly more efficient than QWERTY layout with 2.59 per cent. Hence, QWERTY is said to be the least effective layout among all.

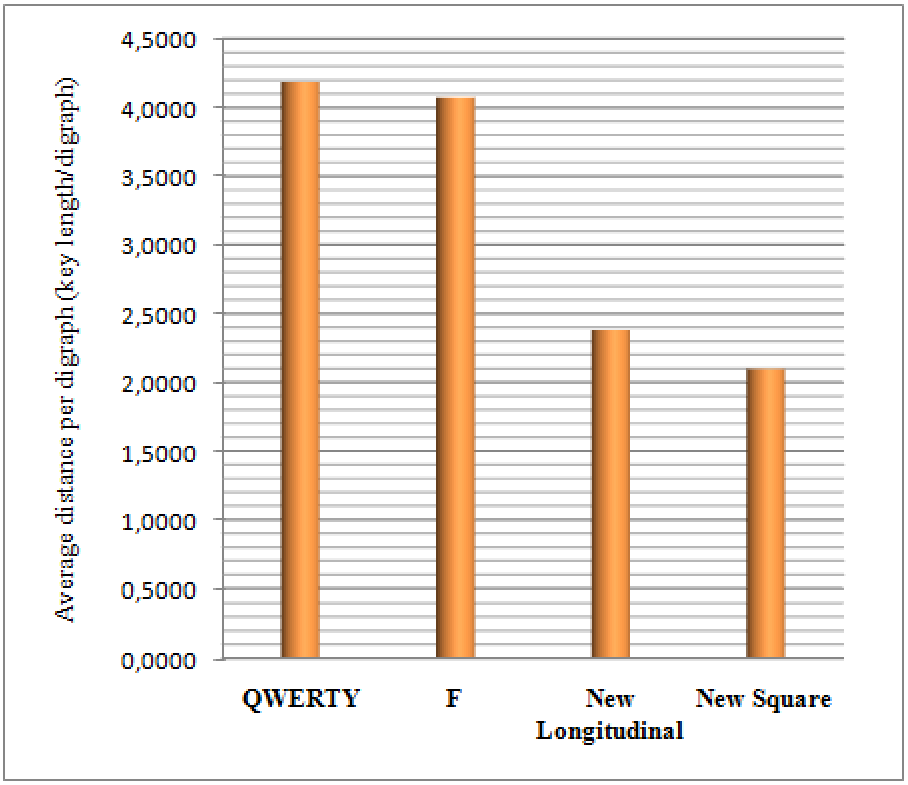


Figure 11.3. Graphically depicted expected average distance per digraph values

11.3. Simulation Illustration with a Single-Finger Robot

To make the simulation method clearly understood, a robotic hand that is programmed to type a loaded e-text using a physical keyboard by stroking its single finger on the centre of the keys may be illustrated as in Figure 11.4. The layout of letters for a keyboard will be installed to the robot, and the robotic single finger will stroke the related keys in the sequence of the letters of the given e-text. This robotic hand will not stroke any key except for the 30 keys. It will give no attention to punctuation marks and foreign letters, if there is any in the text.

After typing the whole of the given e-text, the robot will report the total distance that its top of the finger had taken. Dividing this reported total distance by the number of digraphs in the text, “the average distance per digraph” for the given e-text on that layout will be computed. If a simulation series with 20 different random e-texts is planned for the four layouts, then this robotic arm will be programmed and operated 80 times.

Figure 11.4. Simulation illustration for stylus typing with a single finger robot

To illustrate the operation of the robotic arm, think of typing “***GOL AT***” (*which means “to score a goal” in Turkish*) in QWERTY layout. The robotic finger will take an initial position just over the first letter of the e-text, which is “G”, before the operation starts. Then the tip of the finger will go and stroke the keys labelled “O”, then “L”, then “blank”, then “A” and finally “T”. After the completion of typing operation, the robot will report the total distance as 556.44 units according to our longitudinal layout model, which has square shaped keys of 30-unit-length and 2-units-width borders between keys. Since there are 5 digraphs in this sample text, the average distance per digraph taken by the robotic finger will be 556.44/5≈111.29 units. Finally, by dividing this value by the length of a unique key, which is 30, the average distance to type GOL AT will be calculated as 111.29/30 = 3.71 key-length. This value will be 3.42, 3.32 and 1.71 for F layout, New Longitudinal layout and New Square layout, respectively, if these layouts are installed to the robot.

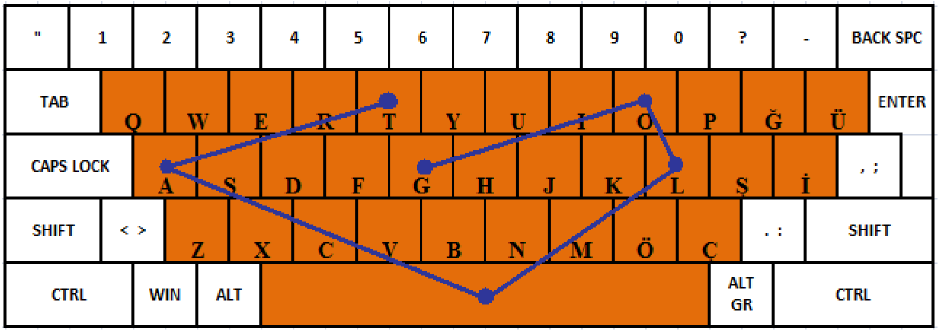


Figure 11.5. Illustration of typing “GOL AT” on QWERTY layout by stylus

11.4. Simulation Method

Due to technical deficiencies, this simulation could not be run in robotics laboratory; but instead, it can be run in computer environment, so called computer aided test.

Analysing a chunk of text letter by letter, in order to compute the total distance, takes long CPU time and memory use. For this reason, a new technique to compute the total distance is improved. This new technique depends on counting each of the digraph types in the given e-text. Next operation is the multiplication of these digraph counts with the distance of that digraph for each of four layouts.

For every e-text, a digraph table involving 900 different digraphs is created. Then, the given e-text is searched in detail to find how many times each of these digraphs is seen in that text. The distance of any digraph for every layout is computed. Later, these digraph counts are multiplied by the distance related that digraph. Finally, the sum of these products is divided by the number of digraphs, which gives “the average distance per digraph” for that layout.

To illustrate this computation technique, the text “GOL AT” will be analysed again. This e-text involves 5 different digraphs. These digraphs and the distance of the letters in that digraph on QWERTY layout are as follows:

- 1. Digraph <go> is seen 1 time and the distance for <go> is the distance between Key17 and Key9 which is 116.48 units
- 2. Digraph <ol> is seen 1 time and the distance for <ol> is the distance between Key9 and Key21 which is 35.78 units
- 3. Digraph <l_> is seen 1 time and the distance for <l_> is the distance between Key21 and Key33 which is 115.38 units
- 4. Digraph <_a> is seen 1 time and the distance for <_a> is the distance between Key33 and Key13 which is 172.33 units
- 5. Digraph <at> is seen 1 time and the distance for <at> is the distance between Key13 and Key5 which is 116.48 units

The total distance is = 1*116.48+1*35.78+1*115.38+1*172.33+1*116.48 = 556.45
The average distance per digraph will be 556.45/5= 111.29 which is the same with the result found by the robotic single-finger illustration. So, both techniques give the same result.

11.5. Simulation Results and Analysis

After a simulation series of 20 runs, the obtained detailed simulation results are shown in Table 11.2. The 20 e-texts used were chosen randomly from different newspapers, magazines and internet portals ranging from different subjects such as literary, education, science and politics.

According to simulation results shown in Table 11.2., the relative gaps between the simulation results and the expected values are less than one per cent for all of four layouts. Also, the ratios (*Standard Deviation/Mean*) are 1.67 per cent, 1.52 per cent, 1.94 per cent and

1.16 per cent for QWERTY, F, New Longitudinal and New Square layouts, respectively, which can be qualified as “acceptable”.

Table 11.2. Simulation results

Text #	Number of Digraphs in Text	Number of Words in Text	Average Distance per Digraph (in key length)			
			Q Layout	F Layout	New Longitudinal Layout	New Square Layout
1	5,540	853	4.1646	4.0834	2.5768	2.1632
2	3,221	434	4.1964	4.0037	2.4016	2.0831
3	2,213	327	4.2440	4.1186	2.4385	2.1033
4	6,135	814	4.1091	4.0438	2.5274	2.1175
5	11,810	1,787	4.1940	4.0941	2.5368	2.1064
6	12,600	1,778	4.2381	4.0784	2.4509	2.0802
7	3,686	530	4.1281	4.1621	2.4873	2.0772
8	5,335	803	4.2464	4.1121	2.4682	2.0868
9	4,249	608	4.0802	4.1430	2.4856	2.0876
10	2,845	389	4.1620	4.1114	2.5328	2.1125
11	6,815	934	4.2716	4.1824	2.4766	2.0731
12	24,860	3,124	4.2315	4.1191	2.4560	2.0816
13	4,494	616	4.2027	4.0743	2.4040	2.0575
14	4,616	599	4.1923	4.1808	2.4705	2.0853
15	2,536	320	4.1398	4.2628	2.4406	2.1208
16	5,726	751	4.3489	4.1113	2.4380	2.0784
17	3,706	492	4.3296	4.1117	2.4055	2.0905
18	6,388	898	4.1074	4.0821	2.4159	2.0625
19	10,772	1,468	4.2153	4.1150	2.4565	2.1164
20	24,603	3,474	4.1881	3.9817	2.4865	2.1021
Simulation Mean			4.1995	4.1086	2.4678	2.0943
St. Deviation of Simulations			0.0703	0.0623	0.0479	0.0242
Expected Mean Value			4.1906	4.0848	2.4651	2.0906
Relative gap between Expected and Simulation Mean Values (in per cent)			**0.21**	**0.58**	**0.11**	**0.17**

12. ESTIMATING TYPING SPEEDS OF EXPERT USERS BY FITTS' LAW

In this chapter, typing speeds of expert users of new generated longitudinal and square layouts are estimated by using Fitts' Law, which is explained in Chapter 3 in detail. These estimations are also made for QWERTY and Turkish F layouts to compare the typing speeds.

12.1. Introduction

Whereas the existence of numerous version of Fitts' Law, most commonly used version for stylus keyboard layout analysing can be written as Zhang and friends' formula as stated in Chapter 3. They defend that according to Fitts', the mean time to move the tapping stylus from one key i to another j for a given distance ($\boldsymbol{D_{ij}}$) and key width ($\boldsymbol{W_j}$) is

$$t_{ij} = a + b\, log_2\left(\frac{D_{ij} + W_j}{W_j}\right) \tag{12.1}$$

Or equivalently,

$$t_{ij} = a + b\, log_2\left(\frac{D_{ij}}{W_j} + 1\right) \tag{12.2}$$

where a and b are empirically determined coefficients. Main idea of Fitts' Law is trying to establish a linear equation that fits the empirical data best in the format of $y = a + bx$, as depicted in Figure 12.1.

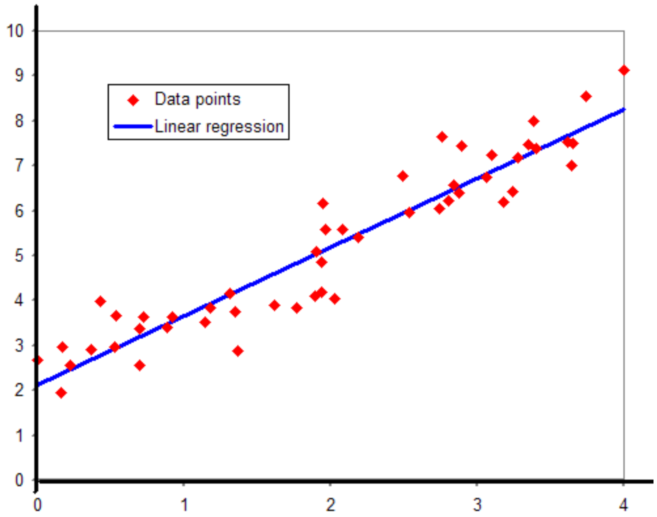


Figure 12.1. A sample for fitting a line to the empirical data

12.2. Calculating Exact Value of W_j in Stylus Keyboards

To calculate the exact value of W_j**,** following graphical depiction of a stylus keyboard layout on the x-y coordinate system used. Think of tapping key *j* after key *i*. In this case, the pointer (*say cursor of mouse*) will start its move from key *i* to key *j* with an approach angle different from 0^0 or 90^0. Let the x-y coordinates of key *i* be $(\boldsymbol{x_i}, \boldsymbol{y_i})$ and of key *j* be $\left(\boldsymbol{x_j}, \boldsymbol{y_j}\right)$**.** Also, let one edge of the squared shaped keys be in a length of l as seen in Figure 12.2.

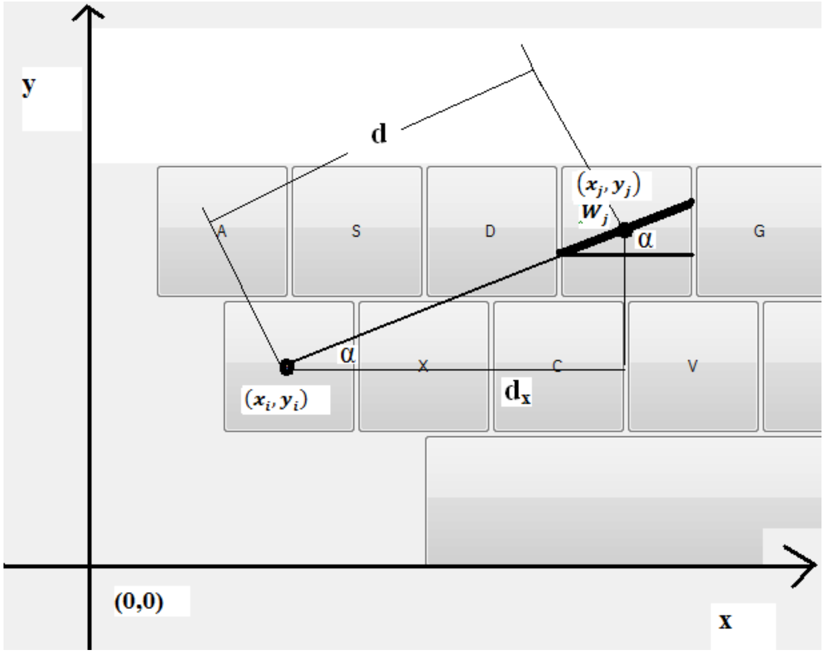


Figure 12.2. Calculation of the exact value of *Wj*

Then, the cosine of the angle *α* formed in target key *j* can be written as:

$$\cos\alpha = \frac{l}{w_j} \tag{12.3}$$

Hence, the exact value of target width in Fitts' Law (W_j) is:

$$w_j = \frac{l}{\cos\alpha} \tag{12.4}$$

Cos α can also be written from the triangular formed between keys *i* and key *j* as:

$$Cos\ \alpha = \frac{d_x}{d} \tag{12.5}$$

where

$$d_x = |x_j - x_i| \quad \text{and} \quad \boldsymbol{d} = \sqrt{(\boldsymbol{x_i} - \boldsymbol{x_j})^{\boldsymbol{2}} + (\boldsymbol{y_i} - \boldsymbol{y_j})^{\boldsymbol{2}}} \tag{12.6}$$

So, *Cos α* can be re-formulated by substitution as follows:

$$Cos\ \alpha = \frac{|x_j - x_i|}{\sqrt{(\boldsymbol{x_i} - \boldsymbol{x_j})^{\boldsymbol{2}} + (\boldsymbol{y_i} - \boldsymbol{y_j})^{\boldsymbol{2}}}} \tag{12.7}$$

Finally, by substitution, W_j can be expressed with the following formula:

$$w_j = \frac{l\ .\sqrt{(\boldsymbol{x_i} - \boldsymbol{x_j})^{\boldsymbol{2}} + (\boldsymbol{y_i} - \boldsymbol{y_j})^{\boldsymbol{2}}}}{|x_j - x_i|} \tag{12.8}$$

By the same manner, in the cases when $\alpha > 45^0$, the divisor will have to be replaced by $|y_j - y_i|$. It should be noted that when $\alpha > 45^0$,

$$|y_j - y_i| > |x_j - x_i| \tag{12.9}$$

Hence, the final formula should be written as a combination of two partial functions as follows:

$$w_j = \begin{cases} \dfrac{l.\sqrt{(\boldsymbol{x_i} - \boldsymbol{x_j})^{\boldsymbol{2}} + (\boldsymbol{y_i} - \boldsymbol{y_j})^{\boldsymbol{2}}}}{|x_j - x_i|}, & if \quad |x_j - x_i| \geq |y_j - y_i| \\[2ex] \dfrac{l\ .\sqrt{(\boldsymbol{x_i} - \boldsymbol{x_j})^{\boldsymbol{2}} + (\boldsymbol{y_i} - \boldsymbol{y_j})^{\boldsymbol{2}}}}{|y_j - y_i|}, & if \quad |x_j - x_i| < |y_j - y_i| \end{cases} \tag{12.10}$$

12.3. Generating the Fitts' Model

The parameters a and b are the intercept and slope of the regression line which fits the empirical data best, respectively. So, tapping time between randomly selected keys are measured and plotted. These time measurements are done in a random order in both longitudinal and square layouts.

The keys on test screen keyboards are tapped by clicking the mouse. The widths of the square shaped keys are 30 pixels and the intervals between keys are two pixels on a 1,280 by 800 pixels resolution screen.

Randomly selected 11 combinations of any two keys from square layout and 11 combinations from longitudinal keyboard are tried. Sample size ranged in 31 and 76. Samples having the greatest time measurements were omitted and removed from statistical analysis, because the tester may have slowed down abnormally due to some noisy effects. The average time measurements of remaining samples, shown in Table D.3 in Appendix D, are analysed.

By using the formula developed in Section 12.2., width of target keys in movement direction tables are generated for both the longitudinal keyboard as in Table D.1 and for square keyboard as in Table D.2 in Appendix D.

Distances between the starting keys and target keys, D_{ij} can be followed from the distance matrices in Table C.1 in Appendix C for longitudinal keyboard and in Table B.2 in Appendix B for square keyboard.

After applying the Fitts' model to the empirical data set, as shown in Table D.4 in Appendix D, the empirical data are plotted on a scatter chart to find the best fitting regression line, as shown in Figure 12.3.

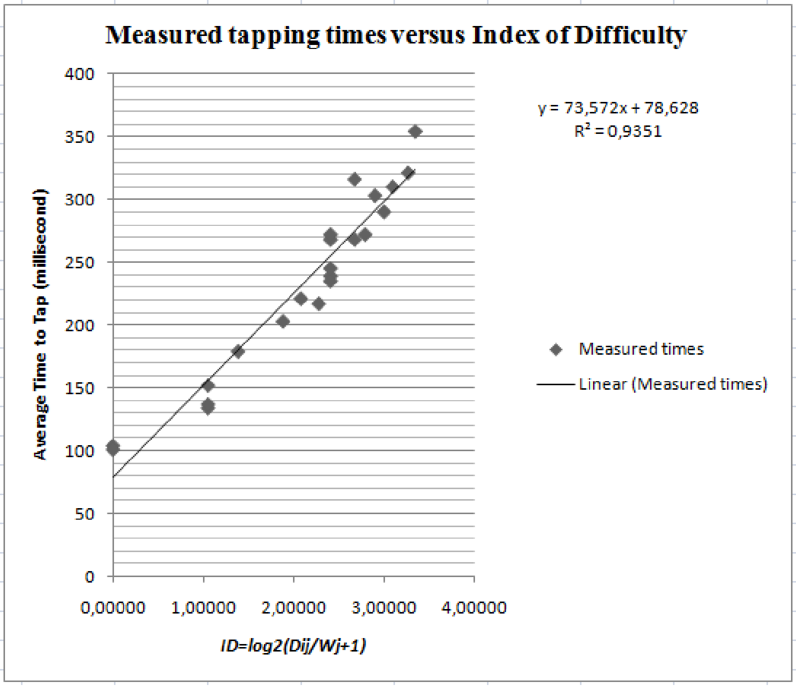


Figure 12.3. Empirical data plotted on scatter chart

R^2 for the regression is over 93.51 per cent, which is an acceptable ratio. Also, the intercept is 78.627 and the slope of the regression is 73.572. Hence, the Fitts' model can be formulated as:

$$t_{ij} = 78.627 + 73.572 \; log_2 \left(\frac{D_{ij}}{W_j} + 1 \right) \tag{12.11}$$

To relate this model with the digraph frequencies;

$$\bar{t} = \sum_{i,j \in S} p_{ij} \,.\, t_{ij} \tag{12.12}$$

$$\bar{t} = \sum_{i,j \in S} [78.627 + 73.572\, p_{ij} . log_2 \left(\frac{D_{ij} + W_j}{W_j} \right)] \tag{12.13}$$

where S is the key set and the unit of t is millisecond.

12.4. Estimating the Typing Speeds of Expert Users

The necessary tables to apply the Fitts' model generated in previous section are as follows:

D_{ij} Values for QWERTY, F and new longitudinal layouts: Table C.1 in Appendix C.

D_{ij} Values for new square layout: Table B.2 in Appendix B.

W_j Values for QWERTY, F and new longitudinal layout: Table D.1 in Appendix D.

W_j Values for new Square layout: Table D.2 in Appendix D.

By using the appropriate tables with the generated Fitts' model typing speeds of expert users are estimated. It should be implied that, in literature *words* are accepted as having five characters, including blank character.

As shown in Table 12.1., the layout having the maximum typing speed is the new generated optimum square keyboard layout with the value of 28.60 wpm. New generated longitudinal keyboard has also greater typing speed estimation than the similar shape layouts, *QWERTY and F layout,* with the typing speed of 27.54 wpm.

Table 12.1. Typing speed estimation for all layouts by Fitts' model

		Typing speed estimation	
Layout	$\sum_{i,j \in S} p_{ij} \cdot t_{ij}$	Character per minute *(cpm)*	Word per minute *(wpm)*
QWERTY	488,478.22	122.83	24.57
F	492,639.58	121.79	24.36
New Longitudinal	435,740.16	137.70	27.54
New Square	419,575.79	143.00	28.60

12.5. Correlation between Average Distance per Digraph and Typing Speed

Despite the great differences in average distance taken by the stylus per digraph values, typing speeds are closer to each other. For instance, average distance per digraph value for new square layout is less than that of QWERTY layout by 50.1 per cent; however typing speed of expert users for new square is only 16.40 per cent higher than QWERTY layout.

One of the interesting points for these typing estimations by Fitts' Law is that, in spite of the superiority of F layout over QWERTY in the comparison of average distance per digraph, QWERTY has slightly greater typing speeds in stylus typing. However, this difference is not significant and can be neglected.

Consequently, it can be now argued that, typing speeds are not fully correlated with average distance per digraph.

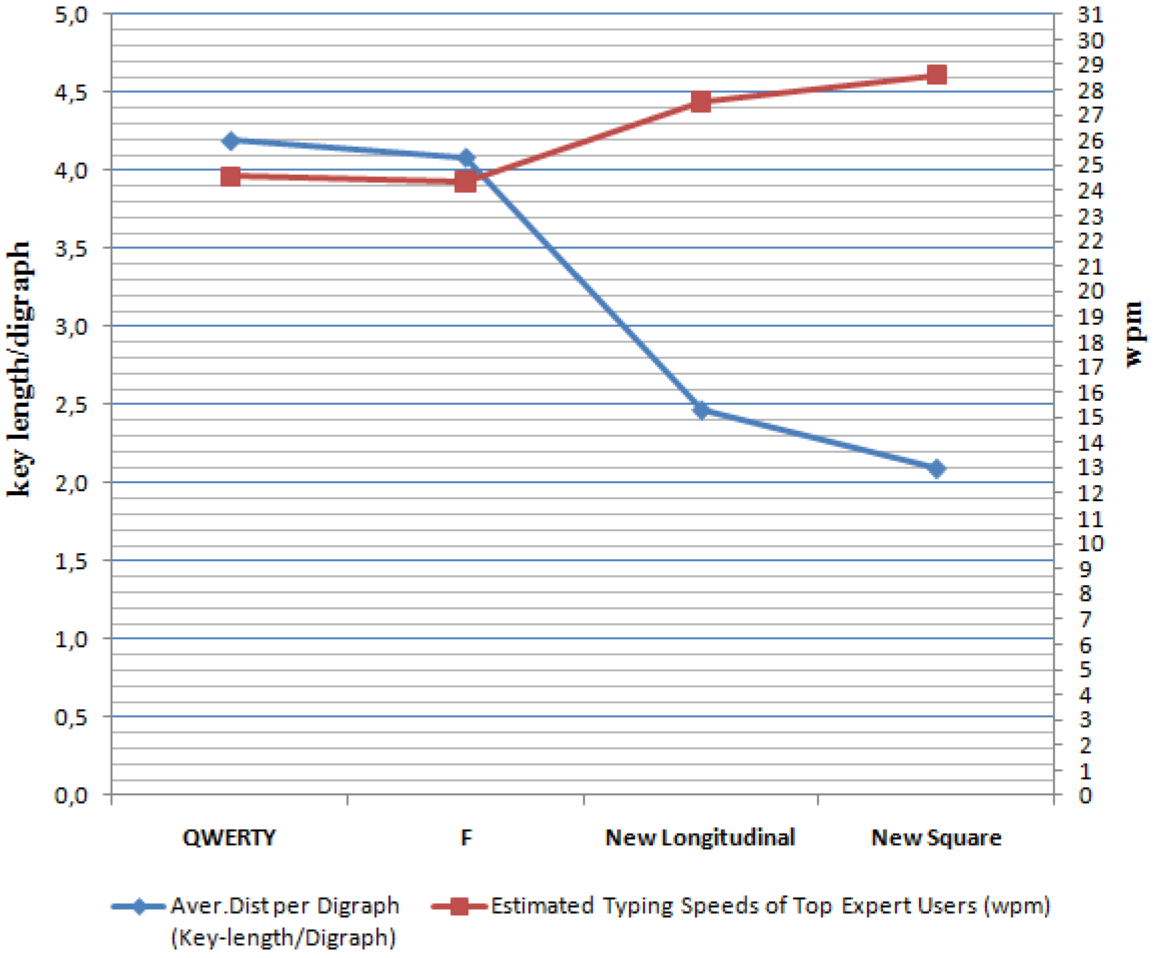


Figure 12.4. Relation between typing speed and average distance per digraph

As depicted in Table 12.2., the correlation between the average distances taken by the stylus with the estimated typing speed of expert users is −0.9948. With no doubt, there is a negative correlation.

