



Technische Universität München
Fakultät für Mathematik
Lehrstuhl für Angewandte Geometrie und Diskrete Mathematik

Arbeit zur Erlangung des akademischen Grades "Master of Science" (M.Sc.) zum Thema:

Optimierung von Tastenbelegungen für Smartphones

Optimization of keyboard assignments for smartphones

Themensteller:

Prof. Dr. Peter Gritzmann

Betreuer:

Dr. Michael Ritter

B.Sc. Wolfgang Riedl

Bearbeitet von:

B.Sc. Michael Lintl

Matrikelnummer: 3600369

Abgabetermin: 28. April 2014

Ehrenwörtliche Erklärung

Ich erkläre hiermit ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus fremden Quellen (einschließlich elektronischer Quellen) direkt oder indirekt übernommenen Gedanken sind ausnahmslos als solche kenntlich gemacht.

München, den 23. April 2014

B.Sc. Michael Lintl

“Proposing a new [...] keyboard is like proposing to reverse the Ten Commandments and the Golden Rule, discard every moral principle, and ridicule motherhood”

Dr. August Dvorak, Juli 1970

Danksagung

Ich bedanke mich bei Herrn Prof. Dr. Gritzmann für die Möglichkeit meine Masterarbeit an seinem Lehrstuhl zu schreiben.

Ein besonderer Dank für ihre Zeit und Mühen gilt meinen beiden Betreuern Dr. Michael Ritter und B.Sc. Wolfgang Riedl, die mir mit wertvollen Dankanstößen und konstruktiver Kritik zur Seite standen.

Darüber hinaus bedanke ich mich bei meiner Familie für die vielfältige Unterstützung in allen Lebenslagen.

Meinen Kommilitonen Jonas Wunderlich und Johannes Hofbauer bin ich dankbar für die fachliche Zusammenarbeit und die moralische Unterstützung in den vergangenen Jahren.

Ganz spezieller Dank gebührt meiner wundervollen Frau Johanna.

Zusammenfassung

Die zunehmende Verbreitung von Smartphones in den letzten Jahren hat zu einer vermehrten Nutzung virtueller Tastaturen geführt und es wurde der Ruf nach einer Alternative zur allgegenwärtigen *QWERTY*-Tastatur laut. Einer der Hauptgründe dafür ist, dass sich die Bedienung eines Smartphones deutlich vom Schreiben auf Schreibmaschinen, für welche die *QWERTY*-Tastatur ursprünglich entwickelt wurde, unterscheidet.

Die vorliegende Arbeit widmet sich der Optimierung von Tastenbelegungen ausgewählter Layouts mit dem Ziel die erwartete zurückzulegende Strecke und die erwartete Schreibdauer eines deutschen Textes zu minimieren. Dabei steht zunächst die einhändige, vertikale Textübertragung im Fokus. Das zugehörige Optimierungsproblem, welches die Suche nach einer leistungsfähigen Tastenbelegung beschreibt, lässt sich als *lineares Zuordnungsproblem* modellieren und kann leicht gelöst werden. Jedoch können die Ergebnisse hinsichtlich ihrer Eignung in der Praxis nicht überzeugen, weshalb unter Verwendung zusätzlicher binärer Variablen ein weiterentwickeltes Optimierungsproblem formuliert wird. Das resultierende *quadratische Zuordnungsproblem* ist NP-hart und deshalb nicht für die Lösung von Instanzen realistischer Größenordnung geeignet. Im Anschluss an dessen Linearisierung werden verschiedene Ansätze zur Verbesserung der zeitlichen Lösbarkeit entwickelt und getestet. In Kombination mit einem Verfahren, das die Anzahl der Variablen reduziert und die globale Optimalität aufgibt, ebnet sie den Weg zu Tastenbelegungen, deren Qualität die bisheriger Tastaturen um bis zu 53% übertrifft.

Neben der einhändigen, vertikalen Handhabung werden Smartphones oftmals in horizontaler Position genutzt und mit beiden Händen bedient. Die vorliegende Arbeit überträgt die Theorie der Optimierung von Tastenbelegungen auf diese relativ neue Art der Bedienung und erweitert so das bisherige Angebot an Literatur für die 2-Daumen-Schreibweise. Außerdem liefert sie Tastenbelegungen, welche für die horizontale Textübertragung optimiert wurden und im Vergleich zur *QWERTY*-Tastatur potentielle Verbesserungen von bis zu 36% erzielen.

Besondere Anforderungen an Tastenbelegungen stellt die abwechselnd vertikale und horizontale Textübertragung, die in der Realität sehr häufig vorkommt, in der vorliegenden Arbeit aber erstmals wissenschaftlich betrachtet wird. In diesem Zusammenhang werden Tastenbelegungen erzeugt, welche für beide Gerätepositionen geeignet sind und damit den problemlosen Wechsel zwischen den Bedienkonzepten ermöglichen. Die Untersuchung und Auswertung der Resultate zeigt eine Überlegenheit der optimierten Belegungen gegenüber der *QWERTY*-Tastatur von bis zu 33%.

Abstract

The vast spread of smartphones in the last decade and thus the rising use of virtual keyboards is accompanied by an increasing demand for an alternative to the standard *QWERTY*-keyboard. The reasons for this lay in the different handling concepts of a smartphone and a typewriter for which *QWERTY* was originally designed.

Intending to minimize the expected total time and the expected travel distance for german text this thesis focuses on the optimization of key assignments on given layouts. The first goal is to develop high-performance assignments for single-finger writing in vertical device position. The modeling yields a *linear assignment problem* which can be easily solved but doesn't generate satisfying results. Using additional binary variables a more sophisticated model arises. This new formulation is the *quadratic assignment problem* which is known to be NP-hard and therefore not suited for solving realistic data. In order to improve running times various methods are developed and tested after the problem was linearized. Together with a technique reducing the number of variables and abandoning global optimality they clear the way for key assignments which perform up to 53 % better than common keyboards.

Besides vertical single-finger utilization smartphones are often used in horizontal position and both hands are employed for text entry. The transfer and adaption of previously investigated theory provide the basis for optimizing keyboard assignments for this relatively new 2-thumb-writing-method. This thesis enlarges the set of available literature on this topic and provides assignments with up to 36 % advantage over *QWERTY*.

The omnipresent text entry in alternating positions leads to special requirements on the key assignment and is studied the first time in this thesis. Switching between vertical and horizontal handling concepts doesn't cause any difficulties due to the optimized keyboard assignments. Analysis and evaluation reveal a potential improvement of up to 33 % compared to the *QWERTY*-assignment.

Inhaltsverzeichnis

Zusammenfassung	ix
I Einleitung und Grundlagen	1
1 Einleitung	3
1.1 Einsatz und Bedeutung virtueller Tastaturen auf Smartphones	3
1.2 Geschichtliche Entwicklung von Tastaturen	4
2 Grundlagen	9
2.1 Optimierungspotential von Tastaturen und Tastenbelegungen	9
2.2 Stand der Forschung	10
2.3 Voraussetzungen und Thematik der vorliegenden Arbeit	17
2.3.1 Voraussetzungen und Konventionen	17
2.3.2 Inhalt und Aufbau	18
2.3.3 Grundlegende Begriffe und Notationen	20
2.4 Daten und deren Beschaffung	20
2.4.1 Buchstaben- und Bigrammwahrscheinlichkeiten	21
2.4.2 Textlänge	21
2.4.3 Tastaturlayouts	22
2.4.4 Entfernungen der Tasten	24
2.4.5 Bewegungsdauer	27
2.4.6 System und Software	28
II Optimierung von Tastenbelegungen	29
3 Distanzminimierung für vertikale Anwendungen	31
3.1 Modellierung als lineares Zuordnungsproblem	31
3.1.1 Beschreibung und mathematische Formulierung	31
3.1.2 Ergebnisse der Optimierung	34
3.2 Modellierung als quadratisches Zuordnungsproblem	36
3.2.1 Beschreibung und mathematische Formulierung	36

3.2.2	Linearisierung des Problems	41
3.2.3	Ansätze zur Verbesserung der Lösungsdauer des linearisierten Problems	46
3.2.4	Auswege aus der zeitlichen Unlösbarkeit	79
3.2.5	Ergebnisse der Optimierung	89
3.3	Verallgemeinerung des verbesserten, linearen Problems	90
3.3.1	Übertragung der Ansätze	91
3.3.2	Formulierung eines verallgemeinerten Problems	93
3.3.3	Ergebnisse der Optimierung	94
3.4	Zusammenfassung der Ergebnisse im vertikalen Fall	95
4	Distanzminimierung für horizontale Anwendungen	97
4.1	Vereinfachte Modellierung der beidhändigen Bedienung	97
4.1.1	Beschreibung und mathematische Formulierung	97
4.1.2	Ergebnisse der Optimierung	98
4.2	Realistische Modellierung der beidhändigen Bedienung	100
4.2.1	Naive Wahl der Wartepositionen	101
4.2.2	Praxisnahe Wahl der Wartepositionen	103
4.2.3	Ergebnisse der Optimierung	104
4.3	Zusammenfassung der Ergebnisse im horizontalen Fall	106
5	Distanzminimierung für Hybrid-Anwendungen	107
5.1	Modellierung	108
5.2	Ergebnisse der Optimierung	109
5.3	Zusammenfassung der Ergebnisse im Hybrid-Fall	111
6	Minimierung der benötigten Zeit	113
6.1	Zeitminimierung für vertikale, horizontale und Hybrid-Anwendungen	113
6.1.1	Modellierung	113
6.1.2	Ergebnisse der Optimierung	116
6.2	Zeitreduktion mithilfe der Hover-over-Schreibweise	118
6.2.1	Ergebnisse im horizontalen Fall	120
6.2.2	Ergebnisse in Hybridanwendungen	122
III	Implementierung	125
7	Implementierung	127
7.1	Aufbau der Inputdaten	127
7.2	Schematischer Aufbau des Tools <i>VirtualKeyboardOptimization.mos</i>	129
7.3	Funktionen von <i>VirtualKeyboardOptimization.mos</i>	129

IV	Auswertung und Diskussion der Ergebnisse	133
8	Auswertung und Vergleich der Ergebnisse	135
8.1	Theoretische Bestimmung der Schreibgeschwindigkeit	135
8.2	Alternative Tastaturen und Tastenbelegungen	136
8.2.1	Belegungen des <i>QWERTY-like</i> -Layouts	136
8.2.2	<i>DVORAK</i> -Tastatur	138
8.2.3	<i>KALQ</i> -Tastatur	138
8.2.4	Weitere Belegungen für rechteckige Layouts	138
8.3	Vergleich der optimierten Tastenbelegungen mit alternativen Tastaturen	140
8.3.1	Vergleich der Eigenschaften im vertikalen Fall	140
8.3.2	Vergleich der Eigenschaften im horizontalen Fall	142
8.3.3	Vergleich der Eigenschaften im Hybrid-Fall	144
8.3.4	Zusammenfassung der Ergebnisse der Vergleiche	145
9	Diskussion	149
10	Ausblick	153
Anhang		155
A	Input-Daten	157
A.1	Konkrete Buchstaben- und Bigrammwahrscheinlichkeiten	157
A.2	Numerische Beschreibung der Layouts	160
A.2.1	Layout 1 - <i>QWERTY-like</i>	160
A.2.2	Layout 2 - <i>3x9</i> -Layout	164
A.2.3	Layout 3 - <i>3x10</i> -Layout	168
A.2.4	Layout 4 - <i>4x8</i> -Layout	173
A.2.5	Layouts der Testinstanzen 1 und 2 - <i>3x3</i> - und <i>2x4</i> -Layout	179
A.2.6	Zusätzliche Layouts - <i>5x6</i> - und <i>5x7</i> -Layout	181
B	Quellcode	193
	Literaturverzeichnis	217
	Abbildungsverzeichnis	229
	Tabellenverzeichnis	233

Teil I

Einleitung und Grundlagen

1 Einleitung

1.1 Einsatz und Bedeutung virtueller Tastaturen auf Smartphones

Das steigende Bedürfnis nach Vernetztheit und ständiger Erreichbarkeit hat in den vergangenen Jahren zu einem rasanten Anstieg der Nachfrage und Verbreitung von mobilen Endgeräten geführt [Pols, 2013]. Im Mittelpunkt dieser Entwicklung stehen zweifellos Smartphones, welche verschiedenste Funktionen in sich vereinen [Bao et al., 2011]. So dienen neue Smartphones beispielsweise als Telefon, Nachrichtenzentrale, Internetschnittstelle und Navigationsgerät.

In den meisten Anwendungen steht die Übertragung von Text im Mittelpunkt der Kommunikation zwischen Nutzer und Endgerät, weshalb eine effiziente Textübertragung enorme Bedeutung für die effektive Smartphone-Nutzung hat. Die Übermittlung eines Textes kann auf einem Smartphone durch mehrere alternative Arten erfolgen, welche sich in drei Gruppen einteilen lassen. Bei der akustischen Übertragung wird der Text in ein Mikrofon gesprochen und mit geeigneter Software digitalisiert. Die visuelle Übertragung funktioniert mithilfe einer Kamera. Die Texteingabe über ein Textfeld oder eine Tastatur bezeichnet man auch als taktile Übermittlung [Schukat-Talamazzini, 1995].

Um eine passende Art der Textübertragung auswählen zu können, müssen je nach Anwendung die Vor- und Nachteile der einzelnen Gruppen gegeneinander aufgewogen werden (siehe Abschnitt 2.1). In den allermeisten Fällen erfolgt die Texteingabe jedoch über eine Tastatur, welche bei Bedarf auf dem Bildschirm eingeblendet wird. Diese sogenannten virtuellen Tastaturen verwenden fast immer das *QWERTY*-Layout und die *QWERTY*-Tastenbelegung (siehe Abb. 57).

Neben dem Einsatz auf Smartphones ist diese Belegung in Ländern mit lateinischem Alphabet seit beinahe 100 Jahren der Standard auf Schreibmaschinen und Computern. Die lange Nutzungsdauer und die daraus resultierende Vertrautheit dürften die Hauptgründe für die Adoption der *QWERTY*-Tastatur auf Smartphones sein. Während beim Umstieg von Schreibmaschinen auf Computer einige Argumente, wie der Transfer bereits erlernter Schreibfähigkeiten, für die Übernahme dieser Tastenbelegung sprachen, gab es bei der Einführung von Smartphones außer oben genannter Gewohnheit keinen Grund, die *QWERTY*-Tastatur beizubehalten. Es gibt sogar einige Punkte, die gegen ihre Verwendung auf Smartphones sprechen.