Table 12.2. Negative correlation between average distance and typing speed

Layout	**Average Distance per Digraph (in Key-length)**	**Estimated Typing Speeds of Expert Users(wpm)**
QWERTY	4.1906	24.57
F	4.0848	24.36
New Longitudinal	2.4651	27.54
New Square	2.0906	28.60
Correlation	- 0.9948	

12.6. Rough Estimation of Typing Speeds on Touch Screen

Yanzhi *et.al* (2006) estimated typing speeds on a touch screen stylus keyboard, shown on Table 4.1. They estimated this value for QWERTY as 31.15 wpm. So, it can be roughly calculated that *touch typing* is $31.15/24.57 = 1.27$ times speedy than *click typing*. Hence, with a rough direct proportion, new generated square layout will have $1.27x28.60 = 36.32\,wpm$ on touch typing screen. Similarly, new longitudinal layout will have $1.27x27.54 = 34.98\,wpm$ on touch typing screen. Hence, rough estimations of typing speeds of these four layouts on touch screen as in Table 12.3.

Table 12.3. Rough estimation of typing speeds on a touch screen

	Typing speeds	
Layout	Character per minute	Word per minute
QWERTY	155.99	31.20
F	154.68	30.94
New Longitudinal	174.87	34.98
New Square	181.61	36.32

13. COMPARING QWERTY AND NEW SQUARE LAYOUT ON SUBJECTS

In this chapter; usability tests to evaluate the performance of new generated square layout compared with the most common layout, *QWERTY layout*, were undertaken. To achieve this, learning curves versus sessions' graphs of these two layouts were tried to be plotted using five subjects.

13.1. Motivation

Main purpose of this usability test is to see the trend in learning curves of the QWERTY layout and New Square layout by plotting the typing speed versus session graph. With no doubt, it is expected that the typing performance of the subjects will be high initially in QWERTY layout, due to the long time familiarization with it. However, at the end of each session, it is also expected that the gap between performance of the New Square and QWERTY layout will get closer and closer. After a sufficient number of sessions, owing to the results of Fitts' Law, the typing speeds of the subjects will get higher than QWERTY layout. Stylus keyboard evaluation strategies of Joon Lee with Zhai and MacKenzie with Zhang are tried to be followed (MacKenzie and Zhang, 1999; Joon Lee and Zhai, 2004).

13.2. Subjects

Five subjects participated in the experiments. All were university graduate. Dominant hands for all were right hand. All were familiar with both desktop and laptop computers. Also, they were unfamiliar to stylus keyboard typing. None of the subjects had an apparent physical or anatomic problem that may affect their typing performance, except for Subject 1 and Subject 4 wearing glasses. All are familiar with QWERTY layout, but not expert typing on it.

Their native languages were Turkish. They were all voluntary participator to the experiments and nothing paid to them for participating. All were male, at the age of 26 and 34.

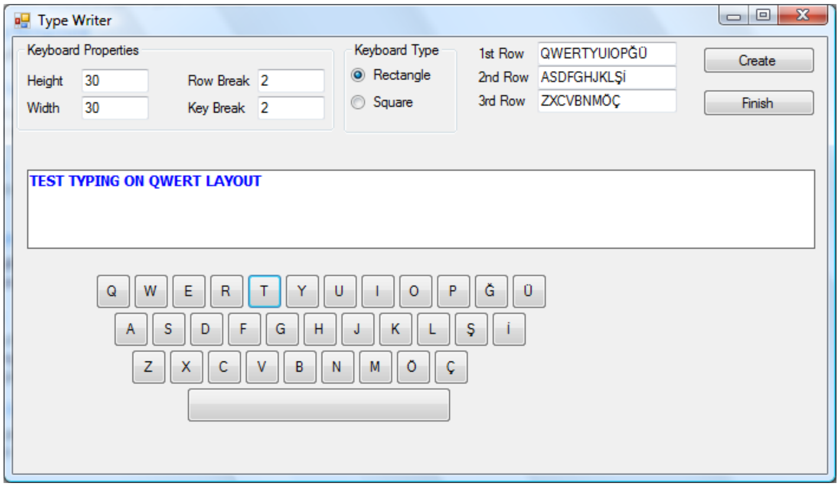
13.3. Apparatus

The experiment software was coded in Visual C# (*C Sharp*). Microsoft Office 2007 Excel was used for the database and data analysing module. The software was run on the same computer, which was Toshiba Satellite A300-20P, during all stages of the experiments. Display options were also kept unchanged, which were 60 Hertz for screen refresh rate; 1280 by 800 pixels for resolution and 32 bit (highest) for colours setting.

The user interface of the experiment software has the following features:

- *Keyboard display options*: *Height* and *width* of the keys can be altered (*in pixels*). During experiments, both the height and width of the keys were set 30 pixels. Also, the *Row Break* and the *Key Break* were set equal to 2 pixels, in order to keep the ratio same as in Square Model of this thesis. (As in Section 6.2.)
- *Keyboard Type*: layout of the keys is arranged according to option. During experiments; this option was set to *Rectangle* for QWERTY layout tests whereas it was set to *Square* for new generated square layout.
- *Row arrangement lines*: keyboard layouts are generated by using these rows. Three rows are available when Rectangle keyboard option is selected, since QWERTY layout has three rows for letters and a fixed space bar for blank character. Five rows become available when square keyboard option is selected.
- *Start/Finish button*: Subject should press *Start* button for data recording of the tests. Time measurements start when the subject presses the first letter of the text; *not starts by just pressing Start button*.

- *Text area*: the subjects can see what they have typed during the experiments by looking at this area.



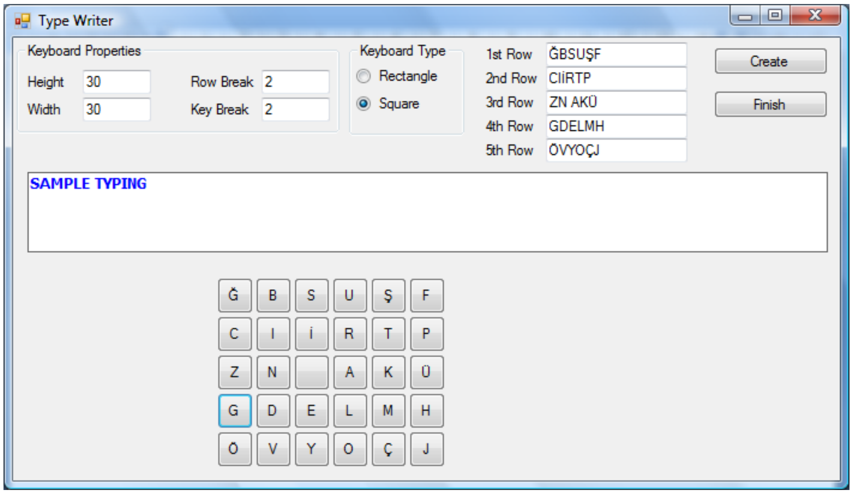
Keyboard Properties
Height 30
Width 30
Row Break 2
Key Break 2

Keyboard Type
● Rectangle
○ Square

1st Row QWERTYUIOPĞÜ
2nd Row ASDFGHJKLŞİ
3rd Row ZXCVBNMÖÇ

TEST TYPING ON QWERT LAYOUT

Figure 13.1. User interface of the experiment software for long. keyboard



Keyboard Properties
Height 30
Width 30
Row Break 2
Key Break 2

Keyboard Type
○ Rectangle
● Square

1st Row ĞBSUŞF
2nd Row CIİRTP
3rd Row ZN AKÜ
4th Row GDELMH
5th Row ÖVYOÇJ

SAMPLE TYPING

Figure 13.2. User interface of the experiment software for square keyboard

13.4. Text Material for Testing

In order to eliminate the differences in letter distribution and effect of altering average distance per digraph value, same text material has been used during all sessions by the subjects. This text material should be chosen in such a way that it must be neither advantageous nor disadvantageous for each of the layouts. In other words, this text material should have similar average distance per digraph distribution with the whole Turkish language for both of the layouts. Also, this text should not be too long to limit the effect of tiredness of the subjects. Finally, this text should not involve punctuation mark, since the new generated layouts are optimized for only the 29 letters of Turkish language and blank character.

After a long research, this text material has been chosen as: "*Reklamveren bu tür reklamı yayınlatan bir reklam şirketi ile çalışmayı tercih edecekse sözüm yoktur*". This text material has 98 digraphs, hence 99 characters, 13 of whose are blank.

Table 13.1. Correlation of the test text material with Turkish language

Layout	Average distance per digraph of the text material (*in key-length*)	Average distance per digraph of Turkish language (*in key-length*)	Relative gap (*per cent*)
QWERTY	4.1848	4.1906	- 0.14
New Square	2.0964	2.0906	0.28
Correlation	0.9999 >		

13.5. Experiment Design

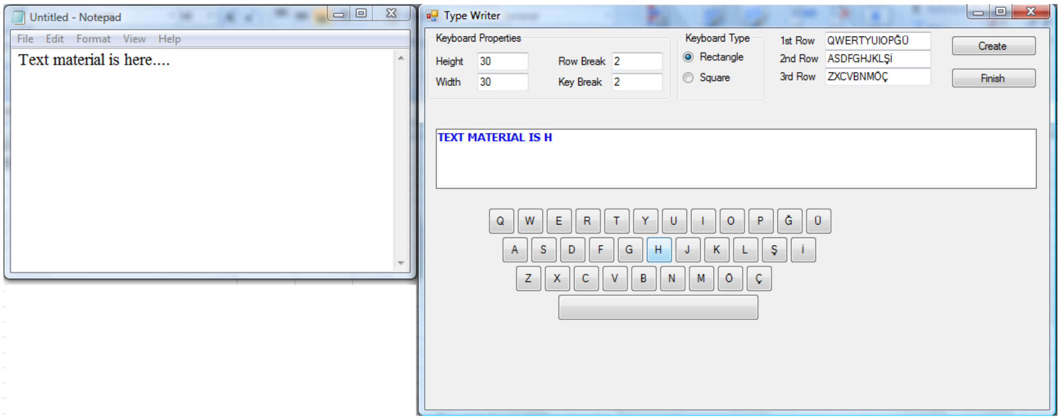
The experiment has a 2 x 30 within-each subject factorial design. The designed factors are;

(i): Keyboard layouts {QWERTY, New Square}
(ii): Session {30 sessions}

As a result, each subject tried 30 sessions for both of the layouts. To decrease the effect of “memorizing the text”, the order of the tries was designed in a random order and each subject followed this design order. By this randomly designed order, the interaction between the experiment parts is tried to be minimized.

13.6. Procedure

Before the experiments, a 10 minute warming test is undertaken without measuring time. The layouts are depicted to the subjects on the test software. It is told to the subjects that “both accuracy and typing speed is important for this experiment”. Also, it is clearly implied to them that the time measurements will not be shown to the other subject, in order to prevent a competition between the subjects. Each participant was given oral instructions explaining the task and the goal of the experiment. They were asked specifically to aim for both entry speed and accuracy. The instructions also stated that if they made an error, that try will be repeated. In other words, a session is repeated if the text typed in that session has missing, extra or different characters when compared with the test material. Hence, error rate after each session should be zero. Fifteen sessions per subject were undertaken at most a day. An error in this study is recorded and counted when the subject typed extra, missing or incorrect characters than the text material. The subjects are told to repeat that session if he made an error.



Text material is here....

Height 30
Width 30
Row Break 2
Key Break 2
● Rectangle
○ Square
1st Row QWERTYUIOPĞÜ
2nd Row ASDFGHJKLŞİ
3rd Row ZXCVBNMÖÇ

TEXT MATERIAL IS H

Figure 13.3. Position of text material during experiments

13.7. Results and Discussion

Since the subjects were initially familiar with the standard QWERTY layout for a long time, their initial typing speeds were generally higher than the New Square layout. This result can also be derived from the typing speed values in first few sessions.

The slope of the learning curve for New Square layout can be said to be greater than that of QWERTY layout. This is the natural result of the initial and final values of the typing speeds of subjects. Their speeds were higher in QWERTY layout until the crossover session and after this session, the speeds on New Square layout surpassed the speeds on QWERTY layout.

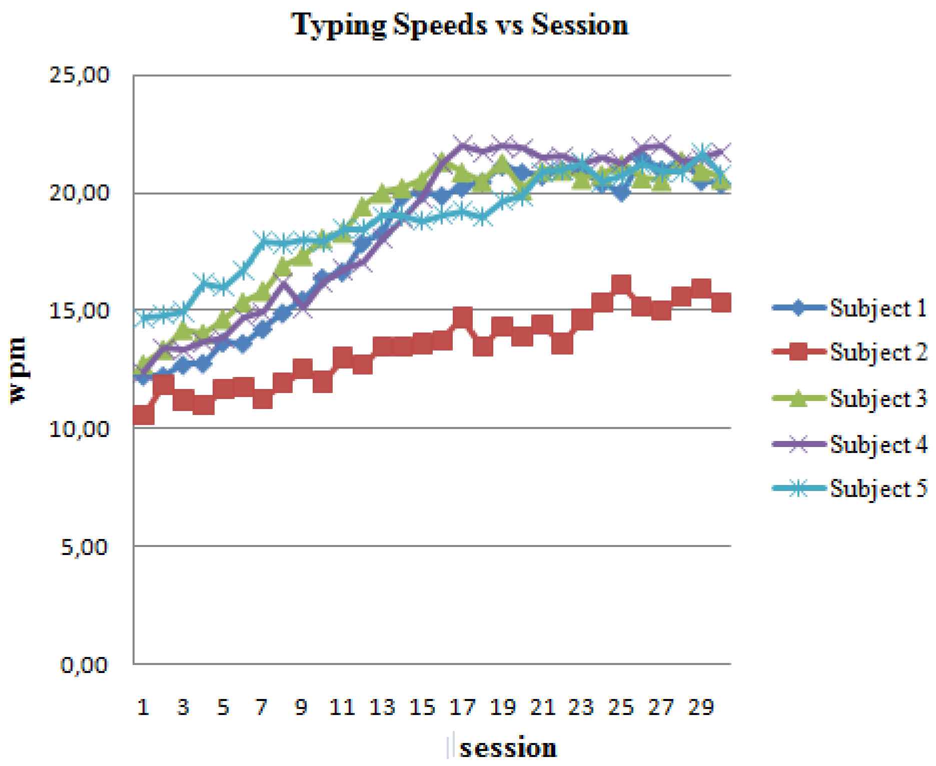


Figure 13.4. Typing speed of subjects on QWERTY layout by session

The crossover session changes from subject to subject, depending on his past experience on QWERTY and also on his ability. When the average of five subjects were analysed, speeds on QWERTY are usually higher until session #12. Between the session 13 and session 18, the orders of speeds alter between both layouts. However, after the session 18, the New Square layout has higher typing speeds upon QWERTY layout.

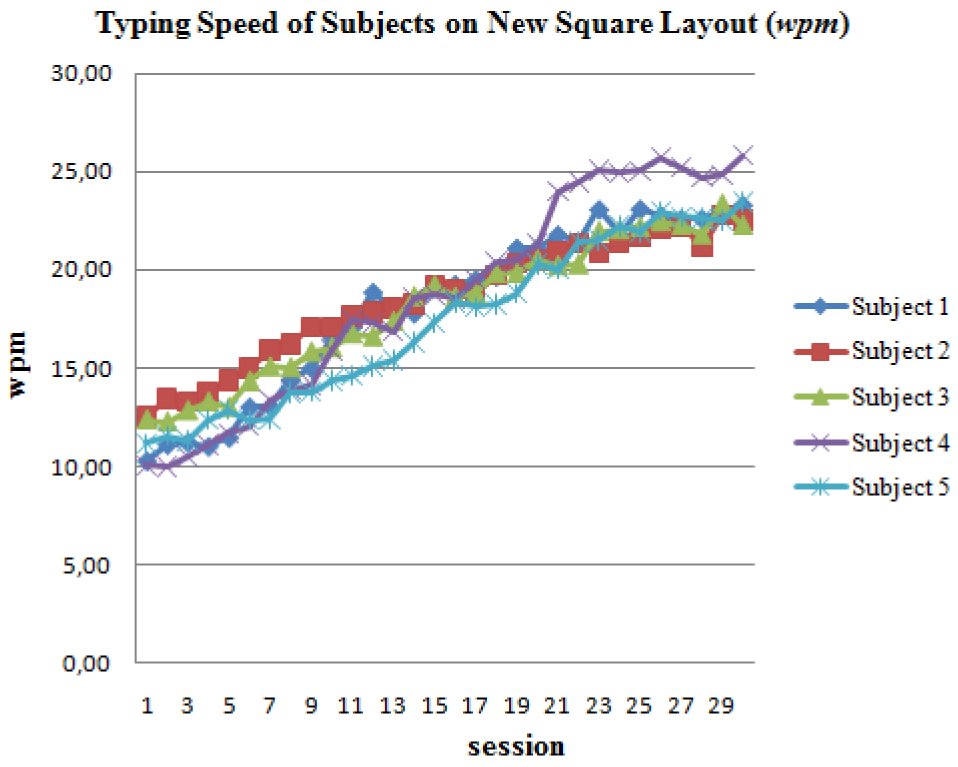


Figure 13.5. Typing speed of subjects on New Square layout by session

Except for the alternating speeds due to noise effects, crossover session for these layouts can be said to be the session 18, and new generated square layout has over 18 per cent higher typing speeds than QWERTY layout after 30 sessions.

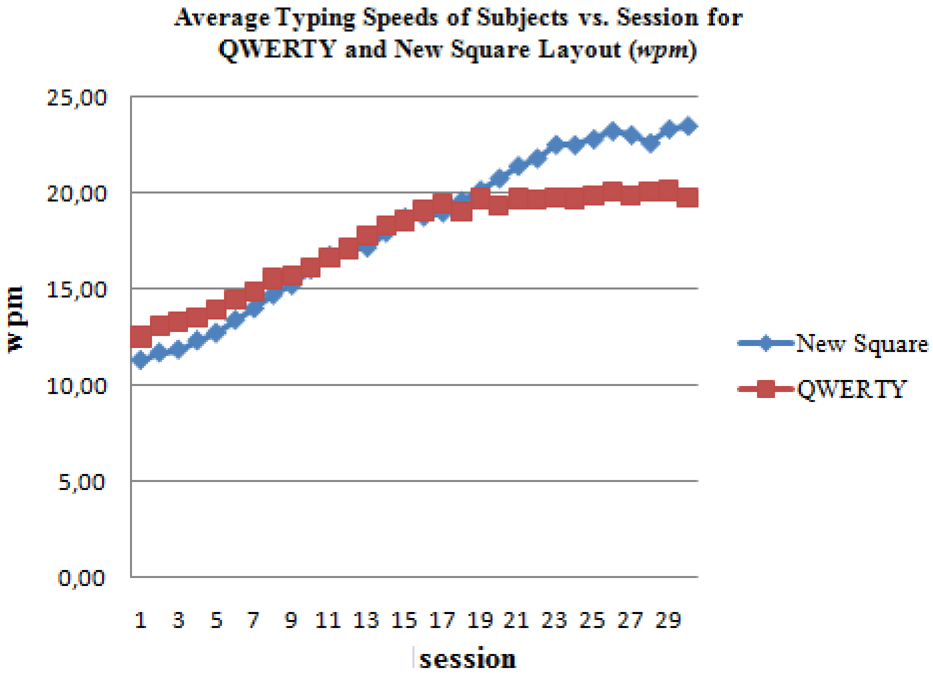


Figure 13.6. Crossover of learning curves along sessions

14. CONCLUSIONS

During this study, two stylus keyboard layout are designed for Turkish language, first of which is classical longitudinal shaped as QWERTY layout involving three rows and the other is square shaped involving six columns and five rows.

1. Digraph frequency table of Turkish language is generated via using computer programs to analyse a large number of texts from different areas, such as daily newspapers, magazines, academic papers and Turkish classics. Totally over 4,000 pages involving over two millions of letters are scanned to generate the table, which depicts the probability of a letter to come before and next to another letter.

2. To design the new layouts, stylus keyboard typing task is modelled as a quadratic assignment problem. Due to its NP-hardness, there is no algorithm or technique that guarantees the optimality of a proposed solution. Hence, a number of metaheuristics for global type problems and problem focused heuristics are developed to search a near optimum solution.

3. Fortunately and generally, there is a common property of languages that help to find a near optimum solution to stylus keyboard layout optimization: rapid convergence of cumulative density function of the frequencies of digraphs, or to express in a different way, the existence of the dominance of a small portion of letters in languages. By employing this property; a deterministic, greedy, constructive and decomposition based heuristic, called Optimum Connected Optimum Blocks (OCOB), is developed for stylus keyboard layout optimization.

4. OCOB has one decision point: the number of maximum letters, say *n*, to be relaxed in a run. The larger *n* means, closer solution to global optimum; however this results in

extremely huge amount of solution time. When *n* versus solution time graphs are plotted, it is clearly noted that after a value of *n*, the improvement in solution has no sensible change. Solution gets better even one or two per thousand when *n* increases by one, which can be disregarded for stylus keyboard layout optimization.

5. Main evaluation and optimization criterion has been decided as the average distance taken by the stylus per digraph. This is the average distance that the tip of the stylus takes between two consecutive letters during typing a Turkish, having the unit *key-length/digraph*.

By the multiplication of digraph frequency matrix of Turkish language and related distance matrices of keyboards, the expected values for average distances per digraph values are calculated as 4.1906, 4.0848, 2.4651 and 2.0906 for QWERTY, F, New Longitudinal and New Square layouts, respectively. Hence, New Longitudinal is 41.18 and New Square is 50.11 more efficient than QWERTY. Also F layout, which is designed specifically for Turkish physical keyboards, is only 2.53 more efficient than QWERTY.

6. Computer aided tests are applied by simulating a single finger robot, which types a text via using its tip of arm according to installed keyboard layouts. In this manner, randomly selected chunk of electronic texts are analysed. Simulation results are compatible with the expected values; with the ratios for "*Standard Deviation/Mean*" as 1.67 per cent, 1.52 per cent, 1.94 per cent and 1.16 per cent for QWERTY, F, New Longitudinal and New Square layouts, respectively. As the size of the text increases, the simulation results get closer to the expected values.

7. Another performance comparison is done by a Human-Computer Interaction model, Fitts' Law. This law is modified to estimate the typing speed of expert users in a keyboard layout. These speeds are predicted as 24.57, 24.36, 27.54 and 28.60 words per minute (*wpm*). It is observed that the efficiencies get smaller in the view of typing speed. For instance, New Square layout is more effective in typing speed of expert user over QWERTY layout by only 16.40 per cent according to Fitts' model application; however the efficiency was 50.11 per cent in the view of average distance per digraph, as previously mentioned. The reason for this

interesting point might be that, typing speed is not affected only by the average distance taken by the stylus, but also affected by the response time of the user with acceleration and deceleration speed ratio during typing task.

8. Additionally, typing speeds of expert users on a touch screen stylus keyboards are roughly estimated with a direct proportion from the literature; as 31.20 wpm, 30.94 wpm, 34.98 wpm and 36.32 wpm for QWERTY, F, New Longitudinal and New Square layouts, respectively.

9. Finally, the typing speeds of QWERTY and New Square layouts were compared with five subjects following a laboratory procedure using a specially designed test package program. These tests lasted 30 sessions and the subjects are instructed to write a short text with zero error rate. Due to the past familiarization of the subjects on QWERTY layout, initial speeds were higher on QWERTY layout. After session 12, the speeds started to alter with each other, and finally after session 18, New Square layout surpassed QWERTY layout completely. After session 30, average speeds of five subjects were 23.47 wpm for New Square and 19.77 wpm for QWERTY layout. Hence, after a training program of 30 sessions, it can be defended that users will reach up 18 per cent higher typing speeds on New Square layout than QWERTY layout, which is compatible with the results of our Fitts’ model.

15. CONTRIBUTIONS TO THE FIELD AND FUTURE WORKS

15.1. Contributions to the Field

The contributions that have been made to either optimization or ergonomics field throughout this thesis could be listed as follows;

i. A detailed digraph frequency table for Turkish language has been prepared. This table shows the probability of the permutation of any two letters in an average Turkish text.

ii. A decomposition-based heuristic for the quadratic assignment model of stylus keyboard layout optimization for any language is developed.

iii. In literature, researchers usually approximate the width of the target key as the length of an edge of a key while applying Fitts' Law in stylus keyboards. So, a partial function for the exact calculation of the width of the target key in Fitts' Law for two dimensional movements is developed in this study.

iv. Optimum longitudinal and square shaped stylus keyboard layouts are designed for Turkish language for the first time in literature. These optimized layouts are also compared with the currently used layouts using different techniques.

15.2. Future Works

Following future works can be advised for those researchers interested in this field;

i. Different shaped keys, such as hexagonal or circular shaped, can be used in keyboard. To apply this, distance matrix should be revised with the same flow matrix.

ii. The efficiency of the new developed heuristic can be compared with the currently used metaheuristics by using the frequency matrices of other languages.

iii. Some metaheuristics can be applied to the flow and distance matrices of this thesis, in order to test the optimality of the already designed layouts.

iv. The *Wj* values of keys can be inserted into the objective function of the quadratic assignment model in order to design a better layout which maximizes the typing speed of expert users calculated by Fitts' Law.

APPENDIX A: DERIVATION OF FITTS' LAW

Assume that the user moves toward the target in a sequence of sub movements. Each submovement requires a constant time t to execute, and moves a constant fraction $1 - r$ of the remaining distance to the centre of the target, where $0 < r < 1$. Thus, if the user is initially at a distance D from the target, the remaining distance after the first submovement is rD, and the remaining distance after the nth submovement is rnD. (In other words, the distance left to the target's centre is a function that decays exponentially over time.) Let N be the (possibly fractional) number of sub movements required to fall within the target and W be the width of target (Fitts' Law, Wikipedia). Then,

$$r^N D = \frac{W}{2} \tag{A.1.}$$

Solving for N:

$$N = log_r \left(\frac{W}{2D}\right) \tag{A.2.}$$

$$= \frac{1}{log_2 r} log_2 \left(\frac{W}{2D}\right) \quad (Since\ log_x y\ = \frac{(log_z y)}{(log_z)x}) \tag{A.3.}$$

$$= \frac{1}{log_2 \left(\frac{1}{r}\right)} log_2 \left(\frac{2D}{W}\right) \quad (Since\ log_x y = -log_x \left(\frac{1}{y}\right)) \tag{A.4.}$$

The time required for all sub movements is:

$$T = Nt = \frac{t}{log_2\left(\frac{1}{r}\right)} log_2(\frac{2D}{W}) \tag{A.5.}$$

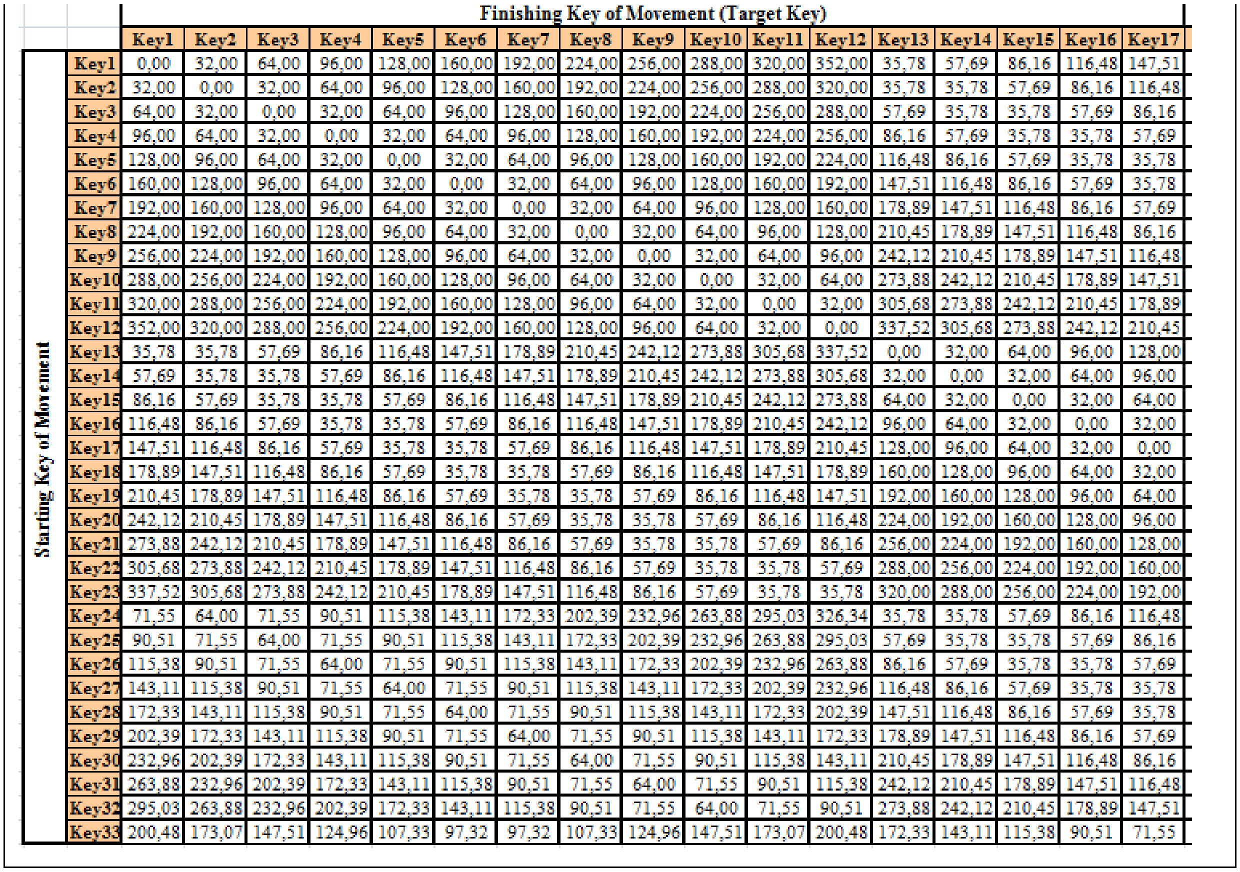
$$= \frac{t}{log_2(1/r)} + \frac{t}{log_2(1/r)} log_2\frac{D}{W} \tag{A.6.}$$

By defining appropriate constants a and b, above formula can be rewritten as:

$$T = a + blog_2\frac{D}{W} \tag{A.7.}$$

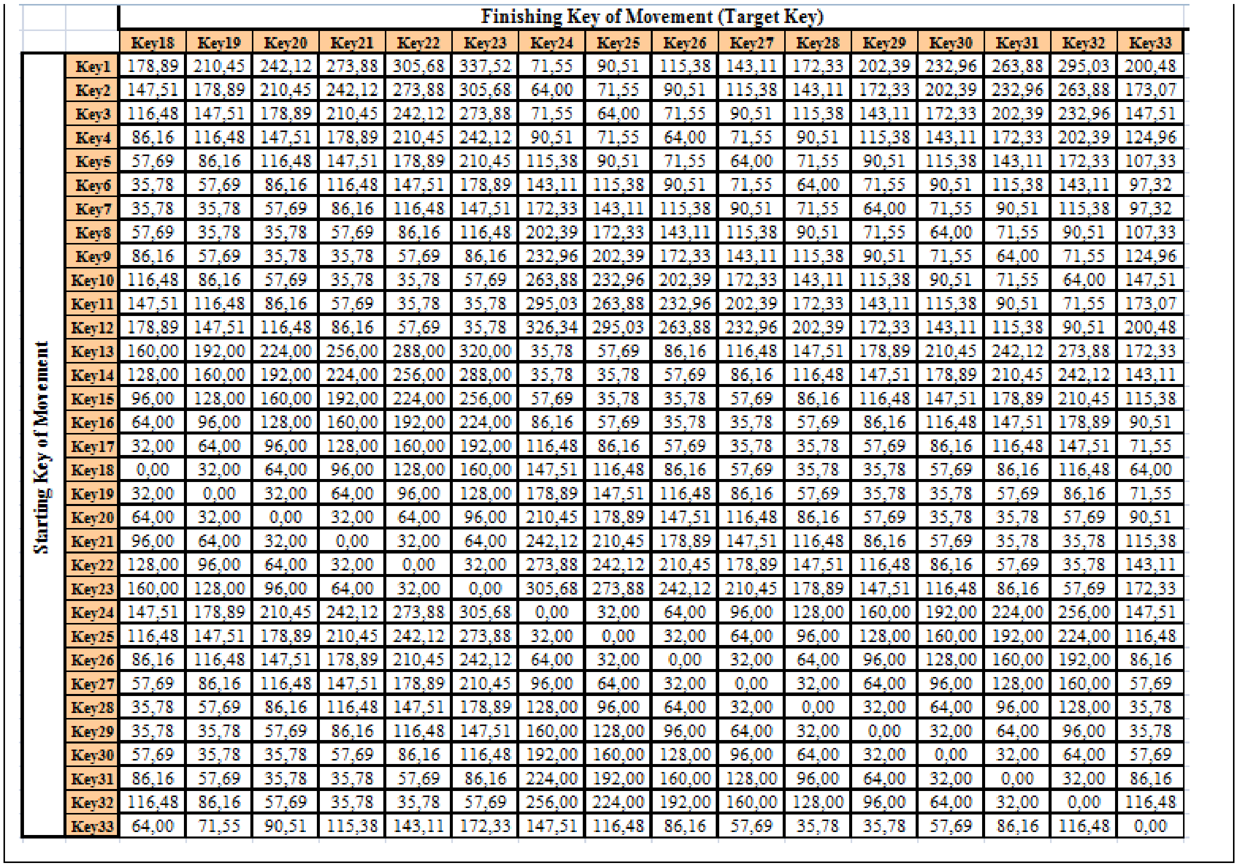
APPENDIX B: PARAMETERS OF STYLUS KEYBOARD LAYOUT MODELS

Table B.1. Distance matrix for longitudinal keyboard layout model



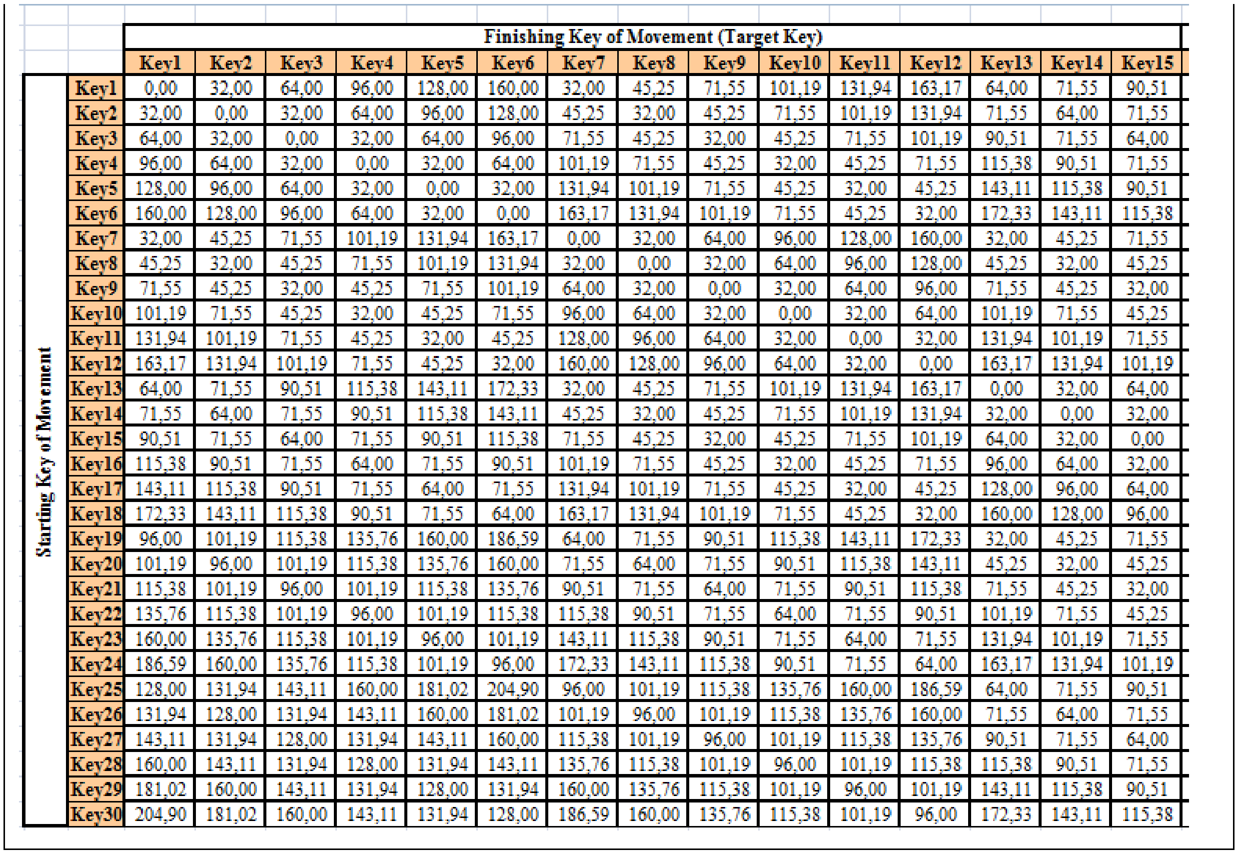
	Finishing Key of Movement (Target Key)																
Starting Key of Movement	Key1	Key2	Key3	Key4	Key5	Key6	Key7	Key8	Key9	Key10	Key11	Key12	Key13	Key14	Key15	Key16	Key17
Key1	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	256,00	288,00	320,00	352,00	35,78	57,69	86,16	116,48	147,51
Key2	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	256,00	288,00	320,00	35,78	35,78	57,69	86,16	116,48
Key3	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	256,00	288,00	57,69	35,78	35,78	57,69	86,16
Key4	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	256,00	86,16	57,69	35,78	35,78	57,69
Key5	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	116,48	86,16	57,69	35,78	35,78
Key6	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	147,51	116,48	86,16	57,69	35,78
Key7	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	178,89	147,51	116,48	86,16	57,69
Key8	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	210,45	178,89	147,51	116,48	86,16
Key9	256,00	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	242,12	210,45	178,89	147,51	116,48
Key10	288,00	256,00	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	273,88	242,12	210,45	178,89	147,51
Key11	320,00	288,00	256,00	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	305,68	273,88	242,12	210,45	178,89
Key12	352,00	320,00	288,00	256,00	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	337,52	305,68	273,88	242,12	210,45
Key13	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	242,12	273,88	305,68	337,52	0,00	32,00	64,00	96,00	128,00
Key14	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	242,12	273,88	305,68	32,00	0,00	32,00	64,00	96,00
Key15	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	242,12	273,88	64,00	32,00	0,00	32,00	64,00
Key16	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	242,12	96,00	64,00	32,00	0,00	32,00
Key17	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	128,00	96,00	64,00	32,00	0,00
Key18	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	160,00	128,00	96,00	64,00	32,00
Key19	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	192,00	160,00	128,00	96,00	64,00
Key20	242,12	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	224,00	192,00	160,00	128,00	96,00
Key21	273,88	242,12	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	256,00	224,00	192,00	160,00	128,00
Key22	305,68	273,88	242,12	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	288,00	256,00	224,00	192,00	160,00
Key23	337,52	305,68	273,88	242,12	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	320,00	288,00	256,00	224,00	192,00
Key24	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	232,96	263,88	295,03	326,34	35,78	35,78	57,69	86,16	116,48
Key25	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	232,96	263,88	295,03	57,69	35,78	35,78	57,69	86,16
Key26	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	232,96	263,88	86,16	57,69	35,78	35,78	57,69
Key27	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	232,96	116,48	86,16	57,69	35,78	35,78
Key28	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	147,51	116,48	86,16	57,69	35,78
Key29	202,39	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	178,89	147,51	116,48	86,16	57,69
Key30	232,96	202,39	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	210,45	178,89	147,51	116,48	86,16
Key31	263,88	232,96	202,39	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	242,12	210,45	178,89	147,51	116,48
Key32	295,03	263,88	232,96	202,39	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	273,88	242,12	210,45	178,89	147,51
Key33	200,48	173,07	147,51	124,96	107,33	97,32	97,32	107,33	124,96	147,51	173,07	200,48	172,33	143,11	115,38	90,51	71,55

Table B.1. (*cont'd)* Distance matrix for longitudinal keyboard layout model



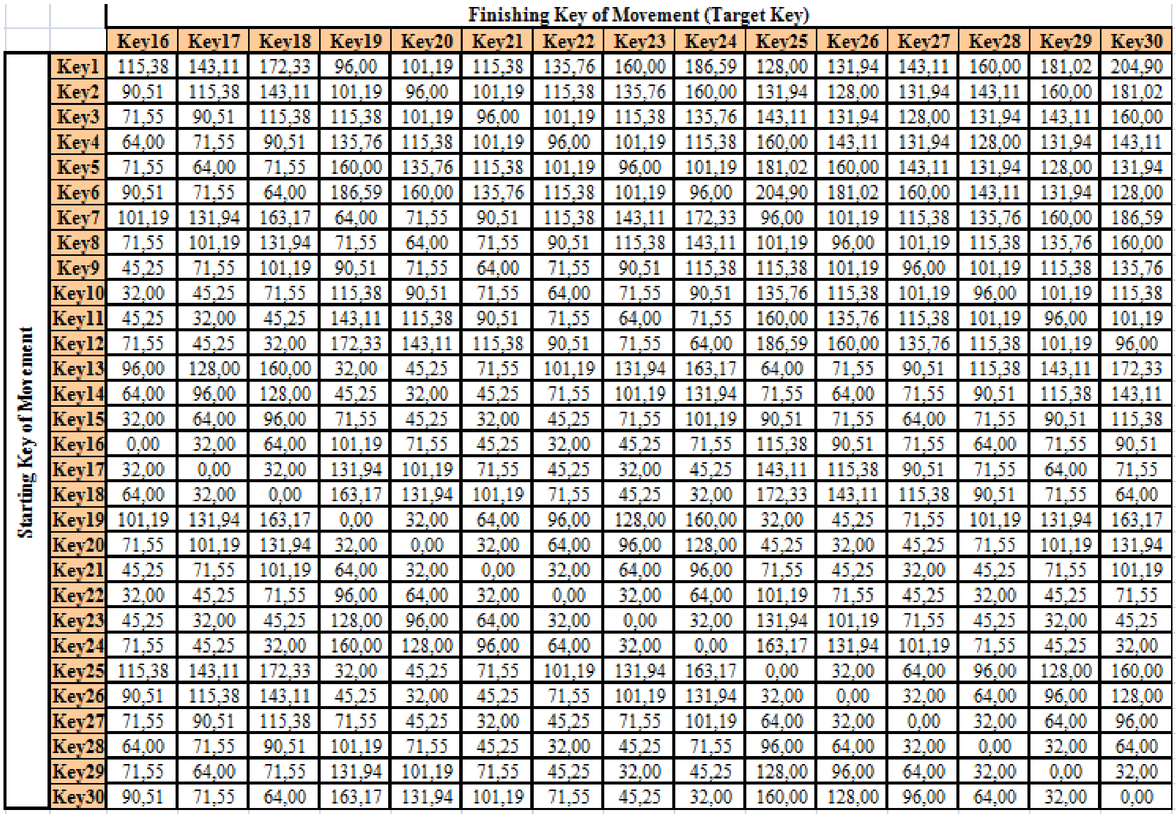
		Finishing Key of Movement (Target Key)															
		Key18	**Key19**	**Key20**	**Key21**	**Key22**	**Key23**	**Key24**	**Key25**	**Key26**	**Key27**	**Key28**	**Key29**	**Key30**	**Key31**	**Key32**	**Key33**
Starting Key of Movement	**Key1**	178,89	210,45	242,12	273,88	305,68	337,52	71,55	90,51	115,38	143,11	172,33	202,39	232,96	263,88	295,03	200,48
	Key2	147,51	178,89	210,45	242,12	273,88	305,68	64,00	71,55	90,51	115,38	143,11	172,33	202,39	232,96	263,88	173,07
	Key3	116,48	147,51	178,89	210,45	242,12	273,88	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	232,96	147,51
	Key4	86,16	116,48	147,51	178,89	210,45	242,12	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	124,96
	Key5	57,69	86,16	116,48	147,51	178,89	210,45	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	107,33
	Key6	35,78	57,69	86,16	116,48	147,51	178,89	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	97,32
	Key7	35,78	35,78	57,69	86,16	116,48	147,51	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	97,32
	Key8	57,69	35,78	35,78	57,69	86,16	116,48	202,39	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	107,33
	Key9	86,16	57,69	35,78	35,78	57,69	86,16	232,96	202,39	172,33	143,11	115,38	90,51	71,55	64,00	71,55	124,96
	Key10	116,48	86,16	57,69	35,78	35,78	57,69	263,88	232,96	202,39	172,33	143,11	115,38	90,51	71,55	64,00	147,51
	Key11	147,51	116,48	86,16	57,69	35,78	35,78	295,03	263,88	232,96	202,39	172,33	143,11	115,38	90,51	71,55	173,07
	Key12	178,89	147,51	116,48	86,16	57,69	35,78	326,34	295,03	263,88	232,96	202,39	172,33	143,11	115,38	90,51	200,48
	Key13	160,00	192,00	224,00	256,00	288,00	320,00	35,78	57,69	86,16	116,48	147,51	178,89	210,45	242,12	273,88	172,33
	Key14	128,00	160,00	192,00	224,00	256,00	288,00	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	242,12	143,11
	Key15	96,00	128,00	160,00	192,00	224,00	256,00	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	115,38
	Key16	64,00	96,00	128,00	160,00	192,00	224,00	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	90,51
	Key17	32,00	64,00	96,00	128,00	160,00	192,00	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	71,55
	Key18	0,00	32,00	64,00	96,00	128,00	160,00	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	64,00
	Key19	32,00	0,00	32,00	64,00	96,00	128,00	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	71,55
	Key20	64,00	32,00	0,00	32,00	64,00	96,00	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	90,51
	Key21	96,00	64,00	32,00	0,00	32,00	64,00	242,12	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	115,38
	Key22	128,00	96,00	64,00	32,00	0,00	32,00	273,88	242,12	210,45	178,89	147,51	116,48	86,16	57,69	35,78	143,11
	Key23	160,00	128,00	96,00	64,00	32,00	0,00	305,68	273,88	242,12	210,45	178,89	147,51	116,48	86,16	57,69	172,33
	Key24	147,51	178,89	210,45	242,12	273,88	305,68	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	256,00	147,51
	Key25	116,48	147,51	178,89	210,45	242,12	273,88	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	116,48
	Key26	86,16	116,48	147,51	178,89	210,45	242,12	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	86,16
	Key27	57,69	86,16	116,48	147,51	178,89	210,45	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	57,69
	Key28	35,78	57,69	86,16	116,48	147,51	178,89	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	35,78
	Key29	35,78	35,78	57,69	86,16	116,48	147,51	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	35,78
	Key30	57,69	35,78	35,78	57,69	86,16	116,48	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	57,69
	Key31	86,16	57,69	35,78	35,78	57,69	86,16	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	86,16
	Key32	116,48	86,16	57,69	35,78	35,78	57,69	256,00	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	116,48
	Key33	64,00	71,55	90,51	115,38	143,11	172,33	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	0,00