Eine charakteristische Eigenschaft der *QWERTY*-Belegung ist die Verteilung von aufeinanderfolgenden Tastenanschlägen auf verschiedene Hände. Diese Eigenschaft, welche bei der beid-

händigen Bedienung einer Computertastatur Vorteile hinsichtlich der Schreibgeschwindigkeit mit sich bringt, entpuppt sich auf einem Smartphone als Nachteil. Der Grund dafür liegt in der unterschiedlichen Schreibweise, da ein Smartphone in vertikaler Position oftmals mit einem Finger bedient wird. Dies geschieht in der Regel mit dem Daumen der dominanten Hand. Es ist offensichtlich, dass oben genannte Buchstabenverteilung größere, vom Finger zurückzulegende Distanzen provoziert, als es vielleicht bei einer Alternativbelegung der Fall wäre. Dies führt zu einer geringeren Schreibgeschwindigkeit, welche sich als Flaschenhals einer schnellen und effizienten Textübertragung in vertikaler Geräteposition identifizieren lässt [Zhai et al., 2002a] [Francis u. Rash, 2003].

In den letzten Jahren wurde durch den Einbau von Lagesensoren in Smartphones das schnelle Umschalten zwischen vertikalem und horizontalem Bildschirm möglich. Die zunehmende horizontale Nutzung führt dazu, dass vermehrt auch mit beiden Händen geschrieben wird. Auf den ersten Blick scheint die *QWERTY*-Tastatur dank der polaren Buchstabenverteilung gut für diese 2-Daumen-Schreibweise geeignet zu sein, jedoch besteht aufgrund anderer Eigenschaften, wie beispielsweise häufig nötigen Zeilenwechselln (siehe [Noyes, 1998]), auch hier Optimierungspotential.

Die zum Teil konträren Anforderungen der beiden positionsabhängigen Bedienkonzepte eines Smartphones gilt es auf der Suche nach einer verbesserten Tastenbelegung, welche das Problem der ineffizienten Texteingabe löst, zu vereinbaren.

1.2 Geschichtliche Entwicklung von Tastaturen

Die Geschichte der Tastaturen beginnt mit der Erfindung der Schreibmaschine. Die erste Beschreibung einer "[...] künstlichen Maschine oder Methode [...]", die in der Lage ist "Buchstaben abzudrucken oder abzuschreiben, [...] fortlaufend [...], sodass jeglicher Text klar und deutlich und deshalb nicht vom Druck zu unterscheiden, auf Papier ausgefertigt werden kann [...]" findet sich in einem Patent, welches dem englischen Erfinder Henry Mill 1714 von der britischen Königin Anne erteilt wurde [Mill, 1714]. Die älteste nachweislich funktionierende Schreibmaschine baute der Italiener Pellegrino Turri im Jahr 1808. Wie die meisten Schreibmaschinen zu Beginn des 19. Jahrhunderts war diese noch dafür konzipiert Blinden das Schreiben zu ermöglichen. Dies änderte sich jedoch mit der sogenannten Schnellschreibmaschine, welche Karl Drais 1829 erstmals präsentierte. Trotz der wachsenden Bedeutung geschäftlicher Korrespondenz Mitte des 19. Jahrhunderts und vielen Weiterentwicklungen und neuen Erfindungen in diesem Bereich ging keine der Schreibmaschinen in Serienproduktion. Erst der 1868 von Christopher Latham Sholes, Carlos Glidden und Samuel W. Soule gebaute *Sholes-and-Glidden-typewriter* konnte sich erfolgreich durchsetzen und wurde ab 1876 von der Firma *E. Remington & Sons* produziert [Buzing, 2003] [Lessing, 2010].

Während fast alle bis dato gebauten Texteingabegeräte auf eine alphabetische Tastenbelegung zurückgriffen, konnte der *Sholes-and-Glidden-typewriter*, welcher später leicht modifiziert unter dem Namen *Remington No. 2* verkauft wurde, bereits eine *QWERTY*-ähnliche Tastatur vorweisen. Über die Gründe, die Sholes zu dieser Tastenbelegung bewegten, ist man sich bis heute uneinig. Einer der Meistgenannten ist, ein Verhaken der Typenhebel, das sogenannte "mechanical jamming", zu verhindern, was bei alphabethischen Tastaturen regelmäßig vorkam und den Schreibfluss enorm behinderte. Sholes soll dies durch die räumliche Trennung von besonders häufig auftretenden Buchstabenpaaren und somit einem ausreichenden Abstand der Typenhebel realisiert haben (siehe [Jain u. Bhattacharya, 2010], [Buzing, 2003], [Zhai et al., 2002a], [Yamada, 1980]). Yasuoka u. Yasuoka [2011] sind der Ansicht, dass die Anordnung der Tasten vom Morse-Alphabet inspiriert wurde und nicht wie anderweitig behauptet dem Marketing diene, indem man das Wort "Typewriter" nur mit den Tasten der obersten Tastaturreihe zu Papier bringen kann [David, 1985]. Andere, weniger verbreitete Versionen über Sholes Belegungswahl sind, dass die unintuitive Anordnung ein Überbleibsel aus seiner früherer Tätigkeit als Drucker war, aber auch Griffiths Hypothese, wonach die *QWERTY*-Belegung nur eine Variante des Alphabets ist, zu der man mit zwei Umordnungsschritten gelangt [Noyes, 1998].

Aufgrund der hohen Verkaufszahlen der *Remington No. 2*, dem Zusammenschluss der fünf größten Schreibmaschinen-Hersteller 1893 zur *Union Typewriter Company* und der Tatsache, dass diese sich auf ein Tastaturlayout und eine gemeinsame Tastenbelegung einigen konnten, wurde *QWERTY* in den Jahren um die Jahrhundertwende herum bekannt [Davis, 2005]. Bis auf wenige Umordnungen und fehlende Sonderzeichen, welche im Lauf der Zeit hinzukommen sollten, entsprach die damalige Tastatur bereits der heute üblichen und weit verbreiteten Standard-Computertastatur. Da Christopher Latham Sholes der Auslöser dieser Entwicklung war, gilt er als der Erfinder der *QWERTY*-Tastatur [Yasuoka u. Yasuoka, 2011].

Trotz der zunehmenden Verbreitung von *Remington*-Schreibmaschinen mit *QWERTY*-Belegung gab es viele konkurrierende Hersteller, welche sich durch unterschiedliche Funktionsweisen oder mithilfe alternativer Tastenbelegungen auf dem Markt behaupten wollten [Noyes, 1998]. Zu dieser Zeit war *Remington* der Marktführer, begründet hauptsächlich durch die kluge Verkaufstrategie, sowie die mit diversen Siegen in Schnellschreibwettbewerben einhergehende öffentliche Aufmerksamkeit [Liebowitz u. Margolis, 1990]. Neben dem reinen Verkauf von Geräten bot *Remington* Kurse an, in denen Benutzer die damals aufkommende 10-Finger-Schreibweise erlernen konnten, um sie so an ihre Schreibmaschinen zu binden [Buzing, 2003]. Obwohl keiner der Konkurrenten die Marktführerschaft von *Remington* ernsthaft gefährden konnte, schafften es weder *QWERTY* noch eine andere Tastatur, sich zu etablieren und es wurden nach wie vor verschiedene Layouts und Belegungen verwendet [Robert, 2001].

Erst in den 1920er Jahren konnte sich die *QWERTY*-Tastatur gegenüber den anderen durchsetzen, indem mehrere Länder die Tastenbelegungen standardisierten und das *QWERTY*-Layout samt zugehöriger Belegung, vermutlich aufgrund des zeitlichen und zahlenmäßigen

Vorsprungs in Normen aufnahmen (z.B. DIN 2112: Tastenanordnungen für handbetriebene Schreibmaschinen). Die Standardisierung hatte zur Folge, dass die meisten Schreibmaschinen ab diesem Zeitpunkt mit *QWERTY*-Tastaturen produziert wurden und neue Schreibkräfte fast ausschließlich auf *QWERTY* ausgebildet wurden. Selbst ernstzunehmende Ansätze, wie der von August Dvorak 1932, welcher auf die Optimierung hinsichtlich ergonomischer Gesichtspunkte zielt [Dvorak, 1943], konnten sich nicht durchsetzen. Die Gründe hierfür sind vor allem die mangelnde Bereitschaft der Konsumenten, das Schreiben auf einer neuen, ungewohnten Tastatur zu erlernen, sowie der finanzielle Aufwand, der durch das Ersetzen der weit verbreiteten *QWERTY*-Schreibmaschinen entstanden wäre [David, 1985].

Ende der 1980er Jahre erreichte die Verbreitung der Schreibmaschinen ihren Höhepunkt. In der Folge wurden sie immer häufiger durch Computer und Drucker ersetzt. Als Texteingabegeräte für Computer dienten Tastaturen, deren Tasten analog zur vertrauten *QWERTY*-Tastatur einer Schreibmaschine angeordnet und belegt waren. Trotz der Weiterentwicklung jedes Bauteils und zahlreicher Innovationen im Bereich der Informationstechnologie wurde die Tastenbelegung einer Computertastatur nie verändert.

Einige neuere Ansätze mit dem Ziel die *QWERTY*-Tastatur auf Computern zu reformieren sind die *Colemak*-Belegung, das *Neo-Design* oder das *RISTOME*-Layout (siehe [Menzel et al., 2005], [Behrens, 2004], [Coleman, 2006]). Die Wahrscheinlichkeit, dass dies gelingt, dürfte jedoch gering sein, da sich an der Umlern-Problematik und dem finanziellen Aufwand für den Austausch der gewohnten und derzeit genutzten Tastaturen wenig geändert hat. Vieles spricht dafür, dass eine umfassende Neuordnung oder Umbelegung der Tasten von Computertastaturen, abgesehen von einigen speziellen Anwendungsbereichen wie beispielsweise der Programmierung, nicht stattfinden wird (siehe Abschnitt 2.1).

Nachdem es ab 1980 zwei Jahrzehnte lang vor allem Computer waren, welche sich rasend schnell ausbreiteten und sowohl in privaten wie geschäftlichen Bereichen Einzug hielten, waren es in den Jahren um die Jahrtausendwende Mobiltelefone. Während die ersten tragbaren Telefone noch mit einer reinen Zifferntastatur ohne Buchstaben auskamen, waren spätere Geräte bereits mit Tastaturen ausgestattet, bei denen mehrere Buchstaben in alphabetischer Reihenfolge auf einer Taste gebündelt wurden. Benötigt wurden sie in den Anfangsjahren vor allem zum Versenden von Textnachrichten. Über die Jahre vervielfachten sich die Einsatzmöglichkeiten der Mobiltelefone und es etablierte sich die Sparte der Smartphones. Aufgrund der zunehmenden Anzahl von Anwendungen mit Texteingabe (Email, Chat, soziale Netzwerke, etc.) und der damit verbundenen regelmäßigen Verwendung der Tastaturen setzten manche Hersteller zunächst auf kleine, vollwertige *QWERTY*-Tastaturen, deren dauerhafter Erfolg jedoch ausblieb.

Mit dem Einbau von berührungsempfindlichen Bildschirmen in Smartphones begann die vorerst letzte Epoche in der Geschichte der Tastaturen. Abermals orientierte man sich beim Design dieser virtuellen Tastaturen an der vertrauten *QWERTY*-Belegung. In Anlehnung an frühere Mo-

biltelefone bediente man die neuen Geräte zu Beginn in vertikaler Position, oftmals mit einem zugehörigen Stift oder einem Finger. Doch spätestens seit der Integration von Lagesensoren in Smartphones kommt es häufig vor, dass die Texteingabe auch in horizontaler Geräteposition erfolgt und dafür beide Daumen benutzt werden.

2 Grundlagen

Dieses Kapitel widmet sich der zugrundeliegenden Motivation der Forschung im Bereich der Tastaturentwicklung und liefert einen Überblick über bisherige Ansätze und Ergebnisse. Es folgt eine Abgrenzung der vorliegenden Arbeit zu bisheriger Literatur sowie das Abstecken des Forschungsbereichs durch die Festlegung auf gewisse Voraussetzungen. Ein Abschnitt zum Thema Datenbeschaffung bildet den Abschluss dieses Kapitels.

2.1 Optimierungspotential von Tastaturen und Tastenbelegungen

Bevor das Potential analysiert wird, das in einer Überarbeitung der Tastenbelegung einer virtuellen Tastatur steckt, werfen wir einen Blick auf die Vor- und Nachteile der einzelnen Textübertragungsmethoden. Die akustische Übertragung besitzt sicherlich das größte Potential im Bezug auf die Eingabegeschwindigkeit. Hierbei ist es theoretisch möglich, bis zu 250 Wörter pro Minute in digitalen Text zu verwandeln, ohne zuvor geübt zu haben, wie es beispielsweise für eine Geschwindigkeit von 100-150 Wörter pro Minute auf einer Tastatur erforderlich wäre. Außerdem benötigt der Benutzer weder Hände noch Augen für eine Übermittlung, so dass er parallel andere Aufgaben erledigen kann. Der Grund warum sich die Spracherkennung bisher nur in wenigen Bereichen der Digitalisierung von Text durchgesetzt hat, ist die relativ schlechte Erkennungsrate für allgemeine, nicht wortschatz- oder kontextbeschränkte Texte [Schukat-Talamazzini, 1995].

Sowohl die visuelle Textübertragung als auch die Handschrifterkennung können nur in Bereichen eingesetzt werden, in denen die Eingabegeschwindigkeit keine wichtige Rolle spielt, da man mithilfe dieser Methoden in der Regel nicht mehr als 25 Wörter pro Minute übermitteln kann. Ähnlich zu akustischen Verfahren hängt die zu erwartende Fehlerrate stark von der Komplexität des Textes ab und ist in der Praxis nur für Anwendungen mit sehr geringem Wort- oder Zeichenschatz akzeptabel. Die höchste Erkennungsgenauigkeit von annähernd 100% bietet eine Tastatur - mit einer erzielbaren Geschwindigkeit von ungefähr 150 Wörtern pro Minute auf einer vollwertigen Hardware-Tastatur. Im Vergleich dazu ist die Übermittlung von Text mit etwa 30 Wörtern pro Minute auf einer virtuellen *QWERTY*-Tastatur deutlich langsamer [Schukat-Talamazzini, 1995] [Wienecke, 2003] [Schenk u. Rigoll, 2010] [Zhai et al., 2002a].

Trotz der geringeren Geschwindigkeit im Vergleich zu anderen Texteingabemethoden ist die Tastatur derzeit das Mittel der Wahl in puncto Texteingabe. Der Hauptgrund für den um ein

Vielfaches häufigeren Einsatz liegt in der geringeren Fehlerrate. Darüber hinaus ist im Gegensatz zur Spracherkennung die Übertragung eines Textes in der Öffentlichkeit problemlos, ohne Einbüßen der Privatsphäre möglich.