Table B.2. Distance matrix for square keyboard layout model



		Finishing Key of Movement (Target Key)														
		Key1	**Key2**	**Key3**	**Key4**	**Key5**	**Key6**	**Key7**	**Key8**	**Key9**	**Key10**	**Key11**	**Key12**	**Key13**	**Key14**	**Key15**
Starting Key of Movement	**Key1**	0,00	32,00	64,00	96,00	128,00	160,00	32,00	45,25	71,55	101,19	131,94	163,17	64,00	71,55	90,51
	Key2	32,00	0,00	32,00	64,00	96,00	128,00	45,25	32,00	45,25	71,55	101,19	131,94	71,55	64,00	71,55
	Key3	64,00	32,00	0,00	32,00	64,00	96,00	71,55	45,25	32,00	45,25	71,55	101,19	90,51	71,55	64,00
	Key4	96,00	64,00	32,00	0,00	32,00	64,00	101,19	71,55	45,25	32,00	45,25	71,55	115,38	90,51	71,55
	Key5	128,00	96,00	64,00	32,00	0,00	32,00	131,94	101,19	71,55	45,25	32,00	45,25	143,11	115,38	90,51
	Key6	160,00	128,00	96,00	64,00	32,00	0,00	163,17	131,94	101,19	71,55	45,25	32,00	172,33	143,11	115,38
	Key7	32,00	45,25	71,55	101,19	131,94	163,17	0,00	32,00	64,00	96,00	128,00	160,00	32,00	45,25	71,55
	Key8	45,25	32,00	45,25	71,55	101,19	131,94	32,00	0,00	32,00	64,00	96,00	128,00	45,25	32,00	45,25
	Key9	71,55	45,25	32,00	45,25	71,55	101,19	64,00	32,00	0,00	32,00	64,00	96,00	71,55	45,25	32,00
	Key10	101,19	71,55	45,25	32,00	45,25	71,55	96,00	64,00	32,00	0,00	32,00	64,00	101,19	71,55	45,25
	Key11	131,94	101,19	71,55	45,25	32,00	45,25	128,00	96,00	64,00	32,00	0,00	32,00	131,94	101,19	71,55
	Key12	163,17	131,94	101,19	71,55	45,25	32,00	160,00	128,00	96,00	64,00	32,00	0,00	163,17	131,94	101,19
	Key13	64,00	71,55	90,51	115,38	143,11	172,33	32,00	45,25	71,55	101,19	131,94	163,17	0,00	32,00	64,00
	Key14	71,55	64,00	71,55	90,51	115,38	143,11	45,25	32,00	45,25	71,55	101,19	131,94	32,00	0,00	32,00
	Key15	90,51	71,55	64,00	71,55	90,51	115,38	71,55	45,25	32,00	45,25	71,55	101,19	64,00	32,00	0,00
	Key16	115,38	90,51	71,55	64,00	71,55	90,51	101,19	71,55	45,25	32,00	45,25	71,55	96,00	64,00	32,00
	Key17	143,11	115,38	90,51	71,55	64,00	71,55	131,94	101,19	71,55	45,25	32,00	45,25	128,00	96,00	64,00
	Key18	172,33	143,11	115,38	90,51	71,55	64,00	163,17	131,94	101,19	71,55	45,25	32,00	160,00	128,00	96,00
	Key19	96,00	101,19	115,38	135,76	160,00	186,59	64,00	71,55	90,51	115,38	143,11	172,33	32,00	45,25	71,55
	Key20	101,19	96,00	101,19	115,38	135,76	160,00	71,55	64,00	71,55	90,51	115,38	143,11	45,25	32,00	45,25
	Key21	115,38	101,19	96,00	101,19	115,38	135,76	90,51	71,55	64,00	71,55	90,51	115,38	71,55	45,25	32,00
	Key22	135,76	115,38	101,19	96,00	101,19	115,38	115,38	90,51	71,55	64,00	71,55	90,51	101,19	71,55	45,25
	Key23	160,00	135,76	115,38	101,19	96,00	101,19	143,11	115,38	90,51	71,55	64,00	71,55	131,94	101,19	71,55
	Key24	186,59	160,00	135,76	115,38	101,19	96,00	172,33	143,11	115,38	90,51	71,55	64,00	163,17	131,94	101,19
	Key25	128,00	131,94	143,11	160,00	181,02	204,90	96,00	101,19	115,38	135,76	160,00	186,59	64,00	71,55	90,51
	Key26	131,94	128,00	131,94	143,11	160,00	181,02	101,19	96,00	101,19	115,38	135,76	160,00	71,55	64,00	71,55
	Key27	143,11	131,94	128,00	131,94	143,11	160,00	115,38	101,19	96,00	101,19	115,38	135,76	90,51	71,55	64,00
	Key28	160,00	143,11	131,94	128,00	131,94	143,11	135,76	115,38	101,19	96,00	101,19	115,38	115,38	90,51	71,55
	Key29	181,02	160,00	143,11	131,94	128,00	131,94	160,00	135,76	115,38	101,19	96,00	101,19	143,11	115,38	90,51
	Key30	204,90	181,02	160,00	143,11	131,94	128,00	186,59	160,00	135,76	115,38	101,19	96,00	172,33	143,11	115,38

Table B.2. (*cont'd)* Distance matrix for square keyboard layout model



		Finishing Key of Movement (Target Key)														
		Key16	**Key17**	**Key18**	**Key19**	**Key20**	**Key21**	**Key22**	**Key23**	**Key24**	**Key25**	**Key26**	**Key27**	**Key28**	**Key29**	**Key30**
Starting Key of Movement	**Key1**	115,38	143,11	172,33	96,00	101,19	115,38	135,76	160,00	186,59	128,00	131,94	143,11	160,00	181,02	204,90
	Key2	90,51	115,38	143,11	101,19	96,00	101,19	115,38	135,76	160,00	131,94	128,00	131,94	143,11	160,00	181,02
	Key3	71,55	90,51	115,38	115,38	101,19	96,00	101,19	115,38	135,76	143,11	131,94	128,00	131,94	143,11	160,00
	Key4	64,00	71,55	90,51	135,76	115,38	101,19	96,00	101,19	115,38	160,00	143,11	131,94	128,00	131,94	143,11
	Key5	71,55	64,00	71,55	160,00	135,76	115,38	101,19	96,00	101,19	181,02	160,00	143,11	131,94	128,00	131,94
	Key6	90,51	71,55	64,00	186,59	160,00	135,76	115,38	101,19	96,00	204,90	181,02	160,00	143,11	131,94	128,00
	Key7	101,19	131,94	163,17	64,00	71,55	90,51	115,38	143,11	172,33	96,00	101,19	115,38	135,76	160,00	186,59
	Key8	71,55	101,19	131,94	71,55	64,00	71,55	90,51	115,38	143,11	101,19	96,00	101,19	115,38	135,76	160,00
	Key9	45,25	71,55	101,19	90,51	71,55	64,00	71,55	90,51	115,38	115,38	101,19	96,00	101,19	115,38	135,76
	Key10	32,00	45,25	71,55	115,38	90,51	71,55	64,00	71,55	90,51	135,76	115,38	101,19	96,00	101,19	115,38
	Key11	45,25	32,00	45,25	143,11	115,38	90,51	71,55	64,00	71,55	160,00	135,76	115,38	101,19	96,00	101,19
	Key12	71,55	45,25	32,00	172,33	143,11	115,38	90,51	71,55	64,00	186,59	160,00	135,76	115,38	101,19	96,00
	Key13	96,00	128,00	160,00	32,00	45,25	71,55	101,19	131,94	163,17	64,00	71,55	90,51	115,38	143,11	172,33
	Key14	64,00	96,00	128,00	45,25	32,00	45,25	71,55	101,19	131,94	71,55	64,00	71,55	90,51	115,38	143,11
	Key15	32,00	64,00	96,00	71,55	45,25	32,00	45,25	71,55	101,19	90,51	71,55	64,00	71,55	90,51	115,38
	Key16	0,00	32,00	64,00	101,19	71,55	45,25	32,00	45,25	71,55	115,38	90,51	71,55	64,00	71,55	90,51
	Key17	32,00	0,00	32,00	131,94	101,19	71,55	45,25	32,00	45,25	143,11	115,38	90,51	71,55	64,00	71,55
	Key18	64,00	32,00	0,00	163,17	131,94	101,19	71,55	45,25	32,00	172,33	143,11	115,38	90,51	71,55	64,00
	Key19	101,19	131,94	163,17	0,00	32,00	64,00	96,00	128,00	160,00	32,00	45,25	71,55	101,19	131,94	163,17
	Key20	71,55	101,19	131,94	32,00	0,00	32,00	64,00	96,00	128,00	45,25	32,00	45,25	71,55	101,19	131,94
	Key21	45,25	71,55	101,19	64,00	32,00	0,00	32,00	64,00	96,00	71,55	45,25	32,00	45,25	71,55	101,19
	Key22	32,00	45,25	71,55	96,00	64,00	32,00	0,00	32,00	64,00	101,19	71,55	45,25	32,00	45,25	71,55
	Key23	45,25	32,00	45,25	128,00	96,00	64,00	32,00	0,00	32,00	131,94	101,19	71,55	45,25	32,00	45,25
	Key24	71,55	45,25	32,00	160,00	128,00	96,00	64,00	32,00	0,00	163,17	131,94	101,19	71,55	45,25	32,00
	Key25	115,38	143,11	172,33	32,00	45,25	71,55	101,19	131,94	163,17	0,00	32,00	64,00	96,00	128,00	160,00
	Key26	90,51	115,38	143,11	45,25	32,00	45,25	71,55	101,19	131,94	32,00	0,00	32,00	64,00	96,00	128,00
	Key27	71,55	90,51	115,38	71,55	45,25	32,00	45,25	71,55	101,19	64,00	32,00	0,00	32,00	64,00	96,00
	Key28	64,00	71,55	90,51	101,19	71,55	45,25	32,00	45,25	71,55	96,00	64,00	32,00	0,00	32,00	64,00
	Key29	71,55	64,00	71,55	131,94	101,19	71,55	45,25	32,00	45,25	128,00	96,00	64,00	32,00	0,00	32,00
	Key30	90,51	71,55	64,00	163,17	131,94	101,19	71,55	45,25	32,00	160,00	128,00	96,00	64,00	32,00	0,00

Table B.3. Newspaper articles used while generating F matrix for Turkish

#	Newspaper	Columnist	Internet Link
1	Hurriyet	Ahmet Altan	http://hurarsiv.hurriyet.com.tr/goster/haber.aspx?id=3542951&
2	Hurriyet	Doğan Hakyemez	http://hurarsiv.hurriyet.com.tr/goster/haber.aspx?id=4296105
3	Hurriyet	Altan Tanrıkulu	http://hurarsiv.hurriyet.com.tr/goster/haber.aspx?id=4632705
4	Hurriyet	Ahmet Hakan	http://hurarsiv.hurriyet.com.tr/goster/haber.aspx?id=8828579
5	Hurriyet	Zeynep BÖLÜKBAŞI	http://hurarsiv.hurriyet.com.tr/goster/haber.aspx?id=8719022&
6	Akşam	Nagehan Alçı	http://www.aksam.com.tr/2009/04/08/yazar/170/
7	Akşam	Ali Saydam	http://www.aksam.com.tr/2009/04/08/yazar/10807/aksam/yazi.html
8	Akşam	Bahri Havadır	http://www.aksam.com.tr/2009/04/08/yazar/11522/
9	Akşam	Serdar Turgut	http://www.aksam.com.tr/2009/04/08/yazar/4878/aksam/yazi.html
10	Milliyet	Melih Aşık	http://www.milliyet.com.tr/Yazar.aspx?aType=
11	Milliyet	Fikret Bila	http://www.milliyet.com.tr/Yazar.aspx?aType=
12	Milliyet	Güneri Civaoğlu	http://www.milliyet.com.tr/Yazar.aspx?aType=YazarDetayArsiv&ArticleID..
13	Milliyet	Abbas Güçlü	http://www.milliyet.com.tr/Yazar.aspx?aType=YazarDetayArsiv&Article..
14	Milliyet	Hurşit Güneş	http://www.milliyet.com.tr/Yazar.aspx?aType=
15	Sabah	Engin Ardıç	http://www.sabah.com.tr/2009/04/06/
16	Sabah	Mehmet Barlas	http://www.sabah.com.tr/2009/03/31/
17	Sabah	Emre Aköz	http://www.sabah.com.tr/2009/03/22/
18	Sabah	Umur Talu	http://www.sabah.com.tr/2009/03/08/talu.html
19	Sabah	Erdal Şafak	http://www.sabah.com.tr/2009/03/04/
20	Vatan	Asaf Savaş Akat	http://haber.gazetevatan.com/haber.vatan?
21	Vatan	Okay Gönensin	http://haber.gazetevatan.com/haber.vatan?
22	Vatan	Reha Muhtar	http://haber.gazetevatan.com/
23	Zaman	Etyen Mahçupyan	http://www.zaman.com.tr/yazar.do?yazarno=1032
24	Zaman	Hüseyin Gülerce	http://www.zaman.com.tr/yazar.do?yazino=824761
25	Zaman	Fikret Ertan	http://www.zaman.com.tr/yazar.do?yazino=831366
26	Milli Gazete	Mehmet Şevki Eygi	http://www.milligazete.com.tr/makale/ilimli-islâm-tuzagi-121697.htm
27	Cumhuriyet	Pınar Keleş	http://www.cumhuriyet.com.tr/?im=yhs&hn=49280
28	Radikal	Hakkı Devrim	http://www.radikal.com.tr/
29	Radikal	Murat Yetkin	http://www.radikal.com.tr/
30	Radikal	İsmet Berkan	http://www.radikal.com.tr/
31	Evrensel	Yücel Özdemir	http://www.evrensel.net/haber.php?haber_id=48935
32	Bugün	Toktamış Ateş	http://www.bugun.com.tr/kose-yazisi/65543-kriz-sorunu
33	Fanatik	Can Çobanoğlu	http://fanatik.ekolay.net/Fanatik/Index.aspx?
34	Yeniçağ	Serap Besimoğlu	http://www.yenicaggazetesi.com.tr/a_haberdetay.php?hityaz=7895
35	Türkiye	Metiner Sezer	http://www.turkiyegazetesi.com.tr/makaledetay.aspx?ID=404456

Table B.3. (*cont'd)* Newspaper articles used while generating F matrix for Turkish

#	Newspaper	Columnist	Internet Link
36	Türkiye	Said Arvaz	http://www.turkiyegazetesi.com.tr/makaledetay.aspx?id=405369
37	Yeni Şafak	İbrahim Karagül	http://yenisafak.com.tr/Yazarlar/?t=09.04.2009&y=IbrahimKaragul
38	Yeni Şafak	Fehmi Koru	http://yenisafak.com.tr/Yazarlar/?i=16008&y=FehmiKoru
39	Yeni Şafak	Davut Dursun	http://yenisafak.com.tr/Yazarlar/?t=09.04.2009&y=DavutDursun
40	Güneş	İdil Çeliker	http://www.gunes.com/2009/04/09/yazarlar/yi.html
41	Ortadoğu	Taylan Sorgun	http://www.ortadogugazetesi.net/makale.php?y
42	Star Gazete	Ardan Zentürk	http://www.stargazete.com/gazete/yazar/
43	Star Gazete	Alin Taşçıyan	http://www.stargazete.com/gazete/yazar/
44	Star Gazete	Ahmet Kekeç	http://www.stargazete.com/gazete/yazar/
45	Milliyet	Taha Akyol	http://www.milliyet.com.tr/Yazar.aspx?aType=YazarDetay..
46	Milliyet	Rıza Türmen	http://www.milliyet.com.tr/Yazar.aspx?aType=
47	Güneş	Rıza Zelyut	http://www.gunes.com/2009/10/18/yazarlar/y4.html,
48	Takvim	Talip Emiroğlu	http://www.takvim.com.tr/Yazarlar/emiroglu
49	Vatan	Can Ataklı	http://haber.gazetevatan.com/haberdetay.asp?
50	Evrensel	İzzettin Önder	http://www.evrensel.net/haber.php?haber_id=59573

Table B.4. Magazine articles used while generating F matrix for Turkish

#	Magazine	Subject Link
1	Anadolu Gençlik	http://www.anadolugenclik.com.tr/tr/default.asp?p=oku&id=903
2	Anadolu Gençlik	http://www.anadolugenclik.com.tr/tr/default.asp?p=oku&id=909
3	Bilim ve Teknik	http://www.biltek.tubitak.gov.tr/gelisim/psikoloji/klinik.htm#nlp
4	Bilim ve Teknik	http://www.biltek.tubitak.gov.tr/gelisim/psikoloji/biyopsi
5	Bilim ve Teknik	http://www.biltek.tubitak.gov.tr/gelisim/psikoloji/beyin.htm
6	İlmi Araştırma	http://www.ilmiarastirma.net/?Pg=Detail&Number=12434
7	PC World	http://www.pcworld.com.tr/index.php?bolum=inceleme 13.04.2009
8	PC World	http://www.pcworld.com.tr/kapatilabilecek-vista
9	Newsweek Türkiye	http://www.newsweek.com.tr/haberler/detay/20348/Tuncay-Guney-kimdir
10	Ötüken	http://www.otuken.net/modules.php?name=News&file=cate
11	Atlas	http://www.kesfetmekicinbak.com/doga/
12	Sivil Toplum Dergisi	http://www.siviltoplum.com.tr/?ynt=icerikdetay&icerik=46&id=101
13	Blue Jean Müzik	http://www.bluejean.com.tr/roportaj/08817/
14	Medikal Bakış	http://www.medikalbakis.net/5/5001.htm
15	Öğretmenin Sesi	http://www.ogretmeninsesi.org/r_egitim.asp
16	Genç Gelişim	http://www.gencgelisim.com/v2/content/view/856/2/
17	Nakliye Dergisi	http://www.nakliyedergisi.com/Kose_Yazilari-op-viewarticle-artid-17
18	Konut Dergisi	http://konutdergisi.com/habergoruntule.asp?bolum=372&katid=21
19	Sultan Şehir Edebiyat	http://www.sultansehir.com/?mrt=dergiayrinti&verix
20	Ev&Kültür Dergisi	http://www.evkultur.com/cevre/enerjiyasamin/enerjiyasamincekirdegi.htm
21	Fotoğrafya	http://www.fotografya.gen.tr/cnd/index.php?id=305,526,0,0,1,0
22	Marketing Türkiye	http://www.marketingturkiye.com/yeni/Yazarlar/Yazar_Detay.aspx?id=648
23	THY Sky Life	http://www.thy.com/tr-TR/corporate/skylife/article.aspx?mkl=1109
24	Gülistan	http://www.gulistandergisi.com/dergi_oku.php?id=625
25	Capital	http://www.capital.com.tr/haber.aspx?HBR_KOD=5312
26	Turkish Time	http://www.turkishtime.org/files/arastirmalar/ihracat/İhracat_2005.doc
27	National Geographic Türkiye	http://www.nationalgeographic.com.tr/ngm/0904/default.aspx# all articles
28	Cinemascope Sinema Dergisi	http://www.cinemascopedergisi.com/azkaldi.asp?ID=23
29	Aksiyon	http://www.aksiyon.com.tr/detay.php?id=32715
30	Ankebut and Meryem parts in Quran	http://www.kuranikerim.com/m_diyanet_index.htm
31	Diplomat	http://www.diplomat.com.tr/atlas/sayilar/sayi5/default.asp
32	Doktor Dergisi	http://www.doktordergisi.com/haberdetay.asp?id=18
33	Doktor Dergisi	http://www.doktordergisi.com/haberdetay.asp?id=17
34	Doktor Dergisi	http://www.doktordergisi.com/49/haberdetay.asp?id=17
35	Kapalıçarşı Dergisi	http://www.kapalicarsidergisi.com/tr/konular/eski_gunler.htm

Table B.4. (*cont'd)* Magazine articles used while generating F matrix for Turkish

#	Magazine	Subject Link
36	Kapalıçarşı Dergisi	http://www.kapalicarsidergisi.com/tr/konular/ayna.htm
37	Sınav Dergisi	http://www.sinav.com.tr/yeni/index.php?sID=29&gID=4&cID=9
38	4-4-2 Spor Dergisi	http://www.442dergi.com/editor.php?id=6
39	Yolcu Dergisi	http://www.yolcudergisi.com/?p=448
40	Köklü Değişim Edebiyat	http://kokludegisim.net/hbroku.php?sayfa=haberoku&id=423
41	Newsweek Türkiye	http://www.newsweek.com.tr/haberler/detay/33076/Yeni-bir-gorev
42	Newsweek Türkiye	http://www.newsweek.com.tr/haberler/detay/32944/Eylem-icin-ekran-basina
43	Gökyüzü Haberci	http://www.gokyuzuhaberci.com/yazar-derya-zengin/
44	Kardelen Dergisi	http://www.kardelendergisi.com/yazi.php?yazi=765
45	Cinemascope Sinema Dergisi	http://www.sinemaloji.com/?p=10528
46	Telepati Telekom Dergisi	http://www.telepati.com/ekim09/konu2.htm
47	Diplomat	http://www.diplomat.com.tr/atlas/sayilar/sayi6/
48	Altı Sigma Forum Dergisi	http://www.altisigma.gen.tr/sonsayi.html
49	Para Ekonomi Dergisi	http://www.paradergi.com.tr/yaz8-308-28,252@300.html
50	Turkish Time	http://www.turkishtime.org/tr/content.asp?PID={CAD0

Table B.5. Turkish classics used while generating F matrix for Turkish

#	Writer	Turkish classic	#	Writer	Turkish classic
1	Reşat Nuri Güntekin	Yaprak Dökümü	34	Reşat Nuri Güntekin	Damga
2	Rakım Çalapala	87 Oğuz	35	Reşat Nuri Güntekin	Akşam Güneşi
3	Hüseyin Rahmi Gürpınar	Gulyabani	36	Ayşe Kulin	Gece Sesleri
4	Namık Kemal	Vatan Yahut Silistre	37	Orhan Pamuk	Masumiyet Müzesi
5	Ömer Seyfettin	Yalnız Efe	38	Peyami Safa	Dokuzuncu Hariciye Koğuşu
6	Ömer Seyfettin	Harem	39	Sait Faik Abasıyanık	Kayıp Aranıyor
7	Ömer Seyfettin	Beyaz Lale	40	Nihat Sami Banarlı	Türkçe'nin Sırları
8	Ömer Seyfettin	Bomba	41	Cengiz Aytmatov	Gün Olur Asra Bedel
9	Sam Paşazade Sezai	Sergüzeşt	42	Hüseyin Rahmi Gürpınar	Mezarından Kalkan Şehit
10	Yusuf Atılgan	Anayurt Oteli	43	Tarık Buğra	Osmancık
11	Y. K. Karaosmanoğlu	Yaban	44	Y. K. Karaosmanoğlu	Anamın Kitabı
12	Tarık Buğra	Küçük Ağa	45	Ahmet Rasim	İki Güzel Günahkar
13	Divan Edebiyatı	Kerem ile Aslı	46	Kerime Nadir	Solan Umut
14	Recaizade Mahmut Ekrem	Araba Sevdası	47	Necati Cumalı	Zeliş
15	Peyami Safa	Fatih Harbiye	48	Sezen Özol	Çanak. Ask. Rütbe Gerek.
16	Ayşe Kulin	Sevdalinka	49	E. Mahmut Karakurt	Dağları Bekleyen Kız
17	Namık Kemal	İntibah	50	Sabahattin Ali	Kuyucaklı Yusuf
18	Y. K. Karaosmanoğlu	Kiralık Konak	51	Hakkı Kamil Beşe	Tek Çarık Yüzbaşı
19	Halide Edip Adıvar	Sinekli Bakkal	52	Peyami Safa	Yalnızız
20	Halit Ziya Uşaklıgil	Kırık Hayatlar	53	Esat Mahmut Kurt	Tren
21	Y. K. Karaosmanoğlu	Sodome ve Gomore	54	Kerime Nadir	Solan Umut
22	Orhan Pamuk	Kar	55	Refik Halid Karay	İstanbul'un Bir Yüzü
23	Orhan Pamuk	Benim Adım Kırmızı	56	Refik Halid Karay	Ekmek Elden Su Gölden
24	Turgut Özakman	Şu Çılgın Türkler	57	Buket Uzuner	Gelibolu
25	Kemalettin Tuğcu	Kuklacı	58	Yaşar Kemal	İnce Memed
26	Halit Ziya Uşaklıgil	Aşk-ı Memnu	59	Halide Edip Adıvar	Handan
27	Reşat Nuri Güntekin	Bir Kadın Düşmanı	60	Orhan Kemal	Eskici Dükkanı
28	Ahmet Rasim	Falaka	61	Refik Halid Karay	Anahtar
29	Ahmet Hamdi Tanpınar	Beş Şehir	62	Orhan Hançerlioğlu	Oyun
30	Mustafa Kemal Atatürk	Nutuk	63	Ömer Seyfettin	Kaşağı
31	Salah Birsel	Boğaziçi Şıngır Mıngır	64	Gaye Hiçyılmaz	Fırtınaya
32	Yusuf Has Hacip	Kutadgu Bilig	65	Cengiz Dağcı	Korkunç Yıllar
33	Halide Nusret Zorlutuna	Benim Küçük Dostlarım	66	Ahmet Günbay Yıldız	Çiçekler Susayınca

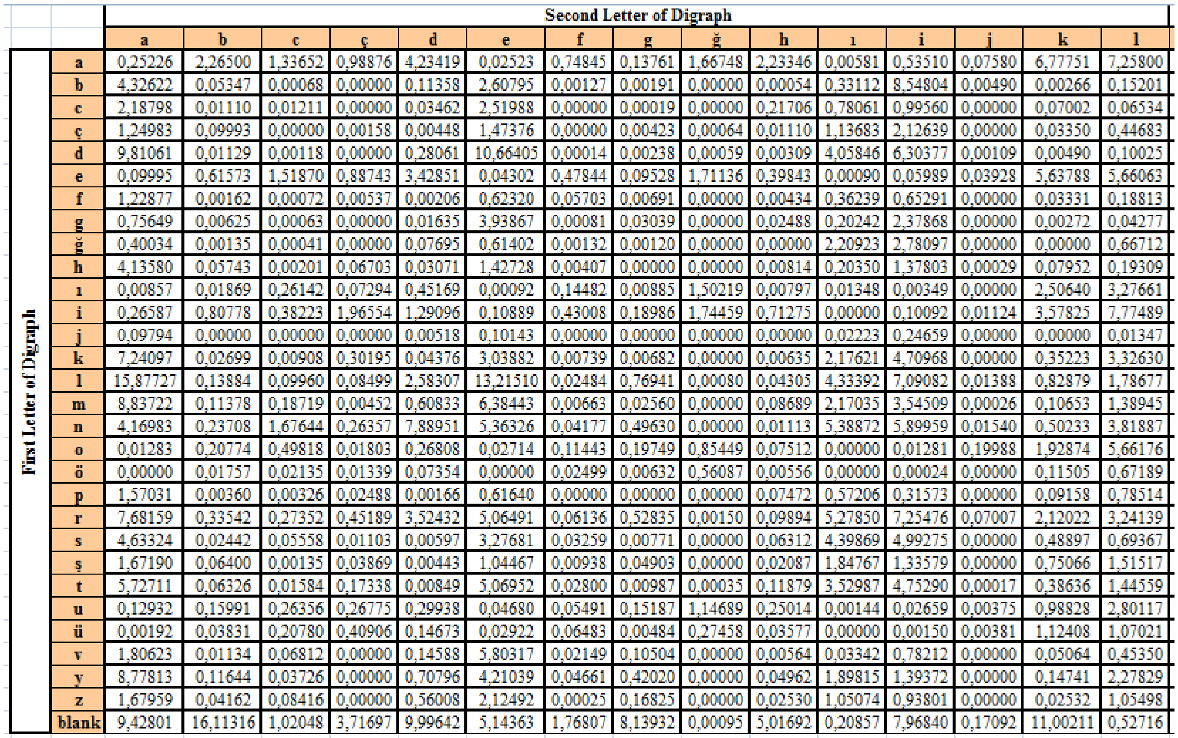
Table B.6. World classics used while generating F matrix of Turkish

#	World classics	#	World classics	#	World classics
1	Sefiller	37	Derviş ve Ölüm	73	Martı
2	Açlık	38	Dişi Kurdun Rüyaları	74	Melekler ve Şeytanlar
3	Beyaz Diş	39	Doğmamış Çocuğa Mektup	75	Monte Kristo Kontu
4	Beyaz Gemi	40	Doğunun Limanları	76	Hikaleyer
5	Büyük Umutlar	41	Don Kişot	77	Truva
6	Vadideki Zambak	42	Drina Köprüsü	78	Oliver Twist
7	Cimri	43	Duman	79	Ölü Canlar
8	Soygun	44	Dünya Nimeti	80	On Küçük Zenci
9	Simyacı	45	Elveda Gülsarı	81	Onlar da İnsandı
10	Eksik Parçalar	46	Eugenie Grandet	82	Pastoral Senfoni
11	Define Adası	47	Fareler ve İnsanlar	83	Robinson Crusoe
12	Madam Bovary	48	Gazap Üzümleri	84	Savaş ve Barış
13	Dünyanın Ucun. Fener	49	Genç Werther'in Acıları	85	Saydam Şeyler
14	Sol Ayağım	50	Gönülçelen	86	Sefiller
15	Diriliş	51	Gora	87	Şeker Portakalı
16	Ferrarisini Satan Bilge	52	Gülün Adı	88	Seksen Günde Devri Alem
17	Alice Harikalar Diyarında	53	Gün Olur Asra Bedel	89	Sel. Eyy. ve As. Yür. Rich.
18	Melekler ve Şeytanlar	54	Güneşi Uyandıralım	90	Ses ve Öfke
19	Ölü Ozanlar Derneği	55	Hamlet	91	Sidarta
20	Acı Kahve	56	İki Şehrin Hikayesi	92	Silahlara Veda
21	Acımasız Miras	57	İnci	93	Simyacı
22	Şeker Portakalı	58	İtiraflarım	94	Suç ve Ceza
23	Ramses, Kadeş Savaşı	59	Ivan Denisoviç'in Bir Günü	95	Tom Sawyer'ın Maceraları
24	Açlık	60	Kadınlar Okulu	96	Tütün Sarı Dünya
25	Akdeniz	61	Kanlı Oyun	97	Vadideki Zambak
26	Ana	62	Karamazov Kardeşler	98	Vahşetin Çağrısı
27	Aşkın Üç Harfi	63	Kat. Blum'un Çiğ. On.	99	Veronica Ölmek İstiyor
28	Babalar ve Oğulları	64	Kazaklar	100	Yabancı
29	Beyaz Diş	65	Klimanjaro'nun Karları	101	Yargıç ve Celladı
30	Beyaz Gemi	66	Kimsesiz Çocuk	102	Yer Altından Notlar
31	Budala	67	Kiraz Çiçekleri	103	Yol
32	Büyük Umutlar	68	Kırmızı Pazartesi	104	Yür. Götür. Yere Git
33	Çanlar Kimin İ.Ç.	69	Kroyçer Sonat	105	Yüzbaşının Kızı
34	Cimri	70	Kumarbaz	106	Zorba
35	Çocukluğum	71	Kutsal Sığınak	107	Zorlu Günler
36	Değişim	72	Madam Bovary		

Table B.7. Random academic articles used for F matrix of Turkish

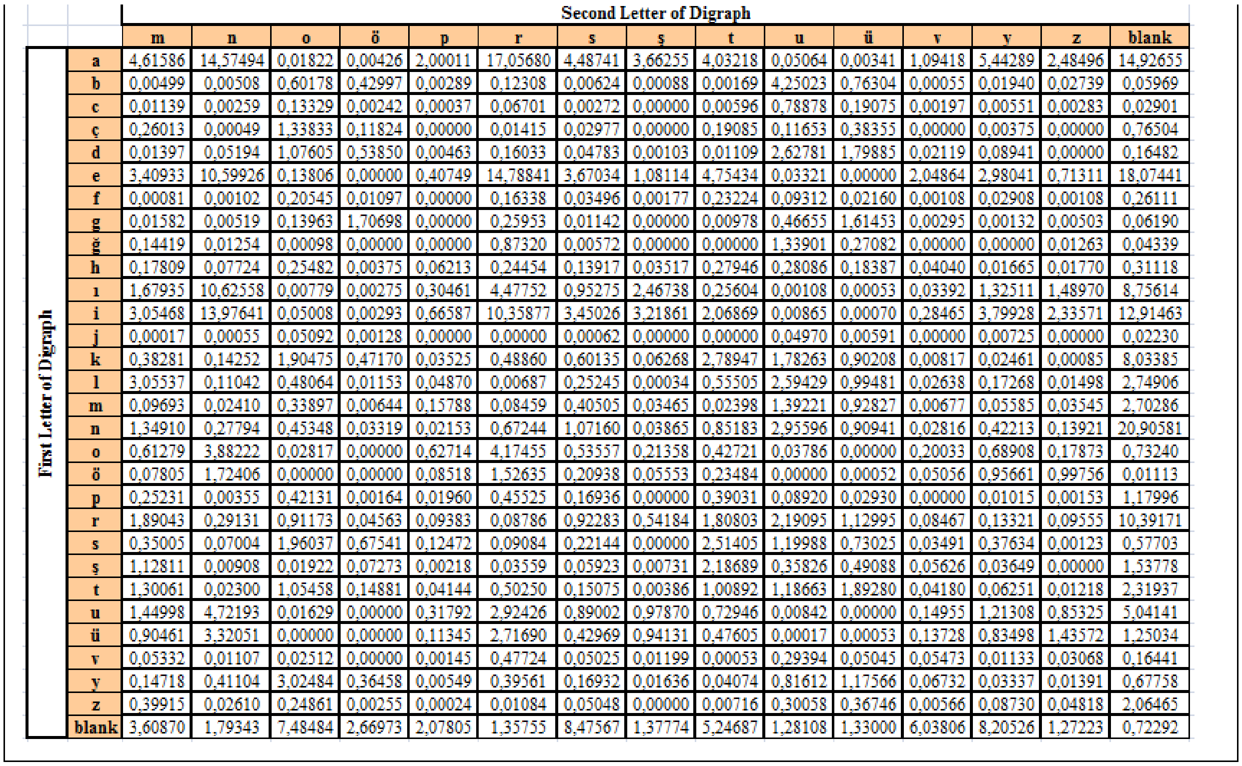
#	Random academic subject
1	Çatışma Teorisi Bağlamında Depresyonun Sınıfsal Karakteri
2	Batı ve Öteki
3	Modern Türkiyenin Felsefesi Kökenleri
4	İletişim Araçları ve Sosyal İlişkiler
5	İlk Şehir Planlaması
6	Hititlerde Sanat
7	Radyo Karbonlama Tarih Yöntemi
8	İstanbul Mahkemeleri
9	Marksizim ve Antropoloji
10	Türkiye-AB İlişkileri
11	Sualtı Hazinelerimiz
12	Editör Gözüyle Danışmanlık
13	Türk Bankacılık Sektöründeki Riskler
14	Vadeli İşlem Piyasalarının Gelişimi ve Fonksiyonu
15	Hükümet Dışı Kuruluşlar
16	Burjuvazi ve Bilim
17	Bilge Köyü Katliamı'nın Gündeme Getirdikleri
18	Atatürk'ün Kurduğu Kurumlar
19	Türk Modernleşmesinin Temelleri: 3. Selim
20	Yeni Osmanlılar'ın Ülkeye Dönmelerinden Sonraki Faaaliyetleri
21	Ceviz'in Faydaları
22	Sigara'nın Zararları
23	Öğrenen Okul
24	Öğretmenlerde Tükenmişlik
25	Drama Sanatı
26	Duygusal Zeka
27	Çoklu Zeka
28	Bireysel Çalışma Yöntemi
29	İlköğretimde Türkçe Eğitimi
30	Küreselleşme Sürecinde Meslek ve İş
31	İş Sağlığı ve Güvenliği
32	Yeni Türk Lirasının Başarı İlkeleri
33	Türkiye'de Yabancı Sermaye Yatırımları
34	Türk Edebiyatında Makaleler

Table B.8. Digraph frequencies of Turkish language (in one thousand)



		Second Letter of Digraph														
		a	**b**	**c**	**ç**	**d**	**e**	**f**	**g**	**ğ**	**h**	**ı**	**i**	**j**	**k**	**l**
First Letter of Digraph	**a**	0,25226	2,26500	1,33652	0,98876	4,23419	0,02523	0,74845	0,13761	1,66748	2,23346	0,00581	0,53510	0,07580	6,77751	7,25800
	b	4,32622	0,05347	0,00068	0,00000	0,11358	2,60795	0,00127	0,00191	0,00000	0,00054	0,33112	8,54804	0,00490	0,00266	0,15201
	c	2,18798	0,01110	0,01211	0,00000	0,03462	2,51988	0,00000	0,00019	0,00000	0,21706	0,78061	0,99560	0,00000	0,07002	0,06534
	ç	1,24983	0,09993	0,00000	0,00158	0,00448	1,47376	0,00000	0,00423	0,00064	0,01110	1,13683	2,12639	0,00000	0,03350	0,44683
	d	9,81061	0,01129	0,00118	0,00000	0,28061	10,66405	0,00014	0,00238	0,00059	0,00309	4,05846	6,30377	0,00109	0,00490	0,10025
	e	0,09995	0,61573	1,51870	0,88743	3,42851	0,04302	0,47844	0,09528	1,71136	0,39843	0,00090	0,05989	0,03928	5,63788	5,66063
	f	1,22877	0,00162	0,00072	0,00537	0,00206	0,62320	0,05703	0,00691	0,00000	0,00434	0,36239	0,65291	0,00000	0,03331	0,18813
	g	0,75649	0,00625	0,00063	0,00000	0,01635	3,93867	0,00081	0,03039	0,00000	0,02488	0,20242	2,37868	0,00000	0,00272	0,04277
	ğ	0,40034	0,00135	0,00041	0,00000	0,07695	0,61402	0,00132	0,00120	0,00000	0,00000	2,20923	2,78097	0,00000	0,00000	0,66712
	h	4,13580	0,05743	0,00201	0,06703	0,03071	1,42728	0,00407	0,00000	0,00000	0,00814	0,20350	1,37803	0,00029	0,07952	0,19309
	ı	0,00857	0,01869	0,26142	0,07294	0,45169	0,00092	0,14482	0,00885	1,50219	0,00797	0,01348	0,00349	0,00000	2,50640	3,27661
	i	0,26587	0,80778	0,38223	1,96554	1,29096	0,10889	0,43008	0,18986	1,74459	0,71275	0,00000	0,10092	0,01124	3,57825	7,77489
	j	0,09794	0,00000	0,00000	0,00000	0,00518	0,10143	0,00000	0,00000	0,00000	0,00000	0,02223	0,24659	0,00000	0,00000	0,01347
	k	7,24097	0,02699	0,00908	0,30195	0,04376	3,03882	0,00739	0,00682	0,00000	0,00635	2,17621	4,70968	0,00000	0,35223	3,32630
	l	15,87727	0,13884	0,09960	0,08499	2,58307	13,21510	0,02484	0,76941	0,00080	0,04305	4,33392	7,09082	0,01388	0,82879	1,78677
	m	8,83722	0,11378	0,18719	0,00452	0,60833	6,38443	0,00663	0,02560	0,00000	0,08689	2,17035	3,54509	0,00026	0,10653	1,38945
	n	4,16983	0,23708	1,67644	0,26357	7,88951	5,36326	0,04177	0,49630	0,00000	0,01113	5,38872	5,89959	0,01540	0,50233	3,81887
	o	0,01283	0,20774	0,49818	0,01803	0,26808	0,02714	0,11443	0,19749	0,85449	0,07512	0,00000	0,01281	0,19988	1,92874	5,66176
	ö	0,00000	0,01757	0,02135	0,01339	0,07354	0,00000	0,02499	0,00632	0,56087	0,00556	0,00000	0,00024	0,00000	0,11505	0,67189
	p	1,57031	0,00360	0,00326	0,02488	0,00166	0,61640	0,00000	0,00000	0,00000	0,07472	0,57206	0,31573	0,00000	0,09158	0,78514
	r	7,68159	0,33542	0,27352	0,45189	3,52432	5,06491	0,06136	0,52835	0,00150	0,09894	5,27850	7,25476	0,07007	2,12022	3,24139
	s	4,63324	0,02442	0,05558	0,01103	0,00597	3,27681	0,03259	0,00771	0,00000	0,06312	4,39869	4,99275	0,00000	0,48897	0,69367
	ş	1,67190	0,06400	0,00135	0,03869	0,00443	1,04467	0,00938	0,04903	0,00000	0,02087	1,84767	1,33579	0,00000	0,75066	1,51517
	t	5,72711	0,06326	0,01584	0,17338	0,00849	5,06952	0,02800	0,00987	0,00035	0,11879	3,52987	4,75290	0,00017	0,38636	1,44559
	u	0,12932	0,15991	0,26356	0,26775	0,29938	0,04680	0,05491	0,15187	1,14689	0,25014	0,00144	0,02659	0,00375	0,98828	2,80117
	ü	0,00192	0,03831	0,20780	0,40906	0,14673	0,02922	0,06483	0,00484	0,27458	0,03577	0,00000	0,00150	0,00381	1,12408	1,07021
	v	1,80623	0,01134	0,06812	0,00000	0,14588	5,80317	0,02149	0,10504	0,00000	0,00564	0,03342	0,78212	0,00000	0,05064	0,45350
	y	8,77813	0,11644	0,03726	0,00000	0,70796	4,21039	0,04661	0,42020	0,00000	0,04962	1,89815	1,39372	0,00000	0,14741	2,27829
	z	1,67959	0,04162	0,08416	0,00000	0,56008	2,12492	0,00025	0,16825	0,00000	0,02530	1,05074	0,93801	0,00000	0,02532	1,05498
	blank	9,42801	16,11316	1,02048	3,71697	9,99642	5,14363	1,76807	8,13932	0,00095	5,01692	0,20857	7,96840	0,17092	11,00211	0,52716

Table B.8. (*cont'd)* Digraph frequencies of Turkish language (*in one thousand)*



	Second Letter of Digraph														
First Letter of Digraph	m	n	o	ö	p	r	s	ş	t	u	ü	v	y	z	blank
a	4,61586	14,57494	0,01822	0,00426	2,00011	17,05680	4,48741	3,66255	4,03218	0,05064	0,00341	1,09418	5,44289	2,48496	14,92655
b	0,00499	0,00508	0,60178	0,42997	0,00289	0,12308	0,00624	0,00088	0,00169	4,25023	0,76304	0,00055	0,01940	0,02739	0,05969
c	0,01139	0,00259	0,13329	0,00242	0,00037	0,06701	0,00272	0,00000	0,00596	0,78878	0,19075	0,00197	0,00551	0,00283	0,02901
ç	0,26013	0,00049	1,33833	0,11824	0,00000	0,01415	0,02977	0,00000	0,19085	0,11653	0,38355	0,00000	0,00375	0,00000	0,76504
d	0,01397	0,05194	1,07605	0,53850	0,00463	0,16033	0,04783	0,00103	0,01109	2,62781	1,79885	0,02119	0,08941	0,00000	0,16482
e	3,40933	10,59926	0,13806	0,00000	0,40749	14,78841	3,67034	1,08114	4,75434	0,03321	0,00000	2,04864	2,98041	0,71311	18,07441
f	0,00081	0,00102	0,20545	0,01097	0,00000	0,16338	0,03496	0,00177	0,23224	0,09312	0,02160	0,00108	0,02908	0,00108	0,26111
g	0,01582	0,00519	0,13963	1,70698	0,00000	0,25953	0,01142	0,00000	0,00978	0,46655	1,61453	0,00295	0,00132	0,00503	0,06190
ğ	0,14419	0,01254	0,00098	0,00000	0,00000	0,87320	0,00572	0,00000	0,00000	1,33901	0,27082	0,00000	0,00000	0,01263	0,04339
h	0,17809	0,07724	0,25482	0,00375	0,06213	0,24454	0,13917	0,03517	0,27946	0,28086	0,18387	0,04040	0,01665	0,01770	0,31118
ı	1,67935	10,62558	0,00779	0,00275	0,30461	4,47752	0,95275	2,46738	0,25604	0,00108	0,00053	0,03392	1,32511	1,48970	8,75614
i	3,05468	13,97641	0,05008	0,00293	0,66587	10,35877	3,45026	3,21861	2,06869	0,00865	0,00070	0,28465	3,79928	2,33571	12,91463
j	0,00017	0,00055	0,05092	0,00128	0,00000	0,00000	0,00062	0,00000	0,00000	0,04970	0,00591	0,00000	0,00725	0,00000	0,02230
k	0,38281	0,14252	1,90475	0,47170	0,03525	0,48860	0,60135	0,06268	2,78947	1,78263	0,90208	0,00817	0,02461	0,00085	8,03385
l	3,05537	0,11042	0,48064	0,01153	0,04870	0,00687	0,25245	0,00034	0,55505	2,59429	0,99481	0,02638	0,17268	0,01498	2,74906
m	0,09693	0,02410	0,33897	0,00644	0,15788	0,08459	0,40505	0,03465	0,02398	1,39221	0,92827	0,00677	0,05585	0,03545	2,70286
n	1,34910	0,27794	0,45348	0,03319	0,02153	0,67244	1,07160	0,03865	0,85183	2,95596	0,90941	0,02816	0,42213	0,13921	20,90581
o	0,61279	3,88222	0,02817	0,00000	0,62714	4,17455	0,53557	0,21358	0,42721	0,03786	0,00000	0,20033	0,68908	0,17873	0,73240
ö	0,07805	1,72406	0,00000	0,00000	0,08518	1,52635	0,20938	0,05553	0,23484	0,00000	0,00052	0,05056	0,95661	0,99756	0,01113
p	0,25231	0,00355	0,42131	0,00164	0,01960	0,45525	0,16936	0,00000	0,39031	0,08920	0,02930	0,00000	0,01015	0,00153	1,17996
r	1,89043	0,29131	0,91173	0,04563	0,09383	0,08786	0,92283	0,54184	1,80803	2,19095	1,12995	0,08467	0,13321	0,09555	10,39171
s	0,35005	0,07004	1,96037	0,67541	0,12472	0,09084	0,22144	0,00000	2,51405	1,19988	0,73025	0,03491	0,37634	0,00123	0,57703
ş	1,12811	0,00908	0,01922	0,07273	0,00218	0,03559	0,05923	0,00731	2,18689	0,35826	0,49088	0,05626	0,03649	0,00000	1,53778
t	1,30061	0,02300	1,05458	0,14881	0,04144	0,50250	0,15075	0,00386	1,00892	1,18663	1,89280	0,04180	0,06251	0,01218	2,31937
u	1,44998	4,72193	0,01629	0,00000	0,31792	2,92426	0,89002	0,97870	0,72946	0,00842	0,00000	0,14955	1,21308	0,85325	5,04141
ü	0,90461	3,32051	0,00000	0,00000	0,11345	2,71690	0,42969	0,94131	0,47605	0,00017	0,00053	0,13728	0,83498	1,43572	1,25034
v	0,05332	0,01107	0,02512	0,00000	0,00145	0,47724	0,05025	0,01199	0,00053	0,29394	0,05045	0,05473	0,01133	0,03068	0,16441
y	0,14718	0,41104	3,02484	0,36458	0,00549	0,39561	0,16932	0,01636	0,04074	0,81612	1,17566	0,06732	0,03337	0,01391	0,67758
z	0,39915	0,02610	0,24861	0,00255	0,00024	0,01084	0,05048	0,00000	0,00716	0,30058	0,36746	0,00566	0,08730	0,04818	2,06465
blank	3,60870	1,79343	7,48484	2,66973	2,07805	1,35755	8,47567	1,37774	5,24687	1,28108	1,33000	6,03806	8,20526	1,27223	0,72292

Table B.9. Relative frequencies of letters for Turkish language

#	**Letter**	**Frequency (per cent)**
1	blank	12.53
2	a	10.37
3	e	8.42
4	i	7.65
5	n	6.66
6	r	6.07
7	l	5.74
8	ı	4.15
9	k	3.92
10	d	3.73
11	t	3.07
12	m	3.01
13	s	2.75
14	y	2.73
15	u	2.63
16	o	2.24
17	b	2.21
18	ü	1.61
19	ş	1.46
20	z	1.18
21	g	1.16
22	v	1.05
23	ç	0.98
24	h	0.97
25	ğ	0.95
26	c	0.81
27	ö	0.74
28	p	0.72
29	f	0.42
30	j	0.06

Table B.10. The keys sorted ascending according to their ρ's in long. keyboard

Order	**Key**	**Centralization Index, ρ**
1	Key18	3,008.75
2	Key17	3,079.87
3	Key19	3,142.27
4	Key28	3,194.04
5	Key6	3,220.56
6	Key29	3,226.04
7	Key7	3,249.77
8	Key16	3,351.84
9	Key27	3,371.53
10	Key5	3,386.71
11	Key30	3,467.53
12	Key8	3,473.72
13	Key20	3,476.12
14	Key33	3,690.84
15	Key26	3,742.42
16	Key4	3,745.00
17	Key15	3,818.38
18	Key9	3,887.45
19	Key31	3,902.42
20	Key21	4,002.82
21	Key3	4,289.71
22	Key25	4,300.10
23	Key14	4,474.07
24	Key10	4,482.04
25	Key32	4,524.10
26	Key22	4,712.16
27	Key2	5,011.58
28	Key24	5,038.94
29	Key11	5,242.61
30	Key13	5,309.21
31	Key23	5,579.12
32	Key1	5,891.76
33	Key12	6,146.55

Table B.11. Letters and keys in which phase to be assigned on long. keyboard for *n*=8

Keys			Letters	
Phase to be Assigned in	Key	#	Letter	Phase to be Assigned in
Phase 1	Key18	1	blank	Initially assigned
Phase 1	Key17	2	a	Phase 1
Phase 1	Key19	3	e	Phase 1
Phase 1	Key28	4	i	Phase 1
Phase 1	Key6	5	n	Phase 1
Phase 1	Key29	6	r	Phase 1
Phase 1	Key7	7	l	Phase 1
Phase 1	Key16	8	ı	Phase 1
Phase 2	Key27	9	k	Phase 1
Phase 2	Key5	10	d	Phase 2
Phase 2	Key30	11	t	Phase 2
Phase 2	Key8	12	m	Phase 2
Phase 2	Key20	13	s	Phase 2
Initially assigned	Key33(Space Bar)	14	y	Phase 2
Phase 2	Key26	15	u	Phase 2
Phase 2	Key4	16	o	Phase 2
Phase 2	Key15	17	b	Phase 2
Phase 3	Key9	18	ü	Phase 3
Phase 3	Key31	19	ş	Phase 3
Phase 3	Key21	20	z	Phase 3
Phase 3	Key3	21	g	Phase 3
Phase 3	Key25	22	v	Phase 3
Phase 3	Key14	23	ç	Phase 3
Phase 3	Key10	24	h	Phase 3
Phase 3	Key32	25	ğ	Phase 3
Phase 4	Key22	26	c	Phase 4
Phase 4	Key2	27	ö	Phase 4
Phase 4	Key24	28	p	Phase 4
Phase 4	Key13	29	f	Phase 4
Phase 4	Key1	30	j	Phase 4

Table B.12. Letters and keys in which phase to be assigned on long. keyboard for *n*=9