Die Geschwindigkeit der Textübermittlung mithilfe einer Tastatur hängt maßgeblich von der Anordnung, Form, Größe und Belegung der Tasten ab. Genau diese Faktoren sind es also, die sich prinzipiell für eine Optimierung eignen. In jede der Richtungen besitzt die gegenwärtige Standardtastatur Optimierungspotential (siehe Abschnitt 2.2). Insbesondere im Bereich der virtuellen Tastaturen sind der Variabilität kaum Grenzen gesetzt, da sie vollständig auf Software basieren und somit schnell und kostengünstig angepasst werden können. Die vorliegende Arbeit widmet sich speziell der Belegung von Tastaturen (siehe Abschnitt 2.3).

Wie in Abschnitt 1.2 beschrieben, werden in nahezu allen Ländern mit lateinischem Alphabet standardmäßig Tastaturen mit *QWERTY*-Belegung produziert und verwendet. In der Herstellung von Schreibmaschinen und dem Bereich der Computer ist es höchst unwahrscheinlich, dass sich daran etwas ändert und sich eine andere, möglicherweise effizientere Tastenbelegung durchsetzen kann [Zhai et al., 2000]. In den Wirtschaftswissenschaften wird dies oftmals damit begründet, dass der Technikpfad bereits zu weit beschritten sei, was aufgrund der Unumkehrbarkeit von Investitionen ein Umschwenken auf Alternativen praktisch unmöglich macht [David, 1997]. Außerdem beruft man sich in diesem Zusammenhang häufig auf die große *QWERTY*-Nutzerzahl und den aus einem Tastaturwechsel resultierenden Aufwand [Bi et al., 2012].

Die Erfolgswahrscheinlichkeit für eine Veränderung der Standardtastatur von Touchscreengeräten ist deutlich höher, da sich aufgrund der kurzen Lebensdauer und regelmäßiger Software-Updates die Einführung einer neuen Tastenbelegung oder eines neuen Layouts ohne finanziellen Aufwand durchführen lässt. Ein weiterer Grund, der für eine tiefgehende Analyse von virtuellen Tastaturen zum jetzigen Zeitpunkt spricht, ist die noch relativ kleine Zahl an Smartphone-Nutzern weltweit, sodass sich das Umlernen in Grenzen hält. Die viel größere Gruppe der potentiellen Neukunden hätte kaum zusätzlichen Aufwand, da sie nicht von ihren Erfahrungen auf einer Computertastatur profitieren kann, weil sich die Schreibweise auf einem mobilem Endgerät mit Touchscreen deutlich vom Schreiben auf einer Hardwaretastatur unterscheidet (siehe Abschnitt 1.1 und [Zhai et al., 2000]).

2.2 Stand der Forschung

Im Bereich der Textübertragung mithilfe von Tastaturen gibt es eine Vielzahl unterschiedlichster Forschungsansätze. Angesichts der langen Geschichte und der großen Bedeutung, die dieses Forschungsfeld aufgrund der vielen Nutzer und der häufigen Verwendung von Tastaturen hat, ist dies kaum verwunderlich. Nachfolgend wird eine Auswahl von Arbeiten vorgestellt.

Die Erforschung von Texteingabe lässt sich grob, aber nicht nur, in vier Hochphasen unterteilen, die jeweils mit der Etablierung neuer technischer Geräte einhergingen. Natürlich sind die Übergänge zwischen den Phasen oft fließend oder es wurde parallel in mehreren Bereichen geforscht.

In der ersten frühen Phase waren es vor allem Erfinder, die es sich zum Ziel gesetzt hatten, ihre eigenen Entwürfe einer Schreibmaschinentastatur (oftmals samt neuem Schreibmaschinenmodell) auf dem Markt zu platzieren. Später waren es Wissenschaftler, die die Vormachtstellung von *QWERTY* brechen wollten, indem sie auf die ergonomischen Nachteile hinwiesen. Die zunehmende Verbreitung von Computern und ihre wachsende Bedeutung für Bürotätigkeiten zu Beginn der 1980er Jahre war der Auslöser der zweiten Phase. Das dritte Mal rückten Tastaturen mit der immer alltäglicheren Nutzung von Mobiltelefonen ab 1990 in den Mittelpunkt. Diese dritte Phase lässt sich unterteilen in die Erforschung numerischer Tastaturen und die Entwicklung virtueller Tastaturen, welche durch den Einbau von Touchscreens in mobile Geräte realisiert wurden. Beiden Tastaturarten liegt jedoch die gleiche einhändige Schreibweise zugrunde. Die vierte und vorerst letzte Phase beginnt mit der Erfindung des Smartphones. Durch eingebaute Lagesensoren etablierte sich eine neue Schreibweise, die auf der Verwendung beider Daumen basiert.

1. Phase

Frühe Versuche die *QWERTY*-Tastatur auf Schreibmaschinen zu ersetzen gab es bereits kurz nach ihrer Einführung. Zu den Bekannteren zählen jene von Rowell 1909 und Nelson 1920, die Buchstaben- und Bigrammhäufigkeiten zugrundelegten, sowie die 1914 vorgestellte geteilte Tastatur von Heidner (siehe [Noyes, 1998]). Die bedeutendste und in diesem Zusammenhang immer wieder genannte Arbeit ist die von August Dvorak, der die Tastenbelegung in den 1930er Jahren durch ergonomische Überlegungen zu reformieren versuchte [Dvorak, 1943]. Mehrere Jahrzehnte gab es keinen Zweifel, dass die *DVORAK*-Belegung (siehe Abb. 62) dem *QWERTY*-Design in puncto Geschwindigkeit überlegen sei. Jedoch wurde diese Annahme, die sich vermutlich aufgrund einer im Nachhinein fragwürdigen Studie der US Navy durchsetzte, von Liebowitz u. Margolis [1990] in Frage gestellt. West [1998] und Buzing [2003] kamen zu dem Ergebnis, dass der Unterschied in der Schreibgeschwindigkeit lediglich 4-5 % beträgt und Vorteile von Dvoraks Tastenbelegung eher im Bereich der ausgeglicheneren Verwendung der Hände liegen, was eine spätere Ermüdung zur Folge hat. Auch in den darauffolgenden Jahrzehnten war man auf der Suche nach besseren Tastenbelegungen für Schreibmaschinen, stellte theoretische Vergleiche zwischen den Kandidaten und der Standardtastatur an und führte praktische Studien zur Evaluation durch (siehe u.a. [Kinkead, 1975] [Burkard u. Offermann, 1977] [Michaels, 1971] [Hirsch, 1970]).

2. Phase

Die Einführung von Computern markiert einen Punkt in der Geschichte, an dem sich die Forschung im Bereich der Texteingabe in mehrere unterschiedliche Richtungen entwickelte. So existierten nach wie vor die klassischen Forschungsfragen, wie die nach einer guten Tastenbelegung (siehe bspw. [Malt, 1977]). Nebenbei rückte jedoch die Entwicklung neuer Tastaturmodelle, die durch den Anschluss an den Computer und seiner Rechenleistung möglich geworden sind, ins Blickfeld [Kölsch u. Turk, 2002]. Mehr und mehr beschäftigte man sich auch mit den gesundheitlichen Auswirkungen der intensiven Nutzung von gewöhnlichen Tastaturen und weniger mit deren Belegungen oder Designs. Eine ausführliche Auflistung von Literatur und Studien zum Thema Tastaturen findet sich in [Kroemer, 2001]. Trotz der weitestgehenden Akzeptanz der *QWERTY*-Computertastatur im Alltag wird weiter an effizienteren Belegungen für spezielle Bereiche gearbeitet. Ein Beispiel dafür ist das *NEO*-Design, welches Sonderzeichen, wie sie in der Programmierung häufig verwendet werden, statt an den Rand der Tastatur auf zusätzlichen Ebenen platziert (siehe [Behrens, 2004]).

3. Phase

Die zunehmende Verbreitung von Mobiltelefonen führte zu einer Intensivierung der Forschung an Tastaturen und leutete damit die dritte Phase ein. Zentrales Forschungsgebiet war zu Beginn die Schreibgeschwindigkeit und die Uneindeutigkeit der Eingabe von numerischen Tastaturen (siehe [Oommen et al., 1991] [Leshner et al., 1998] [Silfverberg et al., 2000]). Die Integration von Wörterbüchern und die Verwendung von Algorithmen, die Worte gemäß ihrer Häufigkeit in der jeweiligen Sprache vorschlagen, löste letzteres Problem [MacKenzie u. Soukoreff, 2002b].

In den Jahren um die Jahrtausendwende verbreiteten sich Geräte mit berührungsempfindlichen Bildschirmen als eine Weiterentwicklung gewöhnlicher Mobiltelefone. Daraufhin konzentrierten sich unzählige Projekte auf das Verständnis und die Beschleunigung der einhändigen stift- oder fingerbasierten Schreibweise auf Touchscreens. Eine Auswahl davon sei im Folgenden vorgestellt.

MacKenzie et al. [1999] vergleichen die Eingabgeschwindigkeit auf sechs Tastaturen, welche sich sowohl im Design als auch in der Tastenbelegung unterscheiden (*QWERTY*, *Dvorak*, *ABC*, *FITALY* (siehe Abb. 65), *JustType*, *telephone*). Dabei wird zwischen der Bedienung durch Anfänger und Experten unterschieden. Alle Tastaturen haben gemein, dass sie nur die 26 Buchstaben samt Leertaste(n) enthalten. Die Vorhersagen zur theoretischen Schreibgeschwindigkeit basieren dabei auf einem in [William Soukoreff u. Scott Mackenzie, 1995] entwickelten Modell, welches neben der reinen Bewegungsdauer auch die Zeit berücksichtigt, die nötig ist, um die Position eines Buchstabens zu identifizieren.

Via "Trial-and-error" entwerfen MacKenzie u. Zhang [1999] eine neue Tastenbelegung namens *OPTI* (5x7-Layout, siehe Abb. 64). Die Qualität der entstehenden Tastatur, sowie der Standard-

QWERTY-Tastatur, werden mithilfe eines Modells basierend auf *Fitts' Law* (siehe Abschnitt 2.4.5) berechnet. Die theoretische Geschwindigkeit liegt angeblich 35 % über der der *QWERTY*-Tastatur. Eine 5x6-Version von *OPTI* (siehe Abb. 66) wird zusammen mit den Ergebnissen einer praktischen Studie in [Zhang, 1998] vorgestellt.

Zwei alternative Tastaturen namens *Hooke's Keyboard* und *Metropolis* werden in [Zhai et al., 2000] vorgestellt. Dabei verwenden die Autoren im ersten Fall ein Modell, das Zugfedern zwischen den Buchstaben simuliert, im zweiten Fall einen *Random-Walk*-Ansatz. Beide Ansätze, die eine Tastenbelegung im Gegensatz zum Vorgehen aus [MacKenzie u. Zhang, 1999] auf systematischerem Weg erzeugen, finden sich auch in [Hunter et al., 2000]. Die entstehenden Belegungen verfügen jedoch über keine geordnete optische Struktur, wie man sie normalerweise von Tastaturen gewohnt ist. Mittels *Fitts' Law* werden in [Zhai et al., 2000] außerdem sieben Tastaturen (*QWERTY*, *Chubon*, *FITALY*, *OPTI*, *OPTI II*, *Metropolis* und *Hooke's*) bewertet und verglichen. Dabei liefert die *Metropolis*-Tastatur das beste Ergebnis und ermöglicht eine 40 % schnellere Texteingabe als die Standardtastatur. Darüber hinaus korrigieren Zhai et al. [2000] einen Fehler in der Berechnung der absoluten Geschwindigkeit in [MacKenzie u. Zhang, 1999]. Leshner u. Moulton [2000] verwenden ein 2-opt-Verfahren, um zwei effizientere Tastenbelegungen zu erhalten (siehe u.a. Abb. 60). Die beiden zugrunde gelegten physikalischen Layouts sind das einer Standard-Computertastatur (mit fixierter Leertaste; ohne Sondertasten, -zeichen und Umlaute) und ein Layout im *FITALY*-Design (5x6-Layout, 26 Tasten und 2 größere Leertasten). Die Reduktion des Schreibaufwandes bewegt sich laut den Autoren im Bereich von etwa 30 %. Ähnlich zu [MacKenzie et al., 1999] werden in [Zhai et al., 2002a] sechs verschiedene Tastaturen (*QWERTY*, *Square alphabetic*, *OPTI*, *FITALY*, *Chubon*, *Lewis-Kennedy-LaLomia*) hinsichtlich ihrer Eingabegeschwindigkeit basierend auf *Fitts' Law* (siehe auch Abschnitt 8.1) bewertet. Die beiden Methoden aus [Zhai et al., 2000] und deren Ergebnisse werden nochmals vorgestellt und der *Random-Walk*-Ansatz um ein "alphabetisches Tuning" erweitert, welches Anfängern die Buchstabensuche auf der Tastatur erleichtern und damit die Erlernbarkeit begünstigen soll. Laut Zhai et al. [2002b] wurden für die Berechnung der Eingabegeschwindigkeit mittels *Fitts' Law* jedoch falsche Parameter benutzt, weshalb die absoluten Werte der Schreibgeschwindigkeit zu niedrig sind. Dies wird zusammen mit einer Studie zur Lernfähigkeit von Tastaturen in [Zhai et al., 2002b] korrigiert.

Eine Gegenüberstellung von tasten- und stiftbasierter Eingabe, verschiedene Ziele der Tastaturoptimierung und grundsätzliche Punkte, welche bei der Evaluation zu beachten sind, finden sich in [MacKenzie u. Soukoreff, 2002b]. Analog zu oben genannter Literatur stellen die Autoren einen Vergleich der bekanntesten, bis dato verfügbaren Tastenbelegungen an, wobei die Empfehlung für die Parameterwerte aus [Zhai et al., 2002b] noch nicht eingearbeitet wurde.

Raynal u. Vigouroux [2005] und Brewbaker [2008] bedienen sich evolutionärer Algorithmen bei der Suche nach einer besseren Tastatur. Wie einige Arbeiten vor ihm verwendet Brewbaker [2008] *Fitts' Law*. Die Werte der verwendeten Parametern stimmen aber auch hier nicht mit

den in [Zhai et al., 2002b] vorgeschlagenen überein. Die verwendeten Bigrammhäufigkeiten stammen aus zwei Quellen, den Werken Shakespeares und einem Linux Kernel Korpus. Als Vorlagen für das physikalische Layout dienen ähnlich zu [Leshner u. Moulton, 2000] ein gitterförmiges *QWERTY-like*-Design und das *5x6-FITALY-like*-Design.

Eine Arbeit, die sich der Aufgabe der Tastaturoptimierung zu Beginn auf mathematischen Wege nähert ist [Li et al., 2006]. Die Autoren formulieren zu diesem Zweck ein ganzzahliges Optimierungsproblem mit dem Ziel die erwarteten Kosten in Form der Strecke oder Zeit pro Tippvorgang zu minimieren. Das Problem erweist sich aufgrund der enormen Anzahl an Variablen als zu aufwendig für eine exakte Lösung, weshalb man sich einer Hybridheuristik aus der *2-opt*-Methode und dem *simulated-annealing*-Verfahren bedient. Angewendet wird diese Heuristik auf drei Tastaturformen (*QWERTY-like*-Design, *5x6-FITALY-like*-Design und ein quasi-kreisförmiges Design). Das Ergebnis sind Tastenbelegungen, welche eine etwa 25 % schnellere Texteingabe als die Standardtastatur erlauben (siehe u.a. Abb. 59).

Dunlop u. Levine [2012] machen es sich zur Aufgabe, neben der Minimierung der vom Finger zurückzulegenden Strecke eine Verbesserung der Autokorrektur und die Maximierung der Vertrautheit mittels einer *QWERTY*-Ähnlichkeitsfunktion anzustreben. Das Ergebnis ihrer Forschung ist eine rechteckige, gitterförmige Tastatur, die neben einer geringeren Fehleranzahl die Eingabeschwindigkeit verbessert. Die Ähnlichkeit zur *QWERTY*-Tastatur könnte die Textübertragung für Anfänger angenehmer gestalten und die Erlernbarkeit verbessern. Letzteres stellen auch Smith u. Zhai [2001] in den Mittelpunkt ihrer Forschung und erhalten so eine alphabetisch geordnete Tastenbelegung mit erhöhter erzielbarer Schreibleistung.

Eine Arbeit mit dem Ziel Tastaturen bezüglich ihrer Eignung für mehrere Sprachen zu verbessern ist [Bi et al., 2012]. In Anlehnung an [Zhai et al., 2002a] wird ein *Random-Walk*-Algorithmus verwendet. Als Ergebnis dient vor allem die Feststellung, dass eine bezüglich mehreren einzelnen Sprachen verbesserte Tastenbelegung im vorliegenden Fall ohne große Leistungseinbußen auch als internationales Layout eingesetzt werden kann.

Dell'Amico et al. [2009] versuchen sich in einer Klassifizierung der verschiedenen Probleme einer optimalen Tastatur (single-finger versus n -finger). Im Anschluss wird die Suche nach einer optimalen Tastatur für die Bedienung mit einem Finger oder einem Stift auf ein quadratisches Zuordnungsproblem zurückgeführt (siehe auch [Burkard u. Offermann, 1977]) und Heuristiken beschrieben, die sich bei der Anwendung auf Probleme dieser Klasse bewährt haben.

Andere Forschungsrichtungen in dieser Phase widmeten sich alternativen Eingabemethoden, wie sensorbestückten Handschuhen und projizierten Tastaturen (siehe [Kölsch u. Turk, 2002] [Du et al., 2005]) oder verwendeten heuristische Methoden für die Verbesserung der Standard-Computertastatur (siehe [Light u. Anderson, 1993] [Wagner et al., 2003]). Magnien et al. [2004] und Al Faraj et al. [2009] forschten an der Fehlerreduktion und der schnelleren Erlernbarkeit der einhändigen Touchscreen-Texteingabe mithilfe von optischen Hinweisen oder der Veränderung der Tastengröße während des Schreibflusses. Einen Vergleich berührungsempfindlicher Geräte

mit herkömmlichen Mobiltelefonen findet man in [Curran et al., 2005].

4. Phase

Die vorerst letzte Phase begann mit dem weltweiten, großen Erfolg von Smartphones. Die Schreibweise auf modernen Smartphones unterscheidet sich deutlich von der auf Schreibmaschinen- oder Computertastaturen, welche beidhändig mit zehn Fingern bedient werden, und der auf numerischen oder den ersten virtuellen Tastaturen, welche mit einem Finger oder einem Stift bedient wurden. Vor allem Lagesensoren machen es möglich, das Gerät in der Praxis sowohl horizontal als auch vertikal zu verwenden. Letzteres gestaltet den Übergang zur vorherigen Phase fließend. Offensichtlich können bis dato entwickelte mathematische Modelle die Schreibweise mit zwei Daumen nur unzureichend beschreiben, weshalb man daran interessiert ist, diese Art der Texteingabe zu erforschen und verbesserte Tastaturen dafür zu entwickeln.

Die Arbeit von Soukoreff [2002] widmet sich sowohl der Bedienung mit einem Finger als auch der Bedienung mit beiden Daumen. Manche Teile der Arbeit, welche sich mit einhändiger Bedienung auseinandersetzen, findet man in [MacKenzie u. Soukoreff, 2002b] wieder. Außerdem entwickelt Soukoreff [2002] ein Modell für das Schreiben mit beiden Daumen und beschäftigt sich mit dessen Robustheit. Darüber hinaus untersucht er den Einfluss, den unterschiedliche Verwendungsarten der Leertaste einer *mini-QWERTY*-Tastatur auf die Schreibgeschwindigkeit haben. Eine gekürzte und speziell auf das neue 2-Daumen-Modell abgestimmte Version der Arbeit wurde in [MacKenzie u. Soukoreff, 2002a] veröffentlicht.

Clarkson et al. [2007] überarbeiten das oben genannte Modell. Im Gegensatz zur Arbeit von MacKenzie u. Soukoreff [2002a] basiert ihr Modell auf der Annahme, dass es Tasten gibt, die von beiden Daumen bedient werden, anstatt das gegebene Tastaturlayout strikt aufzuteilen. Sowohl Soukoreff [2002] als auch Clarkson et al. [2007] bleiben die Antwort auf die Frage nach einer optimalen Tastatur schuldig.

Eine weitere Arbeit, die sich dem Schreiben mit beiden Daumen widmet, ist [Jüngst, 2008]. Der Autor entwickelt mithilfe eines Evolutionsalgorithmus eine, speziell für die *Strokes*-Eingabetechnik, optimierte Tastatur. Bei dieser Form der Texteingabe gleitet der Daumen über die Tasten anstatt diese anzutippen. Nach einer langwierigen Übungsphase stellt der Autor eine zweifach schnellere Geschwindigkeit in Aussicht als auf einer *mini-QWERTY*-Tastatur.

Die in [Li et al., 2011] vorgestellte Tastatur ist ein Exot im Bereich der virtuellen Tastaturen, da sie nur aus einer Tastenzeile besteht und damit den Vorteil hat, dass sie auf kleineren Geräten kaum Bildschirmfläche bedeckt. Da die Tasten mehrere Buchstaben beherbergen, handelt es sich wie bei numerischen Tastaturen um eine uneindeutige Eingabe, weshalb der Nutzer in einem zweiten Schritt das gewünschte Wort auswählen muss. Die Autoren prognostizieren eine Verdopplung der Eingabegeschwindigkeit im Vergleich zu herkömmlichen virtuellen Tastaturen.

Ein interessanter Forschungsansatz, welcher die Lagesensoren eines Smartphones nutzt um die aus der Bewegung des Benutzers resultierenden Schreibungenauigkeiten auszugleichen, findet sich in [Goel et al., 2012]. Die Anzahl der Tippfehler eines gehenden Benutzers ließ sich in der zugehörigen Studie um 45 % reduzieren.

Großes Presse-Echo rief die Vorstellung einer neuen virtuellen Tastatur zur verbesserten Eingabe mit beiden Daumen (siehe [Oulasvirta et al., 2013]) durch Forscher des Max-Planck-Instituts für Informatik in Saarbrücken hervor. Ihr Weg zur verbesserten 2-Daumen-Tastatur umfasst eine Haltungsstudie, ein Modell für die Bewegungen der Daumen, die Optimierung mittels eines heuristischen Hybridansatzes, einen Korrekturalgorithmus und die Evaluation. Als Ergebnis präsentieren sie *KALQ* (siehe Abb. 63), eine zweigeteilte Tastatur, die in der anschließenden Studie bis zu 34 % besser abschneidet als die *QWERTY*-Tastatur.

Zusammenfassung

Zusammengefasst kann man sagen, dass sich ein Großteil der vorhandenen Literatur mit Tastaturen beschäftigt, die entweder mit einem oder mit zehn Fingern bedient werden (siehe Tabelle 1). Der Grund dafür ist, dass sich die Bedienung eines Mobilgeräts mit beiden Daumen erst in den letzten Jahren verbreitete. Wie auch Dell'Amico et al. [2009] bestätigen, waren es bisher vor allem Wirtschaftswissenschaftler, die sich für das Problem einer optimalen Tastatur interessierten, was häufig dazu führte, dass eine mathematische Formulierung des Problems fehlt. Naturgemäß waren deshalb Heuristiken das Mittel der Wahl, um sich der Lösung des Problems anzunähern.

Eine Klassifikation ausgewählter Literatur auf Grundlage der Anzahl am Schreibvorgang beteiligter Finger und verwendeter Methoden zur Verbesserung der Tastenbelegung findet sich in Tabelle 1. Im Mittelpunkt der Auswahl stand die Eingabe eines Buchstabens durch bloßes Antippen sowie die Erzeugung und Präsentation einer Tastatur mit genau einer Taste pro Buchstabe (siehe auch Abschnitt 2.3.1).

Bezüglich der Eingabegeschwindigkeit, die in den häufigsten Fällen mit *Fitts' Law* berechnet wird, lässt sich feststellen, dass den Werten kaum zu trauen ist. Dies liegt vor allem an der fehlenden Standardisierung. So ist man sich uneinig in der Textkorpus- und Parameterwahl, wobei aus Ersterer die Wahrscheinlichkeiten der Buchstaben und ihrer Kombinationen gewonnen werden. Beides beeinflusst die Berechnung der Schreibgeschwindigkeit, weshalb in der Literatur unterschiedlichste Ergebnisse für ein und dieselbe Tastatur präsentiert werden.

Tabelle 1: Klassifikation ausgewählter Literatur aus dem Bereich der Tastaturentwicklung hinsichtlich verwendeter Lösungsmethoden und Anzahl der am Schreibvorgang beteiligten Finger.

Literatur	Finger	Lösungsmethoden
[Burkard u. Offermann, 1977]	10	2-opt, 3-opt
[Wagner et al., 2003]	10	ant colony algorithm
[Light u. Anderson, 1993]	10	simulated annealing
[Li et al., 2006]	1	simulated annealing
[Getschow et al., 1986]	1	greedy algorithm
[Leshner u. Moulton, 2000]	1	2-opt
[Lewis et al., 1999a]	1	minimal-connected-network approach
[Dunlop u. Levine, 2012]	1	pareto front optimization
[Bi et al., 2012]	1	metropolis random walk
[Brewbaker, 2008]	1	evolutionary algorithm
[Zhai et al., 2002a], [Zhai et al., 2000]	1	spring simulation, metropolis rand. walk
[MacKenzie u. Zhang, 1999]	1	trial-and-error
[Oulasvirta et al., 2013]	2	gradient-descent, simulated annealing

2.3 Voraussetzungen und Thematik der vorliegenden Arbeit

2.3.1 Voraussetzungen und Konventionen

Um den Rahmen der Arbeit nicht zu sprengen, werden grundlegende Fähigkeiten eines Mathematikstudiums sowie fortgeschrittene Kenntnisse aus der linearen und ganzzahligen Optimierung vorausgesetzt und nicht weiter erläutert. Zum Verständnis von Kapitel 7 sind Erfahrungen mit dem Programm *FICO IVE Xpress* hilfreich. Darüberhinaus sollen folgende Konventionen und Voraussetzungen im gesamten Verlauf der Arbeit gelten.

Die vorliegende Arbeit widmet sich der Optimierung von Tastenbelegungen gegebener Tastaturlayouts (siehe Abschnitt 2.4.3). Das bedeutet, dass eine Optimierung oder tiefere Diskussion bezüglich der Positionierung, sowie der Form und Größe der Tasten nicht stattfindet. Ebenso wenig werden Ansätze weiterverfolgt, welche eine Mehrfachverwendung von Buchstaben (wie bspw. die *ENTI-key++*-Tastatur, siehe [Barkow, 2007]), eine Mehrfachbelegungen von Tasten (wie bspw. die *1-line*-Tastatur aus [Li et al., 2011]) oder verschiedenen Tastaturebenen (wie bspw. die *NEO*-Belegung, siehe [Behrens, 2004]) voraussetzen. Untersucht werden also Belegungen virtueller, vollwertiger Tastaturen, welche durch ein Antippen der Tasten bedient werden. Vollwertig bedeutet in diesem Zusammenhang, dass jede der Tasten höchstens ein Zeichen beherbergt und jedes Zeichen genau einmal auf den berechneten Tastaturen zu finden sein soll. Alternative Eingabemethoden, wie beispielsweise *Swype*

(siehe [Swype-Inc., 2010]) oder gestenbasierte Schreibweisen (siehe bspw. [Jüngst, 2008], [MacKenzie u. Soukoreff, 2002b]) sind nicht Gegenstand der Arbeit.

Die Menge der zu verteilenden Zeichen umfasst dabei die 26 Standardbuchstaben des deutschen Alphabets. Somit enthalten die Ausgangsdaten für die Entwicklungen neuer Tastenbelegungen keine Sonderzeichen, es wird nicht zwischen Groß- und Kleinschreibung unterschieden und Umlaute werden als Kombination der Standardbuchstaben interpretiert. Symbole sowie die Beschaffenheit und Position der Leertaste werden genauso wenig betrachtet wie Sonder- und Funktionstasten.

Die Begriffe virtuelle Tastatur, Smartphone-Tastatur und Touchscreentastatur werden synonym verwendet und bezeichnen eine Software-basierte Tastatur, welche bei Bedarf auf dem berührungssensitiven Bildschirm eines Smartphones eingeblendet wird. Es wird zwischen den Begriffen Tastatur, Tastaturlayout und Tastenbelegung unterschieden. Eine Tastatur setzt sich zusammen aus einem Layout (oder Design), wozu unter anderem die Anordnung und die Beschaffenheit der Tasten zählt, und der Tastenbelegung. Die Belegung bezeichnet ausschließlich welche Zeichen auf einer Taste zu finden sind.

Länderspezifische Konfigurationen der Standardtastenbelegung, wie beispielsweise das deutsche *QWERTZ* oder das französische *AZERTY*, werden nicht gesondert betrachtet. Sie alle sind unter dem Oberbegriff der *QWERTY*-Tastenbelegung zusammengefasst.