Keys			**Letters**	
Phase to be Optimized in	**Key**	#	**Letter**	**Phase to be Optimized in**
Phase 1	**Key18**	1	**blank**	Initially assigned
Phase 1	**Key17**	2	**a**	Phase 1
Phase 1	**Key19**	3	**e**	Phase 1
Phase 1	**Key28**	4	**i**	Phase 1
Phase 1	**Key6**	5	**n**	Phase 1
Phase 1	**Key29**	6	**r**	Phase 1
Phase 1	**Key7**	7	**l**	Phase 1
Phase 1	**Key16**	8	**ı**	Phase 1
Phase 1	**Key27**	9	**k**	Phase 1
Phase 2	**Key5**	10	**d**	Phase 1
Phase 2	**Key30**	11	**t**	Phase 2
Phase 2	**Key8**	12	**m**	Phase 2
Phase 2	**Key20**	13	**s**	Phase 2
Initially assigned	**Key33(Space Bar)**	14	**y**	Phase 2
Phase 2	**Key26**	15	**u**	Phase 2
Phase 2	**Key4**	16	**o**	Phase 2
Phase 2	**Key15**	17	**b**	Phase 2
Phase 2	**Key9**	18	**ü**	Phase 2
Phase 2	**Key31**	19	**ş**	Phase 2
Phase 3	**Key21**	20	**z**	Phase 3
Phase 3	**Key3**	21	**g**	Phase 3
Phase 3	**Key25**	22	**v**	Phase 3
Phase 3	**Key14**	23	**ç**	Phase 3
Phase 3	**Key10**	24	**h**	Phase 3
Phase 3	**Key32**	25	**ğ**	Phase 3
Phase 3	**Key22**	26	**c**	Phase 3
Phase 3	**Key2**	27	**ö**	Phase 3
Phase 3	**Key24**	28	**p**	Phase 3
Phase 4	**Key13**	29	**f**	Phase 4
Phase 4	**Key1**	30	**j**	Phase 4

Table B.13. Letters and keys in which phase to be assigned on long. keyboard for *n*=10

Keys			Letters	
Phase to be Optimized in	Key	#	Letter	Phase to be Optimized in
Phase 1	Key18	1	blank	Initially assigned
Phase 1	Key17	2	a	Phase 1
Phase 1	Key19	3	e	Phase 1
Phase 1	Key28	4	i	Phase 1
Phase 1	Key6	5	n	Phase 1
Phase 1	Key29	6	r	Phase 1
Phase 1	Key7	7	l	Phase 1
Phase 1	Key16	8	ı	Phase 1
Phase 1	Key27	9	k	Phase 1
Phase 1	Key5	10	d	Phase 1
Phase 2	Key30	11	t	Phase 1
Phase 2	Key8	12	m	Phase 2
Phase 2	Key20	13	s	Phase 2
Initially assigned	Key33(Space Bar)	14	y	Phase 2
Phase 2	Key26	15	u	Phase 2
Phase 2	Key4	16	o	Phase 2
Phase 2	Key15	17	b	Phase 2
Phase 2	Key9	18	ü	Phase 2
Phase 2	Key31	19	ş	Phase 2
Phase 2	Key21	20	z	Phase 2
Phase 2	Key3	21	g	Phase 2
Phase 3	Key25	22	v	Phase 3
Phase 3	Key14	23	ç	Phase 3
Phase 3	Key10	24	h	Phase 3
Phase 3	Key32	25	ğ	Phase 3
Phase 3	Key22	26	c	Phase 3
Phase 3	Key2	27	ö	Phase 3
Phase 3	Key24	28	p	Phase 3
Phase 3	Key13	29	f	Phase 3
Phase 3	Key1	30	j	Phase 3

Table B.14. Letters and keys in which phase to be assigned on long. keyboard for *n*=11

Keys			Letters	
Phase to be Optimized in	**Key**	**#**	**Letter**	**Phase to be Optimized in**
Phase 1	**Key18**	1	**blank**	Initially assigned
Phase 1	**Key17**	2	**a**	Phase 1
Phase 1	**Key19**	3	**e**	Phase 1
Phase 1	**Key28**	4	**i**	Phase 1
Phase 1	**Key6**	5	**n**	Phase 1
Phase 1	**Key29**	6	**r**	Phase 1
Phase 1	**Key7**	7	**l**	Phase 1
Phase 1	**Key16**	8	**ı**	Phase 1
Phase 1	**Key27**	9	**k**	Phase 1
Phase 1	**Key5**	10	**d**	Phase 1
Phase 1	**Key30**	11	**t**	Phase 1
Phase 2	**Key8**	12	**m**	Phase 1
Phase 2	**Key20**	13	**s**	Phase 2
Initially assigned	**Key33(Space Bar)**	14	**y**	Phase 2
Phase 2	**Key26**	15	**u**	Phase 2
Phase 2	**Key4**	16	**o**	Phase 2
Phase 2	**Key15**	17	**b**	Phase 2
Phase 2	**Key9**	18	**ü**	Phase 2
Phase 2	**Key31**	19	**ş**	Phase 2
Phase 2	**Key21**	20	**z**	Phase 2
Phase 2	**Key3**	21	**g**	Phase 2
Phase 2	**Key25**	22	**v**	Phase 2
Phase 2	**Key14**	23	**ç**	Phase 2
Phase 3	**Key10**	24	**h**	Phase 3
Phase 3	**Key32**	25	**ğ**	Phase 3
Phase 3	**Key22**	26	**c**	Phase 3
Phase 3	**Key2**	27	**ö**	Phase 3
Phase 3	**Key24**	28	**p**	Phase 3
Phase 3	**Key13**	29	**f**	Phase 3
Phase 3	**Key1**	30	**j**	Phase 3

Table B.15. Letters and keys in which phase to be assigned on long. keyboard for *n*=12

Keys		**#**	**Letters**	
Phase to be Optimized in	**Key**		**Letter**	**Phase to be Optimized in**
Phase 1	**Key18**	1	**blank**	Initially assigned
Phase 1	**Key17**	2	**a**	Phase 1
Phase 1	**Key19**	3	**e**	Phase 1
Phase 1	**Key28**	4	**i**	Phase 1
Phase 1	**Key6**	5	**n**	Phase 1
Phase 1	**Key29**	6	**r**	Phase 1
Phase 1	**Key7**	7	**l**	Phase 1
Phase 1	**Key16**	8	**ı**	Phase 1
Phase 1	**Key27**	9	**k**	Phase 1
Phase 1	**Key5**	10	**d**	Phase 1
Phase 1	**Key30**	11	**t**	Phase 1
Phase 1	**Key8**	12	**m**	Phase 1
Phase 2	**Key20**	13	**s**	Phase 1
Initially assigned	**Key33(Space Bar)**	14	**y**	Phase 2
Phase 2	**Key26**	15	**u**	Phase 2
Phase 2	**Key4**	16	**o**	Phase 2
Phase 2	**Key15**	17	**b**	Phase 2
Phase 2	**Key9**	18	**ü**	Phase 2
Phase 2	**Key31**	19	**ş**	Phase 2
Phase 2	**Key21**	20	**z**	Phase 2
Phase 2	**Key3**	21	**g**	Phase 2
Phase 2	**Key25**	22	**v**	Phase 2
Phase 2	**Key14**	23	**ç**	Phase 2
Phase 2	**Key10**	24	**h**	Phase 2
Phase 2	**Key32**	25	**ğ**	Phase 2
Phase 3	**Key22**	26	**c**	Phase 3
Phase 3	**Key2**	27	**ö**	Phase 3
Phase 3	**Key24**	28	**p**	Phase 3
Phase 3	**Key13**	29	**f**	Phase 3
Phase 3	**Key1**	30	**j**	Phase 3

Table B.16. Keys of Square Keyboard Model

sorted acc. to Centr. Indices

#	Key	Centralization Rating
1	Key15	2,028.63
2	Key16	2,028.63
3	Key22	2,190.03
4	Key21	2,190.03
5	Key9	2,190.03
6	Key10	2,190.03
7	Key14	2,318.60
8	Key17	2,318.60
9	Key23	2,472.02
10	Key20	2,472.02
11	Key8	2,472.02
12	Key11	2,472.02
13	Key4	2,661.32
14	Key27	2,661.32
15	Key28	2,661.32
16	Key3	2,661.32
17	Key13	2,883.97
18	Key18	2,883.97
19	Key2	2,920.83
20	Key29	2,920.83
21	Key5	2,920.83
22	Key26	2,920.83
23	Key19	3,022.02
24	Key24	3,022.02
25	Key12	3,022.02
26	Key7	3,022.02
27	Key30	3,425.88
28	Key1	3,425,88
29	Key25	3,425,88
30	Key6	3,425,88

Table B.17. Letters and keys in which phase to be assigned on square keyboard for *n*=8

Phase to be Optimized in	Keys	#	Letters	**Phase to be Optimized in**
Phase 1	Key15	1	blank	Phase 1
Phase 1	Key16	2	a	Phase 1
Phase 1	Key22	3	e	Phase 1
Phase 1	Key21	4	i	Phase 1
Phase 1	Key9	5	n	Phase 1
Phase 1	Key10	6	r	Phase 1
Phase 1	Key14	7	l	Phase 1
Phase 1	Key17	8	ı	Phase 1
Phase 2	Key23	9	k	Phase 2
Phase 2	Key20	10	d	Phase 2
Phase 2	Key8	11	t	Phase 2
Phase 2	Key11	12	m	Phase 2
Phase 2	Key4	13	s	Phase 2
Phase 2	Key27	14	y	Phase 2
Phase 2	Key28	15	u	Phase 2
Phase 2	Key3	16	o	Phase 2
Phase 3	Key13	17	b	Phase 3
Phase 3	Key18	18	ü	Phase 3
Phase 3	Key2	19	ş	Phase 3
Phase 3	Key29	20	z	Phase 3
Phase 3	Key5	21	g	Phase 3
Phase 3	Key26	22	v	Phase 3
Phase 3	Key19	23	ç	Phase 3
Phase 3	Key24	24	h	Phase 3
Phase 4	Key12	25	ğ	Phase 4
Phase 4	Key7	26	c	Phase 4
Phase 4	Key30	27	ö	Phase 4
Phase 4	Key1	28	p	Phase 4
Phase 4	Key25	29	f	Phase 4
Phase 4	Key6	30	j	Phase 4

Table B.18. Letters and keys in which phase to be assigned on square keyboard for *n*=9

Phase to be optimized in	**Keys**	**#**	**Letters**	**Phase to be optimized in**
Phase 1	Key15	1	blank	Phase 1
Phase 1	Key16	2	a	Phase 1
Phase 1	Key22	3	e	Phase 1
Phase 1	Key21	4	i	Phase 1
Phase 1	Key9	5	n	Phase 1
Phase 1	Key10	6	r	Phase 1
Phase 1	Key14	7	l	Phase 1
Phase 1	Key17	8	ı	Phase 1
Phase 1	Key23	9	k	Phase 1
Phase 2	Key20	10	d	Phase 2
Phase 2	Key8	11	t	Phase 2
Phase 2	Key11	12	m	Phase 2
Phase 2	Key4	13	s	Phase 2
Phase 2	Key27	14	y	Phase 2
Phase 2	Key28	15	u	Phase 2
Phase 2	Key3	16	o	Phase 2
Phase 2	Key13	17	b	Phase 2
Phase 2	Key18	18	ü	Phase 2
Phase 3	Key2	19	ş	Phase 3
Phase 3	Key29	20	z	Phase 3
Phase 3	Key5	21	g	Phase 3
Phase 3	Key26	22	v	Phase 3
Phase 3	Key19	23	ç	Phase 3
Phase 3	Key24	24	h	Phase 3
Phase 3	Key12	25	ğ	Phase 3
Phase 3	Key7	26	c	Phase 3
Phase 3	Key30	27	ö	Phase 3
Phase 4	Key1	28	p	Phase 4
Phase 4	Key25	29	f	Phase 4
Phase 4	Key6	30	j	Phase 4

Table B.19. Letters and keys in which phase to be assigned on square keyboard for *n*=10

	Keys	#	Letters	
Phase 1	Key15	1	blank	Phase 1
Phase 1	Key16	2	a	Phase 1
Phase 1	Key22	3	e	Phase 1
Phase 1	Key21	4	i	Phase 1
Phase 1	Key9	5	n	Phase 1
Phase 1	Key10	6	r	Phase 1
Phase 1	Key14	7	l	Phase 1
Phase 1	Key17	8	ı	Phase 1
Phase 1	Key23	9	k	Phase 1
Phase 1	Key20	10	d	Phase 1
Phase 2	Key8	11	t	Phase 2
Phase 2	Key11	12	m	Phase 2
Phase 2	Key4	13	s	Phase 2
Phase 2	Key27	14	y	Phase 2
Phase 2	Key28	15	u	Phase 2
Phase 2	Key3	16	o	Phase 2
Phase 2	Key13	17	b	Phase 2
Phase 2	Key18	18	ü	Phase 2
Phase 2	Key2	19	ş	Phase 2
Phase 2	Key29	20	z	Phase 2
Phase 3	Key5	21	g	Phase 3
Phase 3	Key26	22	v	Phase 3
Phase 3	Key19	23	ç	Phase 3
Phase 3	Key24	24	h	Phase 3
Phase 3	Key12	25	ğ	Phase 3
Phase 3	Key7	26	c	Phase 3
Phase 3	Key30	27	ö	Phase 3
Phase 3	Key1	28	p	Phase 3
Phase 3	Key25	29	f	Phase 3
Phase 3	Key6	30	j	Phase 3

Table B.20. Letters and keys in which phase to be assigned on square keyboard for *n*=11

To be Optimized in	Keys	#	Letters	To be Optimized in
Phase 1	Key15	1	blank	Phase 1
Phase 1	Key16	2	a	Phase 1
Phase 1	Key22	3	e	Phase 1
Phase 1	Key21	4	i	Phase 1
Phase 1	Key9	5	n	Phase 1
Phase 1	Key10	6	r	Phase 1
Phase 1	Key14	7	l	Phase 1
Phase 1	Key17	8	ı	Phase 1
Phase 1	Key23	9	k	Phase 1
Phase 1	Key20	10	d	Phase 1
Phase 1	Key8	11	t	Phase 1
Phase 2	Key11	12	m	Phase 2
Phase 2	Key4	13	s	Phase 2
Phase 2	Key27	14	y	Phase 2
Phase 2	Key28	15	u	Phase 2
Phase 2	Key3	16	o	Phase 2
Phase 2	Key13	17	b	Phase 2
Phase 2	Key18	18	ü	Phase 2
Phase 2	Key2	19	ş	Phase 2
Phase 2	Key29	20	z	Phase 2
Phase 2	Key5	21	g	Phase 2
Phase 2	Key26	22	v	Phase 2
Phase 3	Key19	23	ç	Phase 3
Phase 3	Key24	24	h	Phase 3
Phase 3	Key12	25	ğ	Phase 3
Phase 3	Key7	26	c	Phase 3
Phase 3	Key30	27	ö	Phase 3
Phase 3	Key1	28	p	Phase 3
Phase 3	Key25	29	f	Phase 3
Phase 3	Key6	30	j	Phase 3

Table B.21. Letters and keys in which phase to be assigned on square keyboard for *n*=12

to be Optimized in	Keys	#	Letters	to be Optimized in
Phase 1	Key15	1	blank	Phase 1
Phase 1	Key16	2	a	Phase 1
Phase 1	Key22	3	e	Phase 1
Phase 1	Key21	4	i	Phase 1
Phase 1	Key9	5	n	Phase 1
Phase 1	Key10	6	r	Phase 1
Phase 1	Key14	7	l	Phase 1
Phase 1	Key17	8	ı	Phase 1
Phase 1	Key23	9	k	Phase 1
Phase 1	Key20	10	d	Phase 1
Phase 1	Key8	11	t	Phase 1
Phase 1	Key11	12	m	Phase 1
Phase 2	Key4	13	s	Phase 2
Phase 2	Key27	14	y	Phase 2
Phase 2	Key28	15	u	Phase 2
Phase 2	Key3	16	o	Phase 2
Phase 2	Key13	17	b	Phase 2
Phase 2	Key18	18	ü	Phase 2
Phase 2	Key2	19	ş	Phase 2
Phase 2	Key29	20	z	Phase 2
Phase 2	Key5	21	g	Phase 2
Phase 2	Key26	22	v	Phase 2
Phase 2	Key19	23	ç	Phase 2
Phase 2	Key24	24	h	Phase 2
Phase 3	Key12	25	ğ	Phase 3
Phase 3	Key7	26	c	Phase 3
Phase 3	Key30	27	ö	Phase 3
Phase 3	Key1	28	p	Phase 3
Phase 3	Key25	29	f	Phase 3
Phase 3	Key6	30	j	Phase 3

Table B.22. QAP Model of Phase 2 for *n* = 8

$$\min \sum_{i=1}^{17} \sum_{j=1}^{17} \sum_{k=1}^{17} \sum_{p=1}^{17} f_{i,j} * d_{k,p} * x_{i,k} * x_{j,p}$$

$$\sum_{i=1}^{17} x_{i,j} = 1 \qquad 1 \leq j \leq 17,$$

$$\sum_{j=1}^{17} x_{i,j} = 1 \qquad 1 \leq i \leq 17,$$

$x_{3,4} = 1$ *(Key 6 assigned and fixed to e)*

$x_{4,8} = 1$ *(Key 7 assigned and fixed to l)*

$x_{7,5} = 1$ *(Key 16 assigned and fixed to ı)*

$x_{8,12} = 1$ *(Key 17 assigned and fixed to r)*

$x_{9,1} = 1$ *(Key 18 assigned and fixed to a)*

$x_{10,7} = 1$ *(Key 19 assigned and fixed to k)*

$x_{14,10} = 1$ *(Key 28 assigned and fixed to n)*

$x_{15,6} = 1$ *(Key 29 assigned and fixed to i)*

$x_{17,17} = 1$ *(Key 33 assigned and fixed to blank char.)*

$$x_{i,j} \in \{0,1\} \qquad 1 \leq i, j \leq 17$$

Table B.23. QAP Model of Phase 3 for *n* = 8

$$\min \sum_{i=1}^{25} \sum_{j=1}^{25} \sum_{k=1}^{25} \sum_{p=1}^{25} f_{i,j} * d_{k,p} * x_{i,k} * x_{j,p}$$

$$\sum_{i=1}^{25} x_{i,j} = 1 \qquad 1 \le j \le 25,$$

$$\sum_{j=1}^{25} x_{i,j} = 1 \qquad 1 \le i \le 25,$$

$x_{2,17} = 1$ *(Key 4 assigned and fixed to s)*

$x_{3,19} = 1$ *(Key 5 assigned and fixed to t)*

$x_{4,5} = 1$ *(Key 6 assigned and fixed to e)*

$x_{5,12} = 1$ *(Key 7 assigned and fixed to l)*

$x_{6,13} = 1$ *(Key 8 assigned and fixed to m)*

$x_{10,15} = 1$ *(Key 15 assigned and fixed to o)*

$x_{11,9} = 1$ *(Key 16 assigned and fixed to ı)*

$x_{12,16} = 1$ *(Key 17 assigned and fixed to r)*

$x_{13,1} = 1$ *(Key 18 assigned and fixed to a)*

$x_{14,1} = 1$ *(Key 19 assigned and fixed to k)*

$x_{15,20} = 1$ *(Key 20 assigned and fixed to u)*

$x_{18,23} = 1$ *(Key 26 assigned and fixed to y)*

$x_{19,4} = 1$ *(Key 27 assigned and fixed to d)*

$x_{20,14} = 1$ *(Key 28 assigned and fixed to n)*

$x_{21,10} = 1$ *(Key 29 assigned and fixed to i)*

$x_{22,2} = 1$ *(Key 30 assigned and fixed to b)*

$$x_{i,j} \in \{0,1\} \qquad 1 \le i,j \le 25$$

Table B.24. QAP Model of Phase 4 for n = 8

$$\min \sum_{i=1}^{30} \sum_{j=1}^{30} \sum_{k=1}^{30} \sum_{p=1}^{30} f_{i,j} * d_{k,p} * x_{i,k} * x_{j,p}$$

$$\sum_{i=1}^{30} x_{i,j} = 1 \qquad 1 \leq j \leq 30,$$

$$\sum_{j=1}^{30} x_{i,j} = 1 \qquad 1 \leq i \leq 30,$$

$x_{3,27} = 1$ *(Key 3 assigned and fixed to v)*

$x_{4,22} = 1$ *(Key 4 assigned and fixed to s)*

$x_{5,24} = 1$ *(Key 5 assigned and fixed to t)*

$x_{6,6} = 1$ *(Key 6 assigned and fixed to e)*

$x_{7,15} = 1$ *(Key 7 assigned and fixed to l)*

$x_{8,16} = 1$ *(Key 8 assigned and fixed to m)*

$x_{9,23} = 1$ *(Key 9 assigned and fixed to ş)*

$x_{10,29} = 1$ *(Key 10 assigned and fixed to z)*

$x_{12,9} = 1$ *(Key 14 assigned and fixed to ğ)*

$x_{13,18} = 1$ *(Key 15 assigned and fixed to o)*

$x_{14,11} = 1$ *(Key 16 assigned and fixed to ı)*

$x_{15,21} = 1$ *(Key 17 assigned and fixed to r)*

$x_{16,1} = 1$ *(Key 18 assigned and fixed to a)*

$x_{17,14} = 1$ *(Key 19 assigned and fixed to k)*

$x_{18,25} = 1$ *(Key 20 assigned and fixed to u)*

$x_{19,26} = 1$ *(Key 21 assigned and fixed to ü)*

$x_{22,4} = 1$ *(Key 25 assigned and fixed to ç)*

$x_{23,28} = 1$ *(Key 26 assigned and fixed to y)*

$x_{24,5} = 1$ *(Key 27 assigned and fixed to d)*

$x_{25,17} = 1$ *(Key 28 assigned and fixed to n)*

$x_{26,12} = 1$ *(Key 29 assigned and fixed to i)*

$x_{27,2} = 1$ *(Key 30 assigned and fixed to b)*

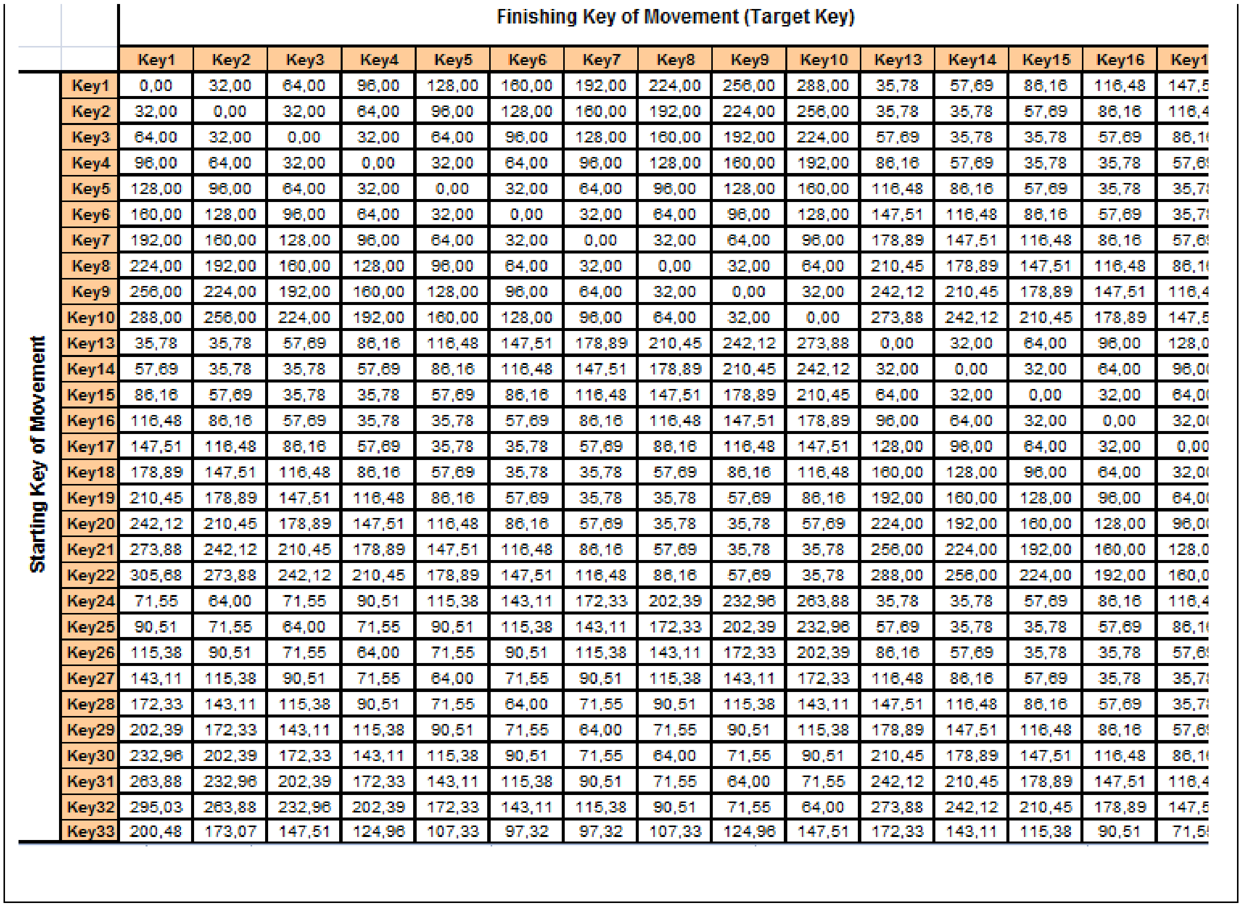
$x_{28,8} = 1$ *(Key 31 assigned and fixed to g)*