Im Verlauf der Arbeit wird zwischen der Textübertragung mithilfe eines Fingers (oder ähnlichem Hilfsmittel) in vertikaler Geräteposition, was in der vorliegenden Arbeit oftmals als vertikaler Fall bezeichnet wird, und dem Schreiben in horizontaler Position unterschieden. Letzteres erfolgt mithilfe von zwei Fingern, in der Regel den beiden Daumen und wird deshalb als 2-Daumen-Schreibweise oder synonym als beidhändiger oder horizontaler Fall bezeichnet. Dessen Besonderheit im Gegensatz zum vertikalen Fall ist die Verwendung einer gesplitteten Tastatur. Deren Teilung muss dabei nicht notwendigerweise optischer Natur sein, aber in jedem Fall findet eine Zuordnung jeder Taste zu einer Seite und damit zu einem Finger statt. Obwohl es möglich wäre, dass bestimmte Tasten von beiden Daumen angesteuert werden können (siehe [Clarkson et al., 2007]), wird diese Möglichkeit nicht betrachtet, da sie in der Realität kaum Verwendung findet. Darüberhinaus wäre es denkbar, die disjunkte Aufteilung der Tasten völlig frei vorzunehmen, jedoch geht man aus praktischen Gründen davon aus, dass sich die abdeckenden Tastaturbereiche der beiden Daumen nicht überschneiden. Analog zu den in der Praxis verwendeten Layouts wird in der vorliegenden Arbeit auf eine optische Teilung der Tastatur verzichtet.

2.3.2 Inhalt und Aufbau

Inhalt und Ziel der vorliegenden Arbeit ist es, dem Leser das Potential aufzuzeigen, das in einer Überarbeitung der auf Smartphones weit verbreiteten *QWERTY*-Belegung steckt und alterna-

tive, verbesserte Tastenbelegungen für unterschiedliche Schreibweisen und Gerätepositionen zur Verfügung zu stellen.

In diesem Zusammenhang stellt sich die Frage, welche Charakteristika eine gute Tastenbelegung ausmachen. Als Ziele der Optimierung werden dabei insbesondere zwei Eigenschaften identifiziert, nämlich die Reduzierung des erwarteten Schreibaufwands und der erwarteten Schreibdauer. Die Minimierung des Schreibaufwands entspricht der Minimierung der erwarteten zurückzulegenden Strecke, welche die am Schreibvorgang beteiligten Finger zurücklegen müssen, um einen Text durchschnittlicher Länge zu übertragen (siehe Kapitel 3, 4 und 5). Die Minimierung der erwarteten benötigten Zeit für die Übertragung eines durchschnittlichen Textes findet sich in Kapitel 6.

Bezüglich der Gerätepositionen wird zunächst zwischen vertikalen (Kapitel 3) und horizontalen (Kapitel 4) Anwendungen unterschieden, bevor der Fokus in Kapitel 5 auf sogenannte Hybrid-Anwendungen gelegt wird, welche beide Positionen in sich vereinen.

Abhängig von der Position des Gerätes existieren verschiedene Schreibweisen, die der Reihe nach vorgestellt und untersucht werden und je nach Realitätsnähe auf Optimierungsprobleme unterschiedlicher Schwierigkeit führen (siehe bspw. Abschnitt 3.1 und 3.2).

Besonderer Wert wird in der gesamten Arbeit auf die Modellierung und die Formulierung der Optimierungsprobleme gelegt. Ansätze und Verfahren, welche die Lösbarkeit der resultierenden Probleme (aus zeitlicher Sicht) ermöglichen, werden an geeigneten Stellen beschrieben und diskutiert (siehe Abschnitt 3.2.3 und 3.2.4).

Kapitel 7 widmet sich der Implementierung der Optimierungsprobleme in eine geeignete Lösungssoftware (siehe auch Abschnitt 2.4.6) und beschreibt die Fähigkeiten des resultierenden Programms.

Im Anschluss folgt ein Vergleich mit alternativen Tastenbelegungen und Tastaturen (siehe Kapitel 8) sowie eine Diskussion über die Ergebnisse der Arbeit und ihren zugrundeliegenden Voraussetzungen (siehe Kapitel 9). Die Arbeit endet mit einem Ausblick, welche Fragen im Mittelpunkt zukünftiger Projekte im Bereich der Tastaturoptimierung stehen könnten (siehe Kapitel 10).

Insbesondere der Fokus auf den mathematischen Problemen stellt einen bedeutenden Unterschied zur bisher verfügbaren Literatur dar. Zum großen Teil bedient man sich dort Heuristiken, um bessere Tastaturen zu erzeugen (siehe Abschnitt 2.2). Ein weiterer Punkt, der diese Arbeit positiv von anderen Publikationen abgrenzt, ist die Verwendung von exakteren Parametern im Bereich der Minimierung der Schreibdauer und der Bestimmung der Schreibgeschwindigkeit (siehe Abschnitt 2.4.5). Darüber hinaus erweitert die vorliegende Arbeit das überschaubare Angebot an Literatur im Bereich der Textübertragung mit zwei Fingern auf Smartphones. Während ein Großteil der verfügbaren Tastaturen und Belegungen für die Eingabe von englischem Text konzipiert ist, eignen sich die im Verlauf der Arbeit präsentierten Tastenbelegungen für die Übertragung deutscher Texten. Die Erzeugung einer Tastatur, welche für vertikale und horizon-

tale Anwendungen geeignet ist, stellt ein Novum im Bereich der Tastaturforschung dar.

2.3.3 Grundlegende Begriffe und Notationen

Aus Gründen der Übersichtlichkeit und Verständlichkeit werden folgende Notationen sofern nicht explizit anders angegeben in der gesamten Arbeit verwendet.

Bezeichne mit $m \in \mathbb{N}$ die Anzahl der Tasten und mit $n \in \mathbb{N}$ die Anzahl der zu verteilenden Buchstaben. Dann ist $[n] = \{1, \dots, n\}$ die Menge der zu verteilenden Zeichen und $[m] = \{1, \dots, m\}$ die Menge der Tasten. Eine Tastenbelegung lässt sich damit beschreiben als die Verteilung von n Zeichen auf m Tasten. In allen Kapiteln der vorliegenden Arbeit gilt dabei $m \geq n$ und in ausgewählten, gekennzeichneten Abschnitten $m = n$.

Im gesamten Verlauf der Arbeit dienen i und k als Buchstabenindizes, das heißt es gilt $i \in [n]$ und $k \in [n]$. Für die Tastenindizes j und l gilt $j \in [m]$ und $l \in [m]$.

Sei X ein lineares, ganzzahliges Optimierungsproblem, dann bezeichne \mathcal{F}_X den Zulässigkeitsbereich des Problems und \mathcal{P}_X das Polytop oder Polyeder, welches durch den Zulässigkeitsbereich der LP-Relaxierung beschrieben wird. Desweiteren ist \mathcal{L}_X die Menge der optimalen Lösungen des Problems und f_X dessen Zielfunktion. Im Verlauf der Arbeit werden $x \in \mathcal{F}_X$ auch als zulässige Punkte und $x^* \in \mathcal{L}_X$ als Lösungen bezeichnet. Taucht in der Arbeit die Notation $\dim(\mathcal{P}_X)$ auf, so bezeichnet das die Dimension des Polytops oder Polyeders \mathcal{P}_X und ist definiert als die Dimension seiner affinen Hülle.

Darüber hinaus sei $\langle \cdot, \cdot \rangle$ die Notation des Standardskalarprodukts.

Wird bei der Untersuchung von Optimierungsproblemen die Anzahl der Nebenbedingungen genannt, so fließen Ganzzahligkeits- und Positivitätsbedingungen nicht in die Zählung ein.

An manchen Stellen der vorliegenden Arbeit wird von Unlösbarkeit von Optimierungsproblemen gesprochen, was sich ausschließlich auf die Dauer der Optimierung bezieht und somit aus zeitlicher Sicht zu verstehen ist.

2.4 Daten und deren Beschaffung

Bei der Entwicklung optimaler Tastenbelegungen, welche nicht nur theoretisch, sondern auch in der Praxis gute Ergebnisse liefern sollen, ist die Qualität der zugrundeliegenden Daten von großer Bedeutung. Um die Qualität zu gewährleisten, wurde darauf geachtet, möglichst realistische Daten zu beschaffen und an Stellen, an denen Annahmen und Einschränkungen nötig waren, nahe an der Realität zu bleiben.

Die kompletten Daten, die den später präsentierten Lösungen zugrundeliegen, finden sich in Anhang A. Dort finden sich neben realistischen Instanzen auch Testinstanzen kleinerer Dimension, die erstellt wurden, um die Lösungsdauer der formulierten Probleme für realistische Größenordnungen vorherzusagen. Sämtliche Instanzen sind identisch aufgebaut (siehe auch

Abschnitt 7.1) und in *.dat-Files gespeichert, um ein problemloses Einlesen in die Optimierungssoftware zu ermöglichen.

2.4.1 Buchstaben- und Bigrammwahrscheinlichkeiten

Um Tastenbelegungen für das Schreibverhalten eines durchschnittlichen Smartphone-Nutzers zu optimieren, ist es nötig einen Textkorpus hinsichtlich der Wahrscheinlichkeiten der auftretenden Buchstaben oder Buchstabenkombinationen zu analysieren. Im Mittelpunkt steht bei der Auswahl des Korpus dessen Aktualität, dessen Herkunft und dessen Umfang. Seine Aktualität ist wichtig, um ein möglichst aktuelles Schreibverhalten der Nutzer zu erfassen. Bezüglich der Herkunft des Textes ist es wünschenswert, ihn aus einer Onlinequelle zu beziehen, da die Texteingabe auf Smartphones häufig im Internet Anwendung findet. Eine ausreichende Korpusgröße ist nötig, um die Repräsentativität sicherzustellen. Bei der Zusammensetzung des Korpus wird auf literarische Werke verzichtet, da sie oftmals älter sind und in der Regel nicht Gegenstand der Textübertragung auf Smartphones. Der verwendete Korpus wird von der Universität Leipzig bereitgestellt und setzt sich aus 10.000 deutschen Sätzen zusammen, welche zufällig aus verschiedenen online-Nachrichtenportalen entnommen wurden. Mit insgesamt 1.012.851 Buchstaben sollte dieser Korpus damit den Aufbau eines zu übertragenden deutschen Textes und somit das Schreibverhalten der Nutzer realistisch abbilden ([Universität Leipzig, 2010]).

Um die in der Optimierung verwendeten Häufigkeiten von Buchstaben oder Buchstabenkombinationen aus obigem Korpus auszulesen, wird das Programm *WordCreator.exe* genutzt ([Trost, 2013]). Wie in Abschnitt 2.3 beschrieben, sollen beispielsweise Sonderzeichen keine Rolle in der Optimierung spielen. Somit ist es zunächst nötig, den Textkorpus für das Auslesen vorzubereiten. Die Vorverarbeitung besteht im Wesentlichen aus der Interpretation der Umlaute als ihre Kombination aus Standardzeichen, der Entfernung von Sonder- und Leerzeichen und der Transformation von Großbuchstaben. Alle Textbearbeitungsschritte wurden in *ApacheOpenOffice v. 3.4.0* ausgeführt. Der resultierende Text wird im Anschluss an das Zählprogramm übergeben, welches die gewünschten Daten ausgibt. Die Einstellungen wurden so festgelegt, dass die Wahrscheinlichkeiten in Prozent ausgegeben und auf zwei Nachkommastellen gerundet werden.

2.4.2 Textlänge

Die Anzahl B der Buchstaben des zu übertragenden Textes, welche auch als Textlänge bezeichnet wird, spielt für die Berechnung und Minimierung der zurückzulegenden Gesamtdistanz eine wichtige Rolle, weshalb eine realistische Parameterwahl dabei im Mittelpunkt steht.

Dazu ist festzustellen, dass das Schreiben langer Texte mit mehreren hundert Wörtern auf Smartphones selten vorkommt, was insbesondere an der vergleichsweise geringen Gerätegröße und daraus resultierenden Nachteilen in der Handhabung liegen dürfte. Ein weiterer Grund

ist, dass ein Großteil der Textübertragung auf Smartphones in den Bereichen Kommunikation und Internetnutzung stattfindet, in denen kurze Sätze und Textfragmente Standard sind.

Ein Hauptbestandteil der Kommunikation sind SMS, die je nach Nutzergruppe eine durchschnittliche Länge von 93.5 oder 106.77 Buchstaben aufweisen [Dittmann et al., 2007]. Die Ergebnisse der vorliegenden Arbeit sollen, unabhängig von der Zusammensetzung der Gruppe, für alle Smartphone-Nutzer geeignet sein, weshalb B als der gerundete Mittelwert der obigen Längen gewählt wird, also $B = 100$. Auch im Bereich der Internetnutzung, die sich vor allem aus Statusmeldungen und Kommentaren zusammensetzt, erscheint diese Wahl sinnvoll, da beispielsweise für eine *Twitter*-Nachricht eine Obergrenze von 140 Zeichen gilt.

Ist man an Tastenbelegungen interessiert, die für eine alternative Parameterwahl optimiert sind, so bietet das Programm *VirtualKeyboardOptimization.mos* passende Einstellungsmöglichkeiten (siehe Kapitel 7). Eine Betrachtung der Auswirkungen einer Änderung von B findet sich in Kapitel 9.

2.4.3 Tastaturlayouts

Allgemein ist ein Tastaturlayout durch die Anzahl und Form, sowie Größe und Position seiner Tasten gegeben. Wie in Abschnitt 2.3 angekündigt, werden vorgegebene Tastaturlayouts verwendet, die über identische, quadratische Tasten verfügen und vollständig innerhalb eines bestimmten, zulässigen Bereichs liegen. Da die Kantenlänge der Tasten nur als Faktor im Zielfunktionswert auftritt, wird diese der Einfachheit halber gleich einer Längeneinheit (1 LE) gewählt. Orientiert man sich an der Größe der in der Praxis verwendeten Tastaturen auf Smartphones, so ist ein zulässiger Tastaturbereich mit einer Höhe von 4 LE auf eine Breite von 10 LE gerechtfertigt (graues Hilfsraster in den Abbildungen). Innerhalb des zulässigen Bereichs liegen die Layouts horizontal zentriert am oberen Rand. Diese Position basiert auf der Ausrichtung von realen Tastaturen und bietet im unteren oder seitlichen Bereich Platz für Sonder- und Funktionstasten. Den im Verlauf der Arbeit präsentierten Ergebnissen liegen folgende Tastaturlayouts zugrunde.



Abbildung 1: QWERTY-like-Layout

Das sogenannte *QWERTY-like*-Layout (siehe Abb. 1) mit zeilenweise 10,9,7 Tasten wurde ausgewählt, um der Gewohnheit der Nutzer Rechnung zu tragen, indem es eine ähnliche Tastenanordnung wie die derzeit verwendete Standardtastatur aufweist. Es bietet Platz für die 26 Standardbuchstaben des deutschen Alphabets. Die Leertaste und etwaige Sonder- und Funktionstasten könnten in einem späteren Schritt unterhalb des Layouts positioniert werden. Dem zweiten Layout-Kandidaten (siehe Abb. 2) liegt eine 3×9 -Gitterstruktur zugrunde, was ihm im Gegensatz zum *QWERTY-like*-Layout eine regelmäßigere Tastenanordnung beschert.

T1	T2	T3	T4	T5	T6	T7	T8	T9	
T10	T11	T12	T13	T14	T15	T16	T17	T18	
T19	T20	T21	T22	T23	T24	T25	T26	T27	

Abbildung 2: 3×9 -Layout

Dieses Design verfügt über 27 Tasten, was bei der Verteilung der 26 Standardbuchstaben dazu führt, dass eine Taste frei bleibt. Es wäre denkbar, dass diese später für Sonderzeichen oder als Leertaste verwendet wird. Die verbleibenden Funktionstasten würden in der Praxis wahrscheinlich in die vierte Zeile gelegt werden.

Da im Verlauf der Arbeit Tastaturen gesplittet werden, ist der Wunsch nach einer symmetrischen Aufteilbarkeit gerechtfertigt. Dies funktioniert jedoch nicht mit den bisherigen Layouts, weshalb zwei weitere Kandidaten betrachtet werden.

T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
T11	T12	T13	T14	T15	T16	T17	T18	T19	T20
T21	T22	T23	T24	T25	T26	T27	T28	T29	T30

Abbildung 3: 3×10 -Layout

Das 3×10 -Layout aus Abbildung 3 unterscheidet sich bis auf die gerade Spaltenanzahl, der daraus resultierenden symmetrischen Splittbarkeit und der größeren Tastenzahl nicht vom 3×9 -Layout.

Das vierte verwendete Layout (siehe Abb. 4) ist mit seiner zugrundeliegenden 4×8 -Gitterstruktur

weniger breit, doch dafür höher. Analog zum 3×9 - und 3×10 -Layout bietet es aufgrund seiner Tastenanzahl, welche die Buchstabenanzahl übersteigt, mehr Belegungsmöglichkeiten als das *QWERTY-like*-Layout. Dieses Layout verfügt über sechs Tasten, die nicht von Buchstaben belegt sind und beispielsweise Platz für Sonderzeichen bieten. Darüberhinaus benötigte Tasten würden an den Seiten des Layouts platziert werden.

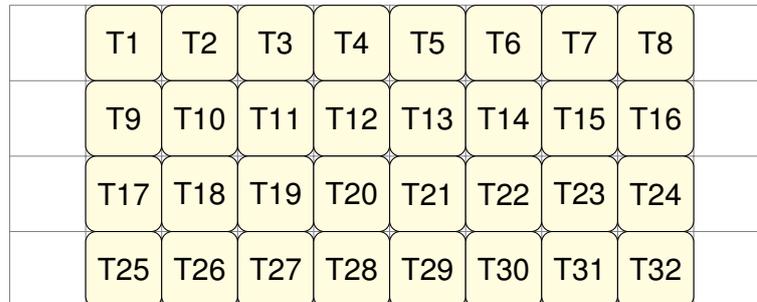


Abbildung 4: 4×8 -Layout

Neben der Anordnung und der Anzahl der Tasten benötigt man Informationen über die Teilung des Layouts, also die eindeutige Verteilung aller Tasten auf die Seiten L und R . Dafür dienen die Mengen $T_L \subseteq [m]$ und $T_R \subseteq [m]$, für die gilt $T_L \dot{\cup} T_R = [m]$.

Aufgrund der weltweit größeren Zahl ($\approx 90\%$) an Rechtshändern ([Cashmore et al., 2008]) wird davon ausgegangen, dass der Benutzer der Touchscreentastatur Rechtshänder ist. Damit gilt im vertikalen Fall $T_R = [m]$, da ein Finger der rechten Hand die gesamte Schreibarbeit übernimmt und das Layout nicht gesplittet wird (siehe Abb. 5).

Um im horizontalen Fall keinen der beiden Finger zu überbeanspruchen, wird jede Tastenzeile eines Layouts mittig geteilt (siehe Abb. 6). Aufgrund der Neigung anspruchsvolle Aufgaben mit der dominanten Hand zu erledigen ([Cashmore et al., 2008]) werden Zeilen mit ungerader Tastenzahl zugunsten der rechten Seite geteilt. Bezeichnet Z die Menge der Tastaturzeilen und $T(z)$ die Anzahl der Tasten in Zeile $z \in Z$, dann gilt $T_R = [m] \setminus T_L$, wobei $T_L = \bigcup_{z \in Z} \left\{ j + \sum_{i=1}^{z-1} T(i) \mid j \in \left\{ 1, \dots, \left\lfloor \frac{T(z)}{2} \right\rfloor \right\} \right\}$.

2.4.4 Entfernungen der Tasten

Um eine Tastenbelegung ermitteln zu können, die das Ziel der Distanzminimierung erfüllt, ist man an den Entfernungen innerhalb eines gewählten Layouts interessiert.

Übernimmt ein Finger die gesamte Schreibarbeit, beschränken sich die benötigten Daten auf die Abstände d_{ji} der Tasten untereinander (siehe (2.1)) und die Abstände d_j der Tasten zum Startpunkt des Fingers SP (siehe (2.2)).

Wird das Smartphone mit beiden Händen bedient, hat dies zur Folge, dass es zwei Startpunkte $SP(S) \in \mathbb{R}^2$ mit $S \in \{L, R\}$ gibt und nicht beide Finger dauerhaft beschäftigt sind. Im Verlauf

der Arbeit werden verschiedene beidhändige Schreibweisen modelliert, die sich darin unterscheiden, auf welcher Warteposition $WP(S) \in \mathbb{R}^2$ ein temporär nicht benutzter Finger pausiert. Damit benötigt man im horizontalen Fall neben den Distanzen d_{jl} zwischen den Tasten (siehe (2.1)) und den Entfernungen \tilde{d}_j von den Startpunkten (siehe (2.3)) auch die Distanzen \hat{d}_j zu den Wartepunkten (siehe (2.4)), die von der benutzten Schreibweise abhängen. Unabhängig von der Position des Gerätes werden die zu einer Taste gehörigen Entfernungen dabei standardmäßig vom Tastenmittelpunkt aus und mithilfe der 2-Norm bestimmt.

Bezeichne $M_j \in \mathbb{R}^2$ die Positionen des Mittelpunktes der Taste $j \in [m]$, dann gilt:

$$d_{jl} := \|M_j - M_l\|_2 \quad \forall j, l \in [m] \quad (2.1)$$

$$d_j := \|SP - M_j\|_2 \quad \forall j \in [m] \quad (2.2)$$

$$\tilde{d}_j := \|SP(S) - M_j\|_2 \quad \forall j \in [m] \quad (2.3)$$

$$\hat{d}_j := \|WP(S) - M_j\|_2 \quad \forall j \in [m] \quad (2.4)$$

Um konkrete Werte berechnen zu können, werden die Koordinaten der Warte- und Startpunkte, sowie der Tastenmittelpunkte benötigt. Zu diesem Zweck interpretiert man den zulässigen Tastaturbereich als Ausschnitt des ersten Quadranten, wobei der Ursprung mit der linken unteren Ecke des Bereichs übereinstimmt. Die Positionen der Tastenmittelpunkte können dann anhand der Layouts aus Abschnitt 2.4.3 bestimmt und damit die Werte von d_{jl} leicht berechnet werden. Für die Wahl der Start- und Wartepunkte muss man sich prinzipiell zwischen zwei Möglichkeiten entscheiden. So ist es denkbar, die Positionen SP , $SP(S)$ und $WP(S)$ fix oder in Abhängigkeit der Tastenbelegung zu wählen. Unabhängig davon ist es wünschenswert die Punkte möglichst realistisch zu wählen, da sie die Ergebnisse der Optimierung beeinflussen. Eine Diskussion über den Einfluss der Wahl findet in Kapitel 9 statt.

Aufgrund der Bauweise eines Smartphones fällt die Entscheidung für oder gegen einen fixen vertikalen Startpunkt SP leicht. Da nahezu die gesamte Bedienung des Gerätes über den Touchscreen erfolgt und die Tastatur bei Bedarf von unten eingeblendet wird, kann man davon ausgehen, dass der Startpunkt am oberen Rand des zulässigen Tastaturbereichs liegt. Aufgrund der Diversität der unzähligen Anwendungen, welche eine Wahl zugunsten einer Seite kaum möglich macht, und aufgrund der Eignung einer Tastenbelegung für Rechts- und Linkshänder bei mittiger Startpunktwahl, gilt $SP = (5, 4)$ (siehe Abb. 5).

Um die Sichtbarkeit des Bildschirms nicht zu beeinträchtigen, liegen die horizontalen Startpunkte $SP(S)$ analog zum vertikalen Fall am Rand des zulässigen Tastaturbereichs. Aus Gründen der stabileren Handhabung befinden sie sich am seitlichen Rand in symmetrischer Lage [Oulasvirta et al., 2013]. Aufgrund der besseren Drehfreudigkeit sind sie (aus horizontaler Sicht) tendenziell in Richtung des unteren Bildschirmrandes positioniert, jedoch nicht direkt am un-

teren Bildschirmrand. Letztgenannte Positionierung würde die Stabilität negativ beeinflussen, da bei modernen Smartphones der Touchscreen meist die komplette Vorderseite des Gerätes einnimmt. Insgesamt kann man somit davon ausgehen, dass für die Lage der horizontalen Startpunkte gilt: $SP(L) = (0, 1)$ und $SP(R) = (10, 1)$ (siehe Abb. 6).

Die exakten Positionen der Wartepunkte $WP(S)$ sind, wie oben erwähnt, abhängig vom Realismusgrad der horizontalen Schreibweise und werden für ein sogenanntes semirealistisches Modell gleich den Startpunkten gewählt, also $WP(S) = SP(S)$ (siehe Abschnitt 4.2.1).

Für realistischere Schreibweisen, betrachtet man die Folgen, die eine fixe oder belegungsabhängige Wahl nach sich zieht. Würde man die Punkte $WP(S)$ nicht fix wählen, sondern in Abhängigkeit der Tastenbelegung, so ließen sich die Distanzen \hat{d}_j (siehe (2.4)) nur allgemein, in Abhängigkeit der Optimierungsvariablen angeben. Dies würde dazu führen, dass anstatt reeller Werte, welche eine fixe Wahl liefern würde, nichtlineare Ausdrücke in die Zielfunktion aufgenommen werden müssten. Eine Linearisierung wäre nur auf Kosten der Lösungsdauer durchführbar. Trotzdem kann man der zugrundeliegenden Idee folgen, indem man versucht die Warteposition für feste Tastenbelegungen zu bestimmen. Dafür stellt sich zunächst die Frage, wo der Wartepunkt $WP(S)$ allgemein liegt. Am sinnvollsten erscheint es, für $WP(S)$ den erwarteten anzusteuernenden Punkt des jeweiligen Daumens zu wählen. Bezeichnet $M_j \in \mathbb{R}^2$ den Vektor zum Tastenmittelpunkt von j , dann liefert folgender Ausdruck den erwarteten anzusteuernenden Punkt.

$$WP(S) = \sum_{i=1}^n \sum_{j \in T_S} p_i \cdot M_j \quad \forall S \in \{L, R\} \quad (2.5)$$

Bestimmt man die erwarteten anzusteuernenden Punkte für mehrere Tastenbelegungen und Layouts, dann stellt man fest, dass sich die Punkte für beide Finger erwartungsgemäß in der Mitte der jeweiligen Layouthälfte häufen. Wägt man die minimale Verbesserung hinsichtlich der zurückzulegenden Distanz gegenüber der deutlich schlechteren Handhabbarkeit des resultierenden Problems ab, so spricht alles für eine fixe Wahl der Wartepunkte $WP(S)$ nahe der Mittelpunkte der Layouthälften. Um die Vergleichbarkeit sicherzustellen werden die beiden Wartepunkte identisch und unabhängig von den zugrundeliegenden Layouts gewählt. Aufgrund obiger Erkenntnisse bietet sich dafür der Mittelpunkt der jeweiligen Hälfte des zulässigen Bereichs an, weshalb gilt $WP(S) = MP(S) \forall S \in \{L, R\}$ mit $MP(L) = (2.5, 2)$ und $MP(R) = (7.5, 2)$ (siehe Abb. 6 und Abschnitt 4.2.2).

Die Abbildungen 5 und 6 visualisieren die Wahl der Aufteilung (siehe Abschnitt 2.4.3) und der oben beschriebenen speziellen Punkte am Beispiel des *QWERTY-like*-Layouts. Die blauen Tasten sind dabei dem rechten Finger zugeordnet, die orangen dem linken. Die roten und grünen Punkte stellen die Start- und Mittelpunkte der jeweiligen Tastaturseite dar.

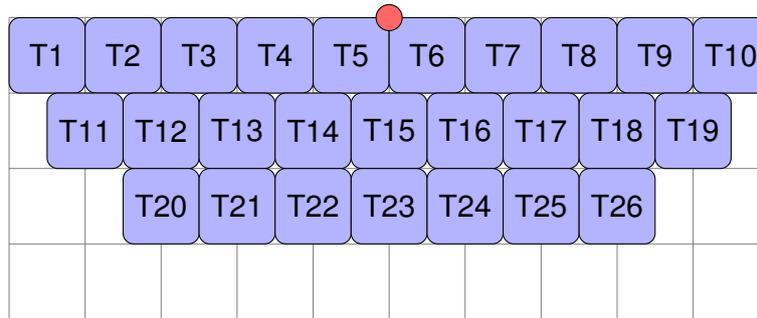


Abbildung 5: Aufteilung und spezielle Punkte des QWERTY-like-Layouts in vertikaler Position.

Diese Farbwahl bleibt mit Ausnahme des Hybrid-Falls, in welchem durch den Positionswechsel des Smartphones eine solche Zuordnung nicht möglich ist, erhalten.

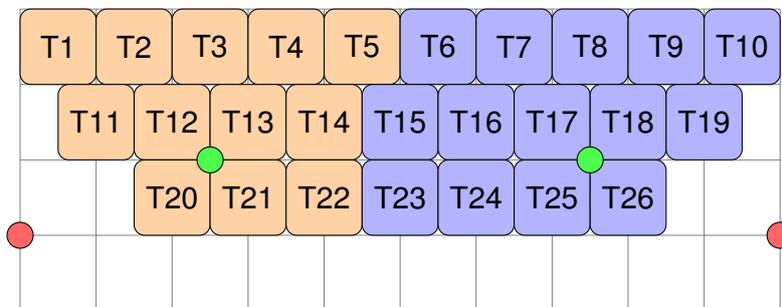


Abbildung 6: Aufteilung und spezielle Punkte des QWERTY-like-Layouts in horizontaler Position.