$x_{29,10} = 1$ *(Key 32 assigned and fixed to h)*

$x_{30,30} = 1$ *(Key 33 assigned and fixed to blank char.)*

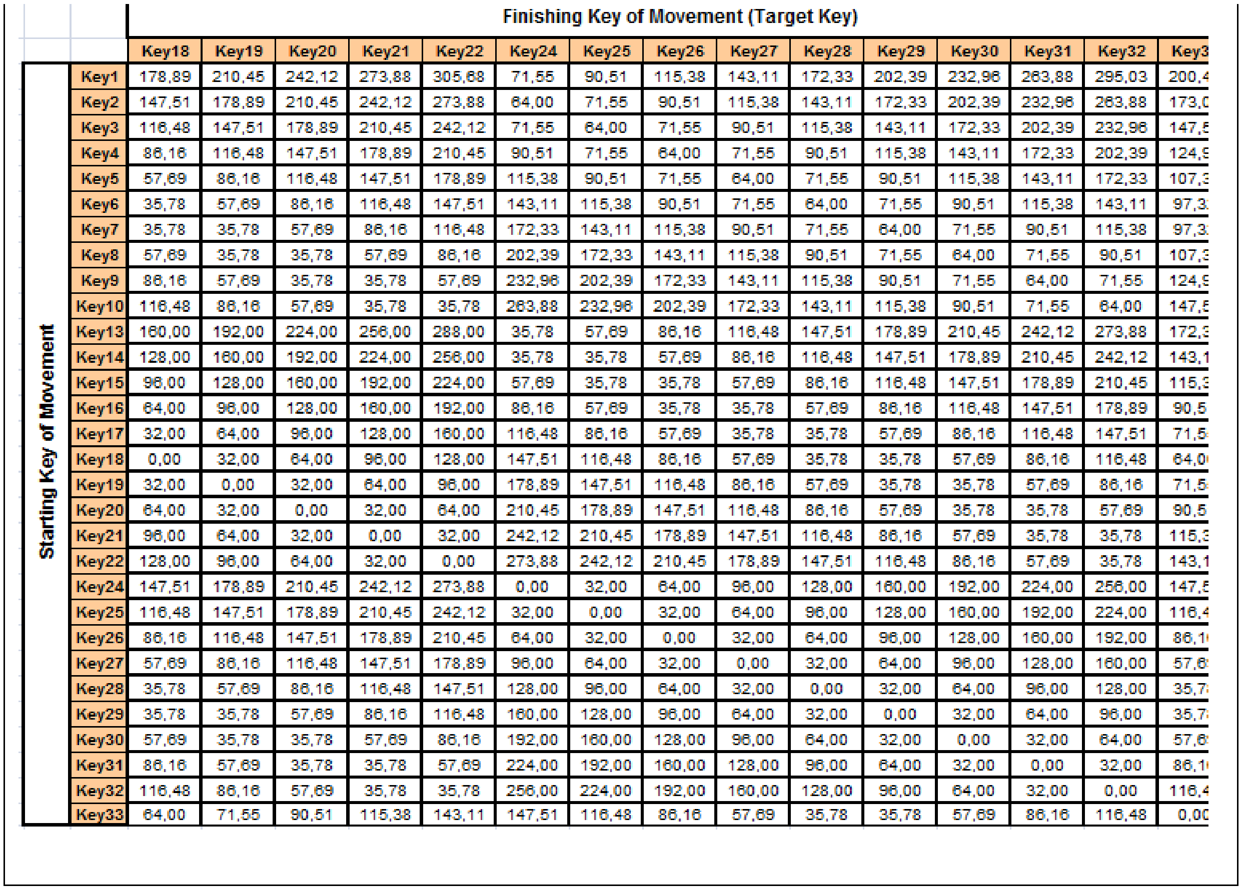
APPENDIX C: PARAMETERS FOR DISTANCE MATRICES

Table C.1. Distance matrix for longitudinal layout, after foreign letters are assigned



	Finishing Key of Movement (Target Key)													
Starting Key of Movement	Key1	Key2	Key3	Key4	Key5	Key6	Key7	Key8	Key9	Key10	Key13	Key14	Key15	Key16	Key1
Key1	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	256,00	288,00	35,78	57,69	86,16	116,48	147,5
Key2	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	256,00	35,78	35,78	57,69	86,16	116,4
Key3	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	57,69	35,78	35,78	57,69	86,1
Key4	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	86,16	57,69	35,78	35,78	57,6
Key5	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	116,48	86,16	57,69	35,78	35,7
Key6	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	147,51	116,48	86,16	57,69	35,7
Key7	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	178,89	147,51	116,48	86,16	57,6
Key8	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	210,45	178,89	147,51	116,48	86,1
Key9	256,00	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	242,12	210,45	178,89	147,51	116,4
Key10	288,00	256,00	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	273,88	242,12	210,45	178,89	147,5
Key13	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	242,12	273,88	0,00	32,00	64,00	96,00	128,0
Key14	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	242,12	32,00	0,00	32,00	64,00	96,0
Key15	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	64,00	32,00	0,00	32,00	64,0
Key16	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	96,00	64,00	32,00	0,00	32,0
Key17	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	128,00	96,00	64,00	32,00	0,00
Key18	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	160,00	128,00	96,00	64,00	32,0
Key19	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	192,00	160,00	128,00	96,00	64,0
Key20	242,12	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	224,00	192,00	160,00	128,00	96,0
Key21	273,88	242,12	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	256,00	224,00	192,00	160,00	128,0
Key22	305,68	273,88	242,12	210,45	178,89	147,51	116,48	86,16	57,69	35,78	288,00	256,00	224,00	192,00	160,0
Key24	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	232,96	263,88	35,78	35,78	57,69	86,16	116,4
Key25	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	232,96	57,69	35,78	35,78	57,69	86,1
Key26	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	86,16	57,69	35,78	35,78	57,6
Key27	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	116,48	86,16	57,69	35,78	35,7
Key28	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	147,51	116,48	86,16	57,69	35,7
Key29	202,39	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	178,89	147,51	116,48	86,16	57,6
Key30	232,96	202,39	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	210,45	178,89	147,51	116,48	86,1
Key31	263,88	232,96	202,39	172,33	143,11	115,38	90,51	71,55	64,00	71,55	242,12	210,45	178,89	147,51	116,4
Key32	295,03	263,88	232,96	202,39	172,33	143,11	115,38	90,51	71,55	64,00	273,88	242,12	210,45	178,89	147,5
Key33	200,48	173,07	147,51	124,96	107,33	97,32	97,32	107,33	124,96	147,51	172,33	143,11	115,38	90,51	71,5

Table C.1. (*cont'd)* Distance matrix for longitudinal layout, after foreign letters are assigned



		Finishing Key of Movement (Target Key)														
		Key18	Key19	Key20	Key21	Key22	Key24	Key25	Key26	Key27	Key28	Key29	Key30	Key31	Key32	Key3
Starting Key of Movement	Key1	178,89	210,45	242,12	273,88	305,68	71,55	90,51	115,38	143,11	172,33	202,39	232,96	263,88	295,03	200,4
	Key2	147,51	178,89	210,45	242,12	273,88	64,00	71,55	90,51	115,38	143,11	172,33	202,39	232,96	263,88	173,0
	Key3	116,48	147,51	178,89	210,45	242,12	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	232,96	147,5
	Key4	86,16	116,48	147,51	178,89	210,45	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	202,39	124,9
	Key5	57,69	86,16	116,48	147,51	178,89	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	172,33	107,3
	Key6	35,78	57,69	86,16	116,48	147,51	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	143,11	97,3
	Key7	35,78	35,78	57,69	86,16	116,48	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	115,38	97,3
	Key8	57,69	35,78	35,78	57,69	86,16	202,39	172,33	143,11	115,38	90,51	71,55	64,00	71,55	90,51	107,3
	Key9	86,16	57,69	35,78	35,78	57,69	232,96	202,39	172,33	143,11	115,38	90,51	71,55	64,00	71,55	124,9
	Key10	116,48	86,16	57,69	35,78	35,78	263,88	232,96	202,39	172,33	143,11	115,38	90,51	71,55	64,00	147,5
	Key13	160,00	192,00	224,00	256,00	288,00	35,78	57,69	86,16	116,48	147,51	178,89	210,45	242,12	273,88	172,3
	Key14	128,00	160,00	192,00	224,00	256,00	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	242,12	143,1
	Key15	96,00	128,00	160,00	192,00	224,00	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	210,45	115,3
	Key16	64,00	96,00	128,00	160,00	192,00	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	178,89	90,5
	Key17	32,00	64,00	96,00	128,00	160,00	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	147,51	71,5
	Key18	0,00	32,00	64,00	96,00	128,00	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	64,0
	Key19	32,00	0,00	32,00	64,00	96,00	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	71,5
	Key20	64,00	32,00	0,00	32,00	64,00	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	57,69	90,5
	Key21	96,00	64,00	32,00	0,00	32,00	242,12	210,45	178,89	147,51	116,48	86,16	57,69	35,78	35,78	115,3
	Key22	128,00	96,00	64,00	32,00	0,00	273,88	242,12	210,45	178,89	147,51	116,48	86,16	57,69	35,78	143,1
	Key24	147,51	178,89	210,45	242,12	273,88	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	256,00	147,5
	Key25	116,48	147,51	178,89	210,45	242,12	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	224,00	116,4
	Key26	86,16	116,48	147,51	178,89	210,45	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	192,00	86,1
	Key27	57,69	86,16	116,48	147,51	178,89	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	160,00	57,6
	Key28	35,78	57,69	86,16	116,48	147,51	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	128,00	35,7
	Key29	35,78	35,78	57,69	86,16	116,48	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	96,00	35,7
	Key30	57,69	35,78	35,78	57,69	86,16	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	64,00	57,6
	Key31	86,16	57,69	35,78	35,78	57,69	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	32,00	86,1
	Key32	116,48	86,16	57,69	35,78	35,78	256,00	224,00	192,00	160,00	128,00	96,00	64,00	32,00	0,00	116,4
	Key33	64,00	71,55	90,51	115,38	143,11	147,51	116,48	86,16	57,69	35,78	35,78	57,69	86,16	116,48	0,0

Table C.2. Coordinates of midpoints of keys for longitudinal shape model

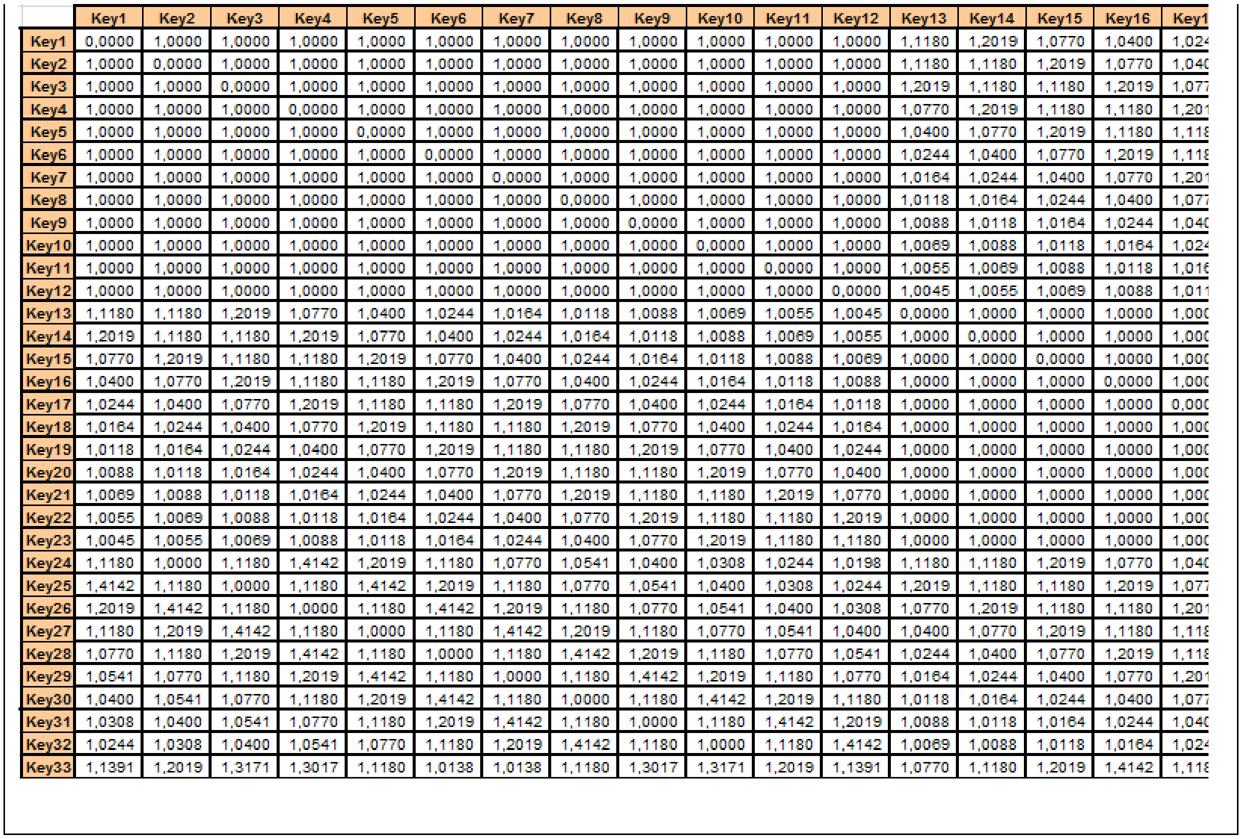
Key	Mid Point's	
	x Coordinate	y Coordinate
Key1	65	49
Key2	97	49
Key3	129	49
Key4	161	49
Key5	193	49
Key6	225	49
Key7	257	49
Key8	289	49
Key9	321	49
Key10	353	49
Key11	385	49
Key12	417	49
Key13	81	81
Key14	113	81
Key15	145	81
Key16	177	81
Key17	209	81
Key18	241	81
Key19	273	81
Key20	305	81
Key21	337	81
Key22	369	81
Key23	401	81
Key24	97	113
Key25	129	113
Key26	161	113
Key27	193	113
Key28	225	113
Key29	257	113
Key30	289	113
Key31	321	113
Key32	353	113
Key33	241	145

Table C.3. Coordinates of midpoints of keys for square shape model

	Mid Point's	
	x Coordinate	y Coordinate
Key1	17	17
Key2	49	17
Key3	81	17
Key4	113	17
Key5	145	17
Key6	177	17
Key7	17	49
Key8	49	49
Key9	81	49
Key10	113	49
Key11	145	49
Key12	177	49
Key13	17	81
Key14	49	81
Key15	81	81
Key16	113	81
Key17	145	81
Key18	177	81
Key19	17	113
Key20	49	113
Key21	81	113
Key22	113	113
Key23	145	113
Key24	177	113
Key25	17	145
Key26	49	145
Key27	81	145
Key28	113	145
Key29	145	145
Key30	177	145

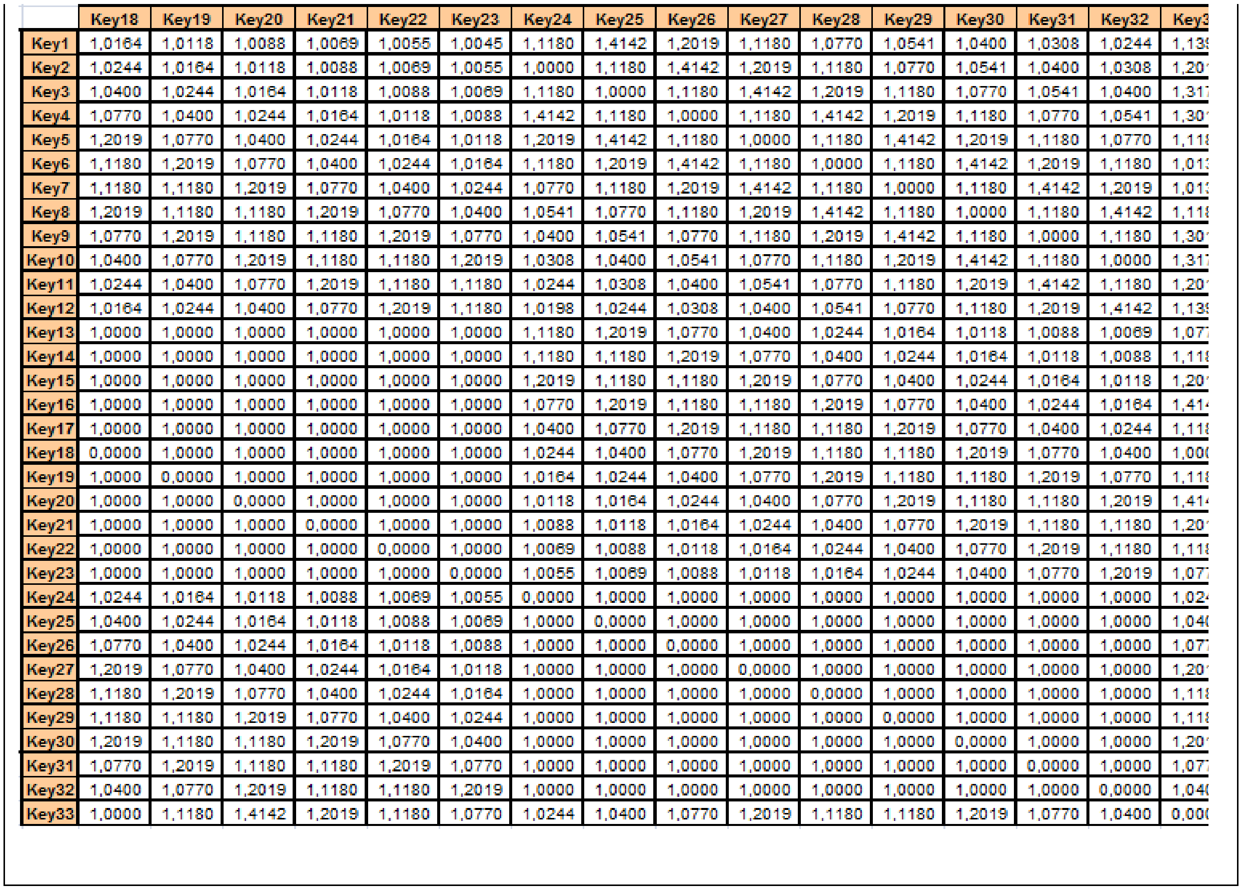
APPENDIX D: FITTS’ LAW PARAMETERS

Table D.1. *Wj* table for longitudinal keyboard keys



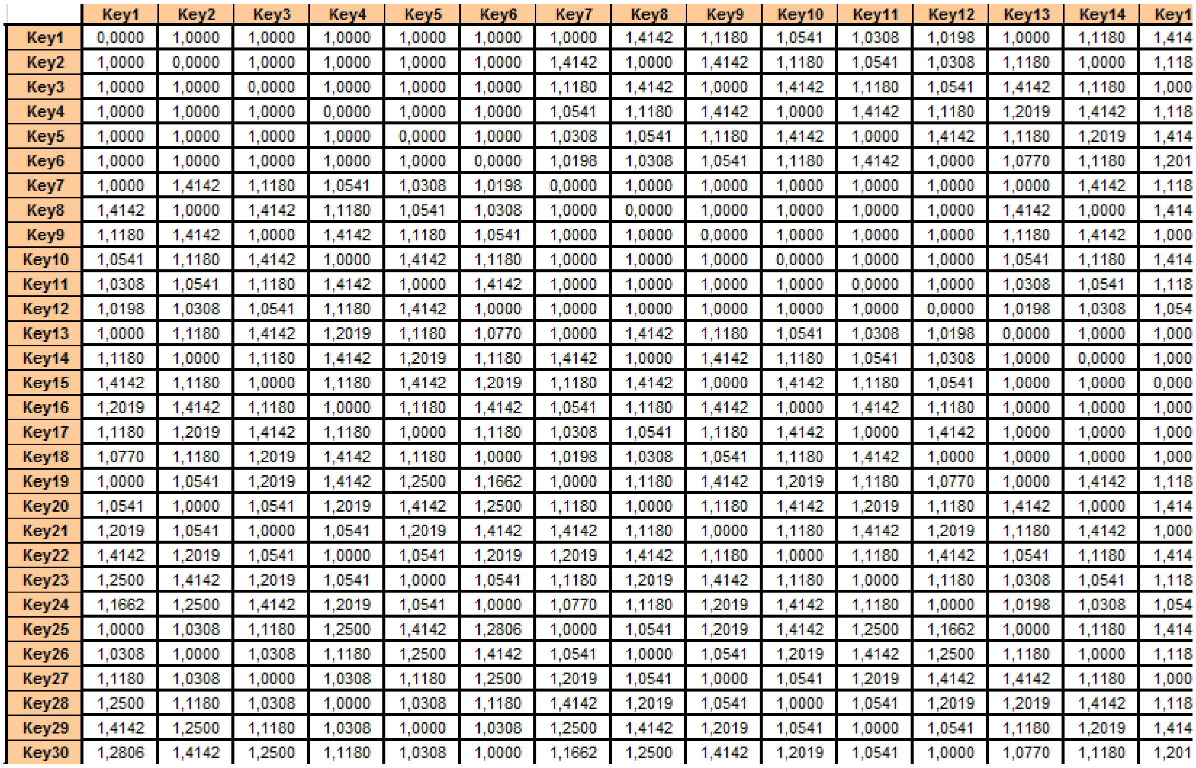
	Key1	Key2	Key3	Key4	Key5	Key6	Key7	Key8	Key9	Key10	Key11	Key12	Key13	Key14	Key15	Key16	Key1
Key1	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,1180	1,2019	1,0770	1,0400	1,024
Key2	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,1180	1,1180	1,2019	1,0770	1,040
Key3	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,2019	1,1180	1,1180	1,2019	1,077
Key4	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0770	1,2019	1,1180	1,1180	1,201
Key5	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0400	1,0770	1,2019	1,1180	1,118
Key6	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0244	1,0400	1,0770	1,2019	1,118
Key7	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0164	1,0244	1,0400	1,0770	1,201
Key8	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,0118	1,0164	1,0244	1,0400	1,077
Key9	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,0088	1,0118	1,0164	1,0244	1,040
Key10	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0069	1,0088	1,0118	1,0164	1,024
Key11	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0055	1,0069	1,0088	1,0118	1,016
Key12	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0045	1,0055	1,0069	1,0088	1,011
Key13	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,0118	1,0088	1,0069	1,0055	1,0045	0,0000	1,0000	1,0000	1,0000	1,000
Key14	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,0118	1,0088	1,0069	1,0055	1,0000	0,0000	1,0000	1,0000	1,000
Key15	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,0118	1,0088	1,0069	1,0000	1,0000	0,0000	1,0000	1,000
Key16	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,0118	1,0088	1,0000	1,0000	1,0000	0,0000	1,000
Key17	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,0118	1,0000	1,0000	1,0000	1,0000	0,000
Key18	1,0164	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,0000	1,0000	1,0000	1,0000	1,000
Key19	1,0118	1,0164	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0000	1,0000	1,0000	1,0000	1,000
Key20	1,0088	1,0118	1,0164	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0000	1,0000	1,0000	1,0000	1,000
Key21	1,0069	1,0088	1,0118	1,0164	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0000	1,0000	1,0000	1,0000	1,000
Key22	1,0055	1,0069	1,0088	1,0118	1,0164	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0000	1,0000	1,0000	1,0000	1,000
Key23	1,0045	1,0055	1,0069	1,0088	1,0118	1,0164	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,0000	1,0000	1,0000	1,0000	1,000
Key24	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0541	1,0400	1,0308	1,0244	1,0198	1,1180	1,1180	1,2019	1,0770	1,040
Key25	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0541	1,0400	1,0308	1,0244	1,2019	1,1180	1,1180	1,2019	1,077
Key26	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0541	1,0400	1,0308	1,0770	1,2019	1,1180	1,1180	1,201
Key27	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0541	1,0400	1,0400	1,0770	1,2019	1,1180	1,118
Key28	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0541	1,0244	1,0400	1,0770	1,2019	1,118
Key29	1,0541	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0164	1,0244	1,0400	1,0770	1,201
Key30	1,0400	1,0541	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,0118	1,0164	1,0244	1,0400	1,077
Key31	1,0308	1,0400	1,0541	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,0088	1,0118	1,0164	1,0244	1,040
Key32	1,0244	1,0308	1,0400	1,0541	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,0069	1,0088	1,0118	1,0164	1,024
Key33	1,1391	1,2019	1,3171	1,3017	1,1180	1,0138	1,0138	1,1180	1,3017	1,3171	1,2019	1,1391	1,0770	1,1180	1,2019	1,4142	1,118