Welche der Distanzen aus (2.1)-(2.4) genau in die Optimierung einfließen, hängt von der jeweiligen Position des Geräts und der Schreibweise des Benutzers ab und wird deshalb zusammen mit der Wahl der Wartepunkte am Beginn des jeweiligen Kapitels geklärt.

Die konkreten Werte der Distanzen wurden in *ApacheOpenOffice v. 3.4.0* berechnet und finden sich in Anhang A.2.

2.4.5 Bewegungsdauer

Mit den Entfernungen, die dank Abschnitt 2.4.4 aus den Layouts gewonnen werden können, ist man theoretisch in der Lage distanz-minimierende Tastenbelegungen zu berechnen oder Belegungen hinsichtlich der zurückzulegenden Distanz zu bewerten. Für eine Optimierung oder Evaluation hinsichtlich der aufzuwendenden Zeit bedarf es darüberhinaus eines Modells, welches die Zeit $t := t(P, Z) \in \mathbb{R}$ liefert, die man benötigt, um von einem Punkt $P \in \mathbb{R}^2$ zu einem Ziel $Z \in \mathbb{R}^2$ zu gelangen und dieses anzutippen.

Das in diesem Zusammenhang am häufigsten verwendete Modell basiert auf *Fitts' Law*, das erstmals 1954 von Paul Fitts vorgestellt wurde (siehe [Fitts, 1954]). Die Genauigkeit des Modells konnte in mehreren Untersuchungen bestätigt werden (siehe bspw. [MacKenzie, 1989]) und es lautet:

$$t = a + b \cdot \log_2\left(\frac{D(P, Z)}{G(Z)} + c\right) \quad (2.6)$$

Dabei bezeichnet $G(Z) \in \mathbb{R}$ die Größe des Ziels $Z \in \mathbb{R}^2$ und $D(P, Z) \in \mathbb{R}$ dessen Distanz vom Punkt $P \in \mathbb{R}^2$ aus.

Trotz der häufigen Verwendung von (2.6) in der Literatur existiert keine standardmäßige Wahl der Parameter $a, b, c \in \mathbb{R}$, was den Vergleich zwischen Tastenbelegungen erschwert. Über die Wahl von $c = 1$ (sog. *Shannon-Formulierung*) ist man sich im Großen und Ganzen einig, die Werte von a und b unterscheiden sich jedoch zum Teil erheblich. Die Arbeit folgt an dieser Stelle der aktuellsten verfügbaren Empfehlung aus [Zhai et al., 2002b] und man wählt $a = 0,083$ und $b = 1/7,9$.

Es sollte nicht unerwähnt bleiben, dass das Modell nur die Dauer der reinen Bewegung liefert. Die Zeit, welche von einem Anfänger für die Suche nach dem nächsten Buchstaben benötigt wird, ist in diesem Modell nicht berücksichtigt.

Die Gleichung (2.6) ist es auch, die der Berechnung der erzielbaren Schreibgeschwindigkeit auf einer Tastenbelegung zugrunde liegt (siehe Abschnitt 8.1). Aufgrund der Vernachlässigung anderer zeitraubender Faktoren (wie bspw. der Suchzeit) ist der resultierende Wert in der Theorie eine obere Schranke an die Schreibgeschwindigkeit.

2.4.6 System und Software

Für die Implementierung (siehe Teil III) und Optimierung wurde das Programm *FICOIVE Xpress v. 7.5* verwendet, das auf einem *Windows-7*-Rechner (64Bit) mit 4 GB Arbeitsspeicher und *AMD-PhenomX6-2.8GHz*-Prozessor installiert war.

Teil II

Optimierung von Tastenbelegungen

3 Distanzminimierung für vertikale Anwendungen

Aufgrund der anfangs intuitiveren Handhabung von Smartphones in vertikaler Position ist es naheliegend, die Optimierung der Tastenbelegung hinsichtlich der zurückzulegenden Distanz mit einem Kapitel für diesen Fall zu beginnen. Das komplette dritte Kapitel widmet sich also der aus der vertikalen Geräteposition resultierenden Ein-Finger-Schreibweise auf virtuellen Tastaturen. Dabei wird zunächst ein Ansatz untersucht, der das Problem einer optimalen Tastenbelegung für vertikale Anwendungen grundsätzlich löst (siehe Abschnitt 3.1). Er basiert jedoch auf einer vereinfachenden Annahme, die einen Einsatz der resultierenden Tastaturen in der Realität nicht sinnvoll erscheinen lässt. Im zweiten Teil dieses Kapitels wird ein Ansatz vorgestellt, welcher die Realität besser abbildet (siehe Abschnitt 3.2). In diesem Zusammenhang stößt man auf Schwierigkeiten, die aus der Modellierung und der Art des Problems resultieren. Mithilfe verschiedener Ideen und Vereinfachungen lassen sich diese in den Griff bekommen und man gelangt zu zufriedenstellenden Ergebnissen. Den dritten Teil bildet die Verallgemeinerung der in Abschnitt 3.2 erarbeiteten Theorie, bevor in 3.4 die Ergebnisse der vertikalen Distanzminimierung zusammengefasst werden.

3.1 Modellierung als lineares Zuordnungsproblem

3.1.1 Beschreibung und mathematische Formulierung

Um eine Verteilung von n Buchstaben auf m Tasten berechnen zu können, die unseren Wunsch nach einer möglichst kurzen zurückzulegenden Distanz erfüllt, ist es zunächst nötig, die verwendete Schreibweise zu modellieren. Zu diesem Zweck nimmt man an, dass die Texteingabe bei dem Punkt beginnt und endet, auf dem der Finger des Benutzers vor Beginn der Texteingabe positioniert ist. Desweiteren nimmt man an, dass der Finger zwischen jeder Eingabe zweier Zeichen zu diesem Startpunkt SP zurückkehrt und sich dann erneut auf den Weg macht. Dies liefert die sogenannte vereinfachte vertikale Schreibweise, welche in Abbildung 7 schematisch dargestellt ist.

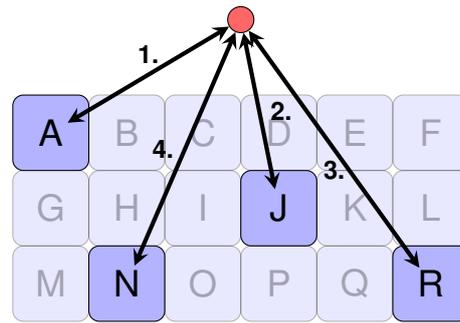


Abbildung 7: Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der vereinfachten Schreibweise in vertikaler Geräteposition.

Um eine Aussage über die erwartete zurückzulegende Gesamtdistanz treffen zu können, benötigt man die Wahrscheinlichkeit p_i mit $0 \leq p_i \leq 1$, dass ein Buchstabe $i \in [n]$ in einem Text vorkommt (siehe Abschnitt 2.4.1). Außerdem ist man auf die Distanz $d_j \in \mathbb{R}_{\geq 0}$ zwischen dem Startpunkt und jeder Taste $j \in [m]$ angewiesen (siehe (2.2) in Abschnitt 2.4.4). Um eine eindeutige Zuordnung von Zeichen auf Tasten zu erhalten, wird für jede Buchstabe-Taste-Kombination $(i, j) \in [n] \times [m]$ eine binäre Optimierungsvariable y_{ij} eingeführt, die angibt, ob sich der Buchstabe i auf Taste j befindet. Dies ist genau dann der Fall, wenn $y_{ij} = 1$. Für $y_{ij} = 0$ liegt der Buchstabe i nicht auf Taste j . Aus diesen Komponenten lässt sich die lineare Funktion formulieren, welche die erwartete zurückzulegende Distanz pro Buchstabe beschreibt und wie folgt lautet:

$$\sum_{i=1}^n \sum_{j=1}^m 2 \cdot d_j \cdot p_i \cdot y_{ij} \quad (3.1)$$

Multipliziert mit der durchschnittlichen Buchstabenanzahl B eines Textes (siehe Abschnitt 2.4.2), erhält man die zurückzulegende Gesamtstrecke.

Eine Bedingung an die Tastenbelegungen ist, dass jeder Buchstabe genau einmal vorkommt (siehe Abschnitt 2.3.1). Mathematisch ausgedrückt lautet diese Forderung:

$$\sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \quad (3.2)$$

Außerdem haben die gewählten Layouts die Eigenschaft, dass jede Taste Platz für höchstens einen Buchstaben bietet (siehe Abschnitt 2.3.1). Diese Eigenschaft lässt sich als lineare Nebenbedingung folgendermaßen formulieren:

$$\sum_{i=1}^n y_{ij} \leq 1 \quad \forall j \in [m] \quad (3.3)$$

Fasst man die obigen Komponenten zusammen, erhält man ein binäres, lineares Optimierungs-

problem, welches die Suche nach einer distanz-minimalen Tastenbelegung für die vereinfachte vertikale Bedienung beschreibt:

$$\begin{aligned}
 \min \quad & B \cdot \sum_{i=1}^n \sum_{j=1}^m 2 \cdot d_j \cdot p_i \cdot y_{ij} \\
 \text{u.d.N.} \quad & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \\
 & \sum_{i=1}^n y_{ij} \leq 1 \quad \forall j \in [m] \\
 & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m]
 \end{aligned} \tag{P1}$$

Da dieses ganzzahlige, lineare Optimierungsproblem (ILP) aus der Klasse der linearen Zuordnungsprobleme mit $m \cdot n$ Variablen und $m+n$ Nebenbedingungen (ohne die Binärbedingungen) weder zu den großen Problemen der kombinatorischen Optimierung zählt, noch die Lösung eine besondere Herausforderung darstellt, kann es in polynomieller Zeit gelöst werden.

Folgende Lemmata widmen sich den Eigenschaften von (P1), wobei standardmäßig $i \in [n]$ und $j \in [m]$ gilt, jedoch an manchen Stellen aus Gründen der Übersichtlichkeit nicht explizit erwähnt wird. Das erste Lemma befasst sich mit der Existenz von zulässigen Punkten.

Lemma 3.1. *Es gilt: $\mathcal{F}_{P1} \neq \emptyset$.*

Beweis.

Zu zeigen: $\exists \tilde{y} \in \mathbb{R}^{mn}$ für das gilt $\tilde{y} \in \mathcal{F}_{P1}$.

Wähle $\tilde{y}_{ij} = 1 \forall i = j$ und $\tilde{y}_{ij} = 0 \forall i \neq j$, dann gilt $\sum_{i=1}^n \tilde{y}_{ij} \leq 1 \forall j$ und $\sum_{j=1}^m \tilde{y}_{ij} = 1 \forall i$

und $\tilde{y}_{ij} \in \{0, 1\} \forall i, j$. Damit erfüllt \tilde{y} alle Nebenbedingungen aus (P1) und es folgt $\tilde{y} \in \mathcal{F}_{P1}$. □

Mehr als an der Existenz von zulässigen Punkten für (P1) ist man an der Existenz einer Lösung interessiert. Aus diesem Grund untersucht folgendes Lemma die Beschränktheit von (P1).

Lemma 3.2. *Sei $C := B \cdot \sum_{i=1}^n \sum_{j=1}^m 2 \cdot d_j \cdot p_i$. Es gilt: $0 \leq f_{P1}(y) \leq C < \infty \forall y \in \mathcal{F}_{P1}$.*

Beweis.

Sei $C := B \cdot \sum_{i=1}^n \sum_{j=1}^m 2 \cdot d_j \cdot p_i$. Die Behauptung $C < \infty$ folgt aufgrund der Tatsache, dass

$B, m, n \in \mathbb{N}$ und $0 \leq p_i \leq 1 \forall i$ und $d_j \in \mathbb{R}_{\geq 0} \forall j$. Außerdem gilt dank $y_{ij} \in \{0, 1\}$ und $2 \cdot d_j \cdot p_i \geq 0 \forall i, j$ und $B \in \mathbb{N}$, dass $f_{P1}(y) \geq 0$. Bleibt zu zeigen, dass gilt $f_{P1}(y) \leq C \forall y \in \mathcal{F}_{P1}$.

Angenommen es existiert ein $\tilde{y} \in \mathcal{F}_{P1}$ für das gilt: $f_{P1}(\tilde{y}) > C$. Dann existieren $\tilde{i} \in [n]$ und $\tilde{j} \in [m]$ mit $\tilde{y}_{\tilde{i}\tilde{j}} > 1$ und damit $\tilde{y}_{\tilde{i}\tilde{j}} \notin \{0, 1\}$. Letzteres ist ein Widerspruch zu $\tilde{y} \in \mathcal{F}_{P1}$. □

Lemma 3.3. Es gilt: $\mathcal{L}_{P_1} \neq \emptyset$.

Beweis.

Angenommen $\mathcal{L}_{P_1} = \emptyset$, dann ist (P1) in Zielfunktionsrichtung unbeschränkt oder $\mathcal{F}_{P_1} = \emptyset$. Dies widerspricht jedoch den Aussagen aus Lemma 3.1 oder Lemma 3.2. □

Dank Lemma 3.3 weiß man, dass das Problem (P1) mindestens eine Lösung besitzt, weshalb es sich für die Suche nach einer optimalen Tastenbelegung eignet. Diese kann im Anschluss an die Optimierung aus den y -Variablen ausgelesen werden.

3.1.2 Ergebnisse der Optimierung

Als Inputdaten dienen die Werte der Buchstabenwahrscheinlichkeiten p_i aus Tabelle 24, sowie die Distanzen d_j des jeweiligen Layouts (siehe Tabelle 27, 31, 35 und 40).

Für das *QWERTY-like*-Tastaturlayout liefert der Solver in 0,08 s eine optimale Lösung (siehe Abb. 8) mit Zielfunktionswert 372,414 LE. Der Zielfunktionswert gibt dabei die Distanz an, die der Finger für einen Text der Länge B zurücklegen müsste, falls die Buchstaben gemäß den verwendeten Wahrscheinlichkeiten vorkommen.

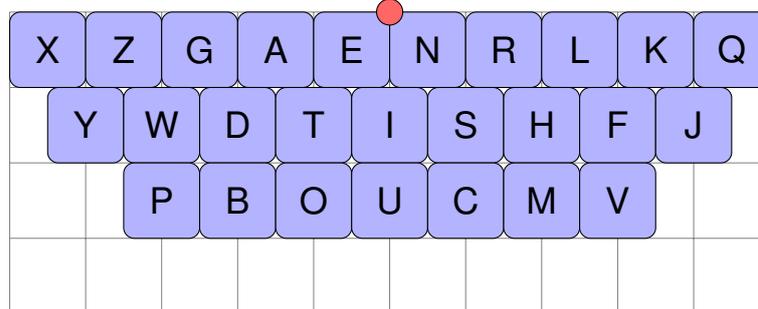


Abbildung 8: Distanz-optimale Tastenbelegung für das *QWERTY-like*-Layout bei vereinfachter Schreibweise im vertikalen Fall.