Table D.1. (*cont'd) Wj* table for longitudinal keyboard keys



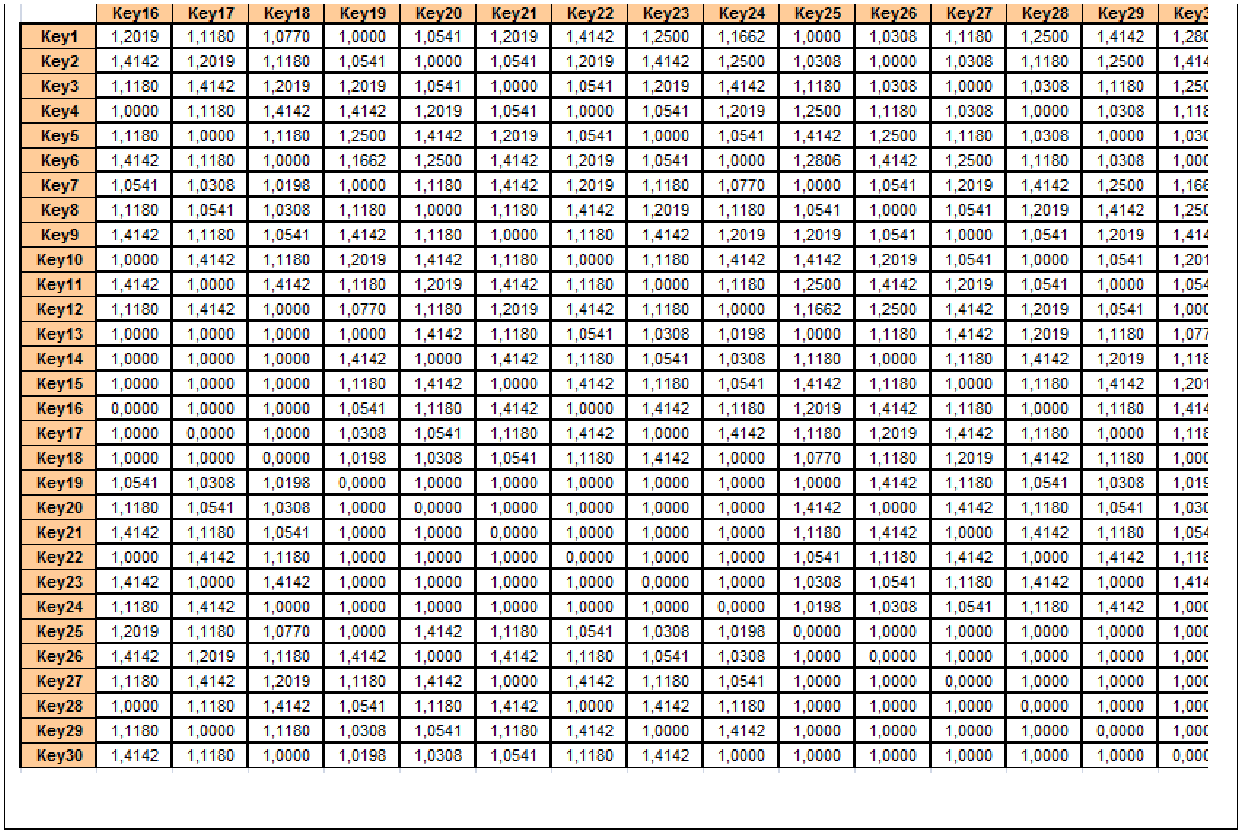
	Key18	Key19	Key20	Key21	Key22	Key23	Key24	Key25	Key26	Key27	Key28	Key29	Key30	Key31	Key32	Key3
Key1	1,0164	1,0118	1,0088	1,0069	1,0055	1,0045	1,1180	1,4142	1,2019	1,1180	1,0770	1,0541	1,0400	1,0308	1,0244	1,13
Key2	1,0244	1,0164	1,0118	1,0088	1,0069	1,0055	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0541	1,0400	1,0308	1,20
Key3	1,0400	1,0244	1,0164	1,0118	1,0088	1,0069	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0541	1,0400	1,31
Key4	1,0770	1,0400	1,0244	1,0164	1,0118	1,0088	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0541	1,30
Key5	1,2019	1,0770	1,0400	1,0244	1,0164	1,0118	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,11
Key6	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,01
Key7	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,01
Key8	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0541	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,11
Key9	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0541	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,30
Key10	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0308	1,0400	1,0541	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,31
Key11	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,0244	1,0308	1,0400	1,0541	1,0770	1,1180	1,2019	1,4142	1,1180	1,20
Key12	1,0164	1,0244	1,0400	1,0770	1,2019	1,1180	1,0198	1,0244	1,0308	1,0400	1,0541	1,0770	1,1180	1,2019	1,4142	1,13
Key13	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,0118	1,0088	1,0069	1,07
Key14	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,0118	1,0088	1,11
Key15	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,0118	1,20
Key16	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,41
Key17	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,11
Key18	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,00
Key19	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,0164	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,11
Key20	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,0118	1,0164	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,41
Key21	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0088	1,0118	1,0164	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,20
Key22	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0069	1,0088	1,0118	1,0164	1,0244	1,0400	1,0770	1,2019	1,1180	1,11
Key23	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0055	1,0069	1,0088	1,0118	1,0164	1,0244	1,0400	1,0770	1,2019	1,07
Key24	1,0244	1,0164	1,0118	1,0088	1,0069	1,0055	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,02
Key25	1,0400	1,0244	1,0164	1,0118	1,0088	1,0069	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,04
Key26	1,0770	1,0400	1,0244	1,0164	1,0118	1,0088	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,07
Key27	1,2019	1,0770	1,0400	1,0244	1,0164	1,0118	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,20
Key28	1,1180	1,2019	1,0770	1,0400	1,0244	1,0164	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,11
Key29	1,1180	1,1180	1,2019	1,0770	1,0400	1,0244	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,11
Key30	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,20
Key31	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,07
Key32	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,04
Key33	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0244	1,0400	1,0770	1,2019	1,1180	1,1180	1,2019	1,0770	1,0400	0,00

Table D.2. *Wj* table for square keyboard keys



	Key1	Key2	Key3	Key4	Key5	Key6	Key7	Key8	Key9	Key10	Key11	Key12	Key13	Key14	Key1
Key1	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,4142	1,1180	1,0541	1,0308	1,0198	1,0000	1,1180	1,414
Key2	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,4142	1,0000	1,4142	1,1180	1,0541	1,0308	1,1180	1,0000	1,118
Key3	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,1180	1,4142	1,0000	1,4142	1,1180	1,0541	1,4142	1,1180	1,000
Key4	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0541	1,1180	1,4142	1,0000	1,4142	1,1180	1,2019	1,4142	1,118
Key5	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0308	1,0541	1,1180	1,4142	1,0000	1,4142	1,1180	1,2019	1,414
Key6	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0198	1,0308	1,0541	1,1180	1,4142	1,0000	1,0770	1,1180	1,201
Key7	1,0000	1,4142	1,1180	1,0541	1,0308	1,0198	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,4142	1,118
Key8	1,4142	1,0000	1,4142	1,1180	1,0541	1,0308	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,4142	1,0000	1,414
Key9	1,1180	1,4142	1,0000	1,4142	1,1180	1,0541	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,1180	1,4142	1,000
Key10	1,0541	1,1180	1,4142	1,0000	1,4142	1,1180	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0541	1,1180	1,414
Key11	1,0308	1,0541	1,1180	1,4142	1,0000	1,4142	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0308	1,0541	1,118
Key12	1,0198	1,0308	1,0541	1,1180	1,4142	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0198	1,0308	1,054
Key13	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0000	1,4142	1,1180	1,0541	1,0308	1,0198	0,0000	1,0000	1,000
Key14	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,4142	1,0000	1,4142	1,1180	1,0541	1,0308	1,0000	0,0000	1,000
Key15	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,4142	1,0000	1,4142	1,1180	1,0541	1,0000	1,0000	0,000
Key16	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,0541	1,1180	1,4142	1,0000	1,4142	1,1180	1,0000	1,0000	1,000
Key17	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,0308	1,0541	1,1180	1,4142	1,0000	1,4142	1,0000	1,0000	1,000
Key18	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,0198	1,0308	1,0541	1,1180	1,4142	1,0000	1,0000	1,0000	1,000
Key19	1,0000	1,0541	1,2019	1,4142	1,2500	1,1662	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0000	1,4142	1,118
Key20	1,0541	1,0000	1,0541	1,2019	1,4142	1,2500	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,4142	1,0000	1,414
Key21	1,2019	1,0541	1,0000	1,0541	1,2019	1,4142	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,4142	1,000
Key22	1,4142	1,2019	1,0541	1,0000	1,0541	1,2019	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,0541	1,1180	1,414
Key23	1,2500	1,4142	1,2019	1,0541	1,0000	1,0541	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,0308	1,0541	1,118
Key24	1,1662	1,2500	1,4142	1,2019	1,0541	1,0000	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,0198	1,0308	1,054
Key25	1,0000	1,0308	1,1180	1,2500	1,4142	1,2806	1,0000	1,0541	1,2019	1,4142	1,2500	1,1662	1,0000	1,1180	1,414
Key26	1,0308	1,0000	1,0308	1,1180	1,2500	1,4142	1,0541	1,0000	1,0541	1,2019	1,4142	1,2500	1,1180	1,0000	1,118
Key27	1,1180	1,0308	1,0000	1,0308	1,1180	1,2500	1,2019	1,0541	1,0000	1,0541	1,2019	1,4142	1,4142	1,1180	1,000
Key28	1,2500	1,1180	1,0308	1,0000	1,0308	1,1180	1,4142	1,2019	1,0541	1,0000	1,0541	1,2019	1,2019	1,4142	1,118
Key29	1,4142	1,2500	1,1180	1,0308	1,0000	1,0308	1,2500	1,4142	1,2019	1,0541	1,0000	1,0541	1,1180	1,2019	1,414
Key30	1,2806	1,4142	1,2500	1,1180	1,0308	1,0000	1,1662	1,2500	1,4142	1,2019	1,0541	1,0000	1,0770	1,1180	1,201

Table D.2. (*cont'd) Wj* table for square keyboard keys



	Key16	Key17	Key18	Key19	Key20	Key21	Key22	Key23	Key24	Key25	Key26	Key27	Key28	Key29	Key3
Key1	1,2019	1,1180	1,0770	1,0000	1,0541	1,2019	1,4142	1,2500	1,1662	1,0000	1,0308	1,1180	1,2500	1,4142	1,280
Key2	1,4142	1,2019	1,1180	1,0541	1,0000	1,0541	1,2019	1,4142	1,2500	1,0308	1,0000	1,0308	1,1180	1,2500	1,414
Key3	1,1180	1,4142	1,2019	1,2019	1,0541	1,0000	1,0541	1,2019	1,4142	1,1180	1,0308	1,0000	1,0308	1,1180	1,250
Key4	1,0000	1,1180	1,4142	1,4142	1,2019	1,0541	1,0000	1,0541	1,2019	1,2500	1,1180	1,0308	1,0000	1,0308	1,118
Key5	1,1180	1,0000	1,1180	1,2500	1,4142	1,2019	1,0541	1,0000	1,0541	1,4142	1,2500	1,1180	1,0308	1,0000	1,030
Key6	1,4142	1,1180	1,0000	1,1662	1,2500	1,4142	1,2019	1,0541	1,0000	1,2806	1,4142	1,2500	1,1180	1,0308	1,000
Key7	1,0541	1,0308	1,0198	1,0000	1,1180	1,4142	1,2019	1,1180	1,0770	1,0000	1,0541	1,2019	1,4142	1,2500	1,166
Key8	1,1180	1,0541	1,0308	1,1180	1,0000	1,1180	1,4142	1,2019	1,1180	1,0541	1,0000	1,0541	1,2019	1,4142	1,250
Key9	1,4142	1,1180	1,0541	1,4142	1,1180	1,0000	1,1180	1,4142	1,2019	1,2019	1,0541	1,0000	1,0541	1,2019	1,414
Key10	1,0000	1,4142	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,4142	1,4142	1,2019	1,0541	1,0000	1,0541	1,201
Key11	1,4142	1,0000	1,4142	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,2500	1,4142	1,2019	1,0541	1,0000	1,054
Key12	1,1180	1,4142	1,0000	1,0770	1,1180	1,2019	1,4142	1,1180	1,0000	1,1662	1,2500	1,4142	1,2019	1,0541	1,000
Key13	1,0000	1,0000	1,0000	1,0000	1,4142	1,1180	1,0541	1,0308	1,0198	1,0000	1,1180	1,4142	1,2019	1,1180	1,077
Key14	1,0000	1,0000	1,0000	1,4142	1,0000	1,4142	1,1180	1,0541	1,0308	1,1180	1,0000	1,1180	1,4142	1,2019	1,118
Key15	1,0000	1,0000	1,0000	1,1180	1,4142	1,0000	1,4142	1,1180	1,0541	1,4142	1,1180	1,0000	1,1180	1,4142	1,201
Key16	0,0000	1,0000	1,0000	1,0541	1,1180	1,4142	1,0000	1,4142	1,1180	1,2019	1,4142	1,1180	1,0000	1,1180	1,414
Key17	1,0000	0,0000	1,0000	1,0308	1,0541	1,1180	1,4142	1,0000	1,4142	1,1180	1,2019	1,4142	1,1180	1,0000	1,118
Key18	1,0000	1,0000	0,0000	1,0198	1,0308	1,0541	1,1180	1,4142	1,0000	1,0770	1,1180	1,2019	1,4142	1,1180	1,000
Key19	1,0541	1,0308	1,0198	0,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	1,4142	1,1180	1,0541	1,0308	1,019
Key20	1,1180	1,0541	1,0308	1,0000	0,0000	1,0000	1,0000	1,0000	1,0000	1,4142	1,0000	1,4142	1,1180	1,0541	1,030
Key21	1,4142	1,1180	1,0541	1,0000	1,0000	0,0000	1,0000	1,0000	1,0000	1,1180	1,4142	1,0000	1,4142	1,1180	1,054
Key22	1,0000	1,4142	1,1180	1,0000	1,0000	1,0000	0,0000	1,0000	1,0000	1,0541	1,1180	1,4142	1,0000	1,4142	1,118
Key23	1,4142	1,0000	1,4142	1,0000	1,0000	1,0000	1,0000	0,0000	1,0000	1,0308	1,0541	1,1180	1,4142	1,0000	1,414
Key24	1,1180	1,4142	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,0000	1,0198	1,0308	1,0541	1,1180	1,4142	1,000
Key25	1,2019	1,1180	1,0770	1,0000	1,4142	1,1180	1,0541	1,0308	1,0198	0,0000	1,0000	1,0000	1,0000	1,0000	1,000
Key26	1,4142	1,2019	1,1180	1,4142	1,0000	1,4142	1,1180	1,0541	1,0308	1,0000	0,0000	1,0000	1,0000	1,0000	1,000
Key27	1,1180	1,4142	1,2019	1,1180	1,4142	1,0000	1,4142	1,1180	1,0541	1,0000	1,0000	0,0000	1,0000	1,0000	1,000
Key28	1,0000	1,1180	1,4142	1,0541	1,1180	1,4142	1,0000	1,4142	1,1180	1,0000	1,0000	1,0000	0,0000	1,0000	1,000
Key29	1,1180	1,0000	1,1180	1,0308	1,0541	1,1180	1,4142	1,0000	1,4142	1,0000	1,0000	1,0000	1,0000	0,0000	1,000
Key30	1,4142	1,1180	1,0000	1,0198	1,0308	1,0541	1,1180	1,4142	1,0000	1,0000	1,0000	1,0000	1,0000	1,0000	0,000

Table D.3. Data set for estimating the parameters of Fitts' Law

Keyboard Type	Move from	Target Key	Sample Size	Average of Measured Times (in millisecond)
Square	Key1	Key2	52	134
Longitudinal	Key15	Key27	49	179
Longitudinal	Key14	Key30	43	272
Square	Key2	Key12	39	239
Longitudinal	Key13	Key21	31	321
Square	Key4	Key3	40	137
Square	Key24	Key28	51	217
Square	Key14	Key30	47	235
Square	Key22	Key17	44	152
Longitudinal	Key6	Key15	39	203
Longitudinal	Key3	Key32	46	310
Square	Key6	Key14	35	245
Longitudinal	Key9	Key25	40	303
Longitudinal	Key13	Key10	38	354
Longitudinal	Key24	Key28	40	268
Square	Key8	Key11	41	221
Square	Key6	Key25	46	316
Longitudinal	Key22	Key26	48	290
Square	Key7	Key24	34	268
Longitudinal	Key16	Key20	37	272
Square	Key7	Key7	76	104
Longitudinal	Key3	Key3	65	101

Table D.4. Operations on data set for plotting the scatter chart

Keyboard Type	Move from	Target key	Sample Size	Average of Measured Times ($millisecond$)	D_{ij}	W_j	$log_2(\frac{D_{ij}}{W_j} + 1)$
Square	Key1	Key2	52	134	1.0667	1.0000	1.04731
Longitudinal	Key15	Key27	49	179	1.9230	1.2019	1.37849
Longitudinal	Key14	Key30	43	272	5.9630	1.0164	2.77963
Square	Key2	Key12	39	239	4.3980	1.0308	2.39687
Longitudinal	Key13	Key21	31	321	8.5333	1.0000	3.25298
Square	Key4	Key3	40	137	1.0667	1.0000	1.04731
Square	Key24	Key28	51	217	4.2667	1.1180	2.26794
Square	Key14	Key30	47	235	4.7703	1.1180	2.39694
Square	Key22	Key17	44	152	1.5083	1.4142	1.04723
Longitudinal	Key6	Key15	39	203	2.8720	1.0770	1.87447
Longitudinal	Key3	Key32	46	310	7.7653	1.0400	3.08179
Square	Key6	Key14	35	245	4.7703	1.1180	2.39694
Longitudinal	Key9	Key25	40	303	6.7463	1.0541	2.88754
Longitudinal	Key13	Key10	38	354	9.1293	1.0069	3.33153
Longitudinal	Key24	Key28	40	268	4.2667	1.0000	2.39689
Square	Key8	Key11	41	221	3.2000	1.0000	2.07039
Square	Key6	Key25	46	316	6.8300	1.2806	2.66299
Longitudinal	Key22	Key26	48	290	7.0150	1.0118	2.98790
Square	Key7	Key24	34	268	5.7443	1.0770	2.66304
Longitudinal	Key16	Key20	37	272	4.2667	1.0000	2.39689
Square	Key7	Key7	76	104	0.0000	1.0000	0.00000
Longitudinal	Key3	Key3	65	101	0.0000	1.0000	0.00000

REFERENCES

Burkard, R. E., 2009, *QAPLIB Library, A Quadratic Assignment Problem Library*, http://www.opt.math.tu-graz.ac.at/qaplib.

Beevis, D. and I. M. Slade, 2003, “Ergonomics - costs and benefits”, *Applied Ergonomics*, Vol. 34, pp. 413–418.

Bergamasco, R., C. Girola, and D. Colombini, 1998, “Guidelines for designing jobs featuring repetitive tasks”, *Ergonomics*, Vol. 41, pp. 1364–1383.

Commander, C. W., 2005, “A Survey of the Quadratic Assignment Problem, with Applications”, *Morehead Electronic Journal of Applicable Mathematics*, Issue 4.

Dell’Amico, M., *et al*., 2009, “The single-finger keyboard layout problem”, *Computers & Operations Research*, Vol. 36, pp. 3002 – 3012.

Dorigo, M., 2009, *Ant colony optimization*, Scholarpedia, 2(3):1461, http://www.scholarpedia.org/article/Ant_colony_optimization.

Eggers, J., D. Feillet, S. Kehl, M. O. Wagner, and B. Yannou, 2003, “Optimization of the keyboard arrangement problem using an Ant Colony algorithm”, *European Journal of Operational Research*, Vol. 148, pp. 672–686.

Fagarasanu, M. and S. Kumar, 2003, “Carpal tunnel syndrome due to keyboarding and mouse tasks: a review”, *International Journal of Industrial Ergonomics*, Vol. 31, pp. 119–136.

Fitts, P. M., 1954, “The information capacity of the human motor system in controlling the amplitude of movement”, *Journal of Experimental Psychology*, Vol. 47, pp. 381–391.

Gilad, I. and S. Harel, 2000, “Muscular effort in four keyboard designs”, *International Journal of Industrial Ergonomics*, Vol. 26, pp. 1-7.

Lin Chuan, Y., 2009, “Carpal tunnel syndrome and finger movement dysfunction caused by tophaceous gout: a case report”, *Kaohsiung J Medical Science*, Vol. 25, pp. 34–9.

Lingaard, G. and D. Caple, 2001, “A case study in iterative keyboard design using participatory design techniques”, *Applied Ergonomics*, Vol. 32, pp. 71-80.

Loiola, E. M, *et al.*, 2007, “A survey for the quadratic assignment problem”, *European Journal of Operational Research*, Vol. 176, pp. 657–690.

MacKenzie, I. S., 1995, “Movement time prediction in human-computer interfaces”, *Readings in Human-Computer Interaction*, 2nd ed., pp. 483-493.

MacKenzie, I. S., A. Sellen, and W. Buxton, 1991, “A comparison of input devices in elemental pointing and dragging tasks”, *Proceedings of the CHI ’91: ACM Conference on Human Factors in Computing Systems*, pp. 161–166.

MacKenzie, I. S., and R. W. Soukoreff, 2002, “Text entry for mobile computing: models and methods, theory and practice”, *Human–Computer Interaction*, Vol. 17, pp. 147–198.

MacKenzie, I. S., and S. X. Zhang, 1999, “The design and evaluation of a high performance soft keyboard”, *Proceedings of the CHI ’99: ACM Conference on Human Factors in Computing Systems*, pp. 25–31.

Marklin, R. W., G. G. Simoneau, and J. F. Monroe, 1999, “Wrist and forearm posture from typing on split and vertically inclined computer keyboards”, *Human Factors*, Vol. 41, pp. 559–569.

Nelson, J. E., D. E. Treaster, and W. S. Marras, 2000, “Finger motion, wrist motion and tendon travel as a function of keyboard angles”, *Clinical Biomechanics,* Vol. 15, pp. 489–498.

Niosh, *The National Institute for Occupational Safety and Health Program*, 1998, The Alternative Keyboards, NIOSH Publications, pp. 97-148, http://www.cdc.gov/niosh.

Peosh, *Public Employees Occupational Safety and Health Program*, 1997, Department of Health and Senior Services of State of New Jersey, http://www.state.nj.us/.

Prajapati, C., D. S. Jwalanta, and J. Shishir, 2008, *Nepali Unicode Keyboard Layout Standardization based on Genetic Algorithm*, http://www.jwalanta.com.np/ nepalikeyboard.

Swanson, N. G., *et al*., 1997, “The impact of keyboard design on comfort and productivity in a text entry task”, *Applied Ergonomics*, Vol. 28, No. 1, pp. 9-16.

Vliet, W., 2009, *Heuristics for the Quadratic Assignment Problem*, Bachelor Thesis in Mathematics, Faculty of Mathematics and Natural Sciences, University of Groningen, Holland.

Walker, C. P., 2003, *Evolving a More Optimal Keyboard*, http://citeseerx.ist.psu.edu/viewdoc.

Ward, D., J. Blackwell, A.F. MacKay, 2002, “Dasher: a gesture driven data entry interface for mobile computing”, *Human–Computer Interaction*, Vol. 17, pp. 199–228.

Werner, R., T. J. Armstrong, and M. K. Aylard, 1997, “Intracarpal canal pressure: the role of finger, hand, wrist and forearm position”, *Clinical Biomechanics*, Vol. 12, pp. 44–51.

Wikipedia, 2009, *Fitts’ Law*, http://en.wikipedia.org/wiki/Fitts.

Wikipedia, 2009, *Genetic algorithm*, http://en.wikipedia.org /wiki/Genetic_algorithm.

Wikipedia, 2009, *Human computer interaction*, http://en.wikipedia.org/wiki/HumanComputer Interaction.

Wikipedia, 2009, *Hick's Law*, http://en.wikipedia.org/wiki/Hick's_law.

Wikipedia, 2009, *Metaheuristic*, http://en.wikipedia.org/wiki/Metaheuristic.

Wikipedia, 2009, *Simulated annealing*, http://en.wikipedia.org/wiki/Simulated_annealing.

Yanzhi, L., *et al*., 2006, "A heuristic-based approach to optimize keyboard design for single-finger keying applications", *International Journal of Industrial Ergonomics,* Vol. 36, pp. 695–703.

Yongzong, W., and L. Hazohao, 2006, "A Solution Method for the Quadratic Assignment Problem (QAP)", *the Sixth International Symposium on Operations Research and Its Applications (ISORA'06)*, Xinjiang, China, August 8–12, 2006. ORSC & APORC, pp. 106–117.

Zhai, S. and P. O. Kristensson, 2005, "In Search of Effective Text Input Interfaces for Off the Desktop Computing", *Interacting with Computers,* Vol.17, pp. 229–250.

Zhai, S., 2004, "Characterizing computer input with Fitts' law parameters - the information and non-information aspects of pointing", *International Journal of Human-Computer Studies,* Vol. 61, pp. 791–809.

Zhai, S., M. Hunter and B. A. Smith, 2000, "The Metropolis keyboard - an exploration of quantitative techniques for virtual keyboard design", *Proceedings of the ACM Symposium on User Interface Software and Technology,* UIST 2000, ACM, New York, pp. 119–128.