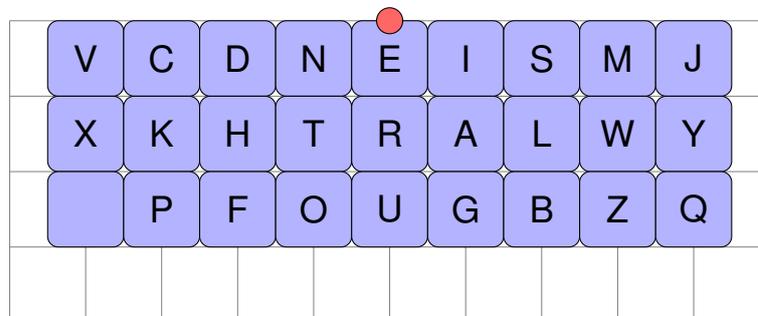


Abbildung 9: Distanz-optimale Tastenbelegung für das 3x9-Layout bei vereinfachter Schreibweise im vertikalen Fall.

Die Lösung des Optimierungsproblems für das 3x9-Layout (siehe Abb. 9) erhält man in 0,08 s, wobei das Minimum der Zielfunktion gleich 368,661 LE ist.

Die Bestimmung der Lösung für das 3x10-Layout (siehe Abb. 10) dauert 0,09 s, der Zielfunktionswert beträgt 372,685 LE.

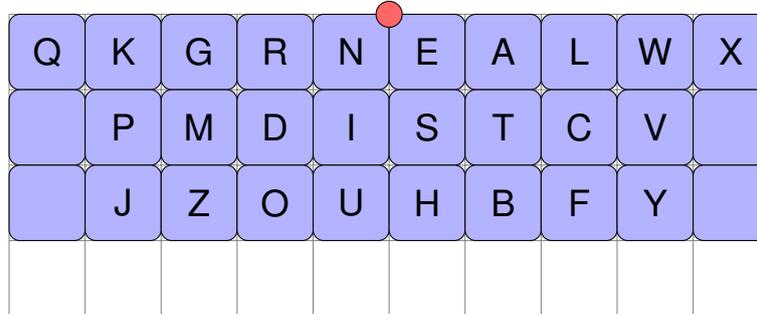


Abbildung 10: Distanz-optimale Tastenbelegung für das 3x10-Layout bei vereinfachter Schreibweise im vertikalen Fall.

Für das 4x8-Layout erhält man in 0,1 s die Tastenbelegung aus Abbildung 11, die einen Zielfunktionswert von 371,037 LE aufweist.

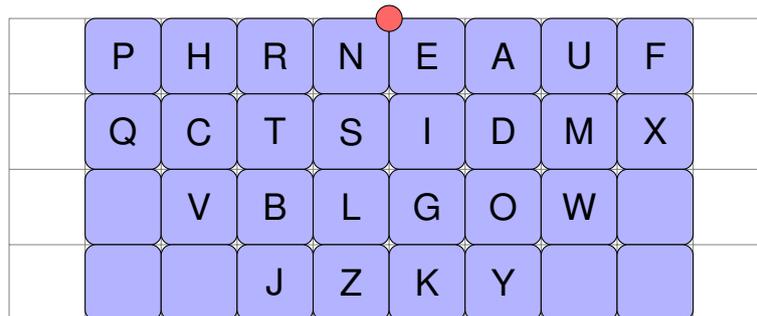


Abbildung 11: Distanz-optimale Tastenbelegung für das 4x8-Layout bei vereinfachter Schreibweise im vertikalen Fall.

Die Ergebnisse der Optimierung stimmen mit den Erwartungen überein. So sind Buchstaben, die häufig in einem Text auftreten, also eine hohe Wahrscheinlichkeit besitzen (z.B. "e", $p(e) \approx 17,5\%$, siehe Tabelle 24) nahe am Startpunkt zu finden. Diese geringe Entfernung erscheint sinnvoll, da die dazugehörige Strecke zum Startpunkt oftmals zurückgelegt werden muss. Dagegen sind Tasten ohne Belegung oder mit seltenen Buchstaben weiter entfernt vom Startpunkt, da sie seltener angesteuert werden.

Außerdem kann man feststellen, dass die Zielfunktionswerte und damit die zurückzulegenden Distanzen für die Tastenbelegungen der vier Layouts sehr ähnlich sind (maximale Differenz ca. 1%). Der Grund dafür ist, dass sich die Layouts aus Sicht des Startpunktes kaum unterscheiden. Alle verfügbaren Tasten liegen lückenlos in einem 180°-Kreissektor mit dem Startpunkt als Kreismittelpunkt. Darüberhinaus verfügen alle Layouts über eine ähnliche Anzahl an Tasten in bestimmter Entfernung zum Startpunkt.

Löst man sich von den gegebenen Tastaturlayouts, so würde eine kompakte, kreisförmige Tastatur mit Startpunkt im Kreismittelpunkt den besten Zielfunktionswert liefern, da sie die Anzahl der Tasten in geringer Entfernung maximiert. Die Buchstaben wären dabei mit von innen nach außen absteigender Wahrscheinlichkeit verteilt.

3.2 Modellierung als quadratisches Zuordnungsproblem

Wie bereits zu Beginn des Kapitels erwähnt, ist die Annahme, dass der Benutzer zwischen je zwei Buchstaben zum Startpunkt zurückkehrt nicht besonders realitätsnah, weshalb die in Abschnitt 3.1.2 präsentierten Tastenbelegungen für ein normales Schreibverhalten nicht optimal wären und deren Einsatz deshalb nicht zu empfehlen ist. Aus diesem Grund widmet sich der folgende Abschnitt der Erzeugung von Tastenbelegungen, welche besser für den Einsatz in der Praxis geeignet sind.

3.2.1 Beschreibung und mathematische Formulierung

Analog zum vorherigen Abschnitt zielen wir auf die Minimierung der erwarteten zurückzulegenden Distanz ab. Wir wollen uns zu Beginn auf Tastaturlayouts beschränken, deren Anzahl an Tasten gleich der Anzahl an zu verteilenden Buchstaben ist (also $m = n$). Diese Forderung ist jedoch nur vorübergehend und wird in Abschnitt 3.3 aufgegeben. Um die Erkenntnisse später leichter übertragen zu können, werden $[m]$ und $[n]$ als Indexmengen weiterverwendet. Im gesamten Abschnitt gilt standardmäßig, dass $i, k \in [n]$ und $j, l \in [m]$.

Im Gegensatz zur vereinfachten vertikalen Schreibweise aus Abschnitt 3.1 soll nun zwischen dem Tippen zweier Tasten keine Rückkehr zum Startpunkt erfolgen. Das heißt, der Benutzer kann sich nach dem Antippen einer Taste direkt und ohne Umwege zur nächsten Taste begeben. Abbildung 12 veranschaulicht diese Schreibweise, welche im weiteren Verlauf als realistische vertikale Schreibweise bezeichnet wird.

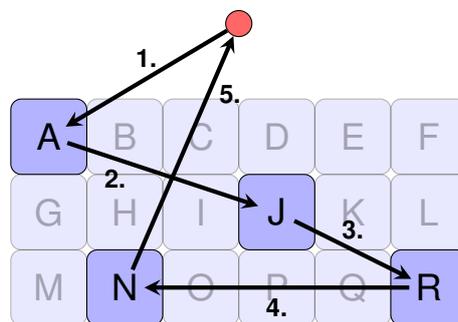


Abbildung 12: Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der realistischen Schreibweise in vertikaler Geräteposition.

In der Optimierung führt das dazu, dass die Buchstaben nicht mehr unabhängig voneinander und nur gemäß ihrer Wahrscheinlichkeiten verteilt werden, sondern dass zusätzlich die Häufigkeit des Auftretens jeder Kombination zweier Buchstaben (Bigramm) in die Lösung einfließt. Somit benötigt man die Wahrscheinlichkeiten $0 \leq p_{ik} \leq 1$ für alle Bigramme $(i, k) \in [n] \times [n]$. Desweiteren benötigt man die Distanzen $d_{jl} \in \mathbb{R}_{\geq 0}$ zwischen allen Tastenpaaren $(j, l) \in [m] \times [m]$ (siehe Abschnitt 2.4.4).

Modelliert man zunächst die durchschnittliche Strecke, die zwischen einem Buchstabenpaar zurückgelegt werden muss, so gelangt man mit Hilfe der binären Optimierungsvariable y_{ij} (siehe Abschnitt 3.1.1) zu folgender nichtlinearer Funktion:

$$\sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot y_{ij} \cdot y_{kl} \quad (3.4)$$

Eine Umrechnung auf die zurückzulegende Gesamtstrecke zwischen allen Buchstabenpaaren gelingt durch die Multiplikation von Funktion (3.4) mit der Anzahl der Bigramme $B - 1$.

Die Forderung, dass jeder Buchstabe genau einmal auf der Tastatur vorkommt (siehe Gleichung (3.2)), in Verbindung mit $m = n$ und der Tatsache, dass jede Taste Platz für höchstens einen Buchstaben bietet (siehe Ungleichung (3.3)), führt dazu, dass sich auf jeder Taste genau ein Buchstaben befindet. Dies liefert die Nebenbedingung:

$$\sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \quad (3.5)$$

Damit ergibt sich folgendes nichtlineares Optimierungsproblem:

$$\begin{aligned} & \min (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot y_{ij} \cdot y_{kl} \\ & \text{u.d.N.} \quad \sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \\ & \quad \quad \quad \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \\ & \quad \quad \quad y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \end{aligned} \quad (P2)$$

Aus den ganzzahligen Lösungswerten der y -Variablen des Problems (P2) lässt sich analog zu (P1) die optimale Tastenbelegung rekonstruieren. Zu den Eigenschaften von (P2) kann man folgende Aussagen machen.

Korollar 3.4. *Es gilt:*

1. $\mathcal{F}_{P2} \neq \emptyset$.
2. $\mathcal{F}_{P2} \subseteq \mathcal{F}_{P1}$ und $\mathcal{P}_{P2} \subseteq \mathcal{P}_{P1}$.

3 Distanzminimierung für vertikale Anwendungen

3. $\exists C \in \mathbb{R} \forall y \in \mathcal{F}_{P_2} : 0 \leq f_{P_2}(y) \leq C < \infty$.

4. $\mathcal{L}_{P_2} \neq \emptyset$.

Beweis.

Zu 1.: Wähle \tilde{y} (analog zum Beweis von Lemma 3.1) so, dass $\tilde{y}_{ij} = 1 \forall i = j$ und $\tilde{y}_{ij} = 0 \forall i \neq j$. Dann gilt $\sum_{i=1}^n \tilde{y}_{ij} = 1 \forall j$ und $\sum_{j=1}^m \tilde{y}_{ij} = 1 \forall i$ und $\tilde{y}_{ij} \in \{0, 1\} \forall i, j$. Damit erfüllt \tilde{y} die Nebenbedingungen aus (P2) und es gilt $\tilde{y} \in \mathcal{F}_{P_2}$.

Zu 2.: Die Behauptungen folgen direkt aus der Tatsache, dass die Nebenbedingungen von (P2) eine Teilmenge der Nebenbedingungen aus (P1) sind. Denn angenommen es existiert ein $\tilde{y} \in \mathcal{F}_{P_2}$ mit $\tilde{y} \notin \mathcal{F}_{P_1}$, dann gibt es mindestens eine Nebenbedingung in (P1), die von \tilde{y} verletzt wird. Diese Nebenbedingung ist auch in (P2) verletzt, weshalb gilt: $\tilde{y} \notin \mathcal{F}_{P_2} \Rightarrow$ Widerspruch.

Zu 3.: Sei $C := (B - 1) \cdot \sum_{i,k=1}^n \sum_{j,l=1}^m p_{ik} \cdot d_{jl}$, dann gilt analog zum Beweis von Lemma 3.2, dass $C < \infty$, da $B, m, n \in \mathbb{N}$ und $0 \leq p_{ik} \leq 1 \forall i, k$ und $d_{jl} \in \mathbb{R}_{\geq 0} \forall j, l$. Außerdem gilt mit $y_{ij} \in \{0, 1\}$ und $2 \cdot d_{jl} \cdot p_{ik} \geq 0 \forall i, j, k, l$ und $B \in \mathbb{N}$, dass $f_{P_2}(y) \geq 0$. Bleibt zu zeigen, dass gilt $f_{P_2}(y) \leq C \forall y \in \mathcal{F}_{P_2}$. Angenommen es existiert ein $\tilde{y} \in \mathcal{F}_{P_2}$ für das gilt: $f_{P_2}(\tilde{y}) > C$. Dann existieren $\tilde{i}, \tilde{k} \in [n]$ und $\tilde{j}, \tilde{l} \in [m]$ mit $\tilde{y}_{\tilde{i}\tilde{j}} \cdot \tilde{y}_{\tilde{k}\tilde{l}} > 1$ und damit $\tilde{y}_{\tilde{i}\tilde{j}} \notin \{0, 1\} \vee \tilde{y}_{\tilde{k}\tilde{l}} \notin \{0, 1\}$. Somit verletzt \tilde{y} mindestens eine der Binärbedingungen aus (P2) und es gilt: $\tilde{y} \notin \mathcal{F}_{P_2} \Rightarrow$ Widerspruch.

Zu 4.: Die Behauptung folgt analog zum Beweis von Lemma 3.3, denn angenommen $\mathcal{L}_{P_2} = \emptyset$ dann wäre (P2) in Zielfunktionsrichtung unbeschränkt oder $\mathcal{F}_{P_2} = \emptyset$. Dies widerspricht jedoch den Aussagen 1 und 3 dieses Korollars. □

Lemma 3.5. Sei $m = n$, dann gilt: $\dim(\mathcal{P}_{P_2}) = n^2 - 2n + 1$.

Beweis.

Sei $m = n$. Definiere $\mathcal{A} := \{y \in \mathbb{R}^{n^2} \mid 0 \leq y_{ij} \forall i, j \text{ und } \sum_{i=1}^n y_{ij} = 1 \forall j \text{ und } \sum_{j=1}^n y_{ij} = 1 \forall i\}$.

Dann gilt nach Billera u. Sarangarajan [1996], dass $\dim(\mathcal{A}) = (n - 1)^2 = n^2 - 2n + 1$.

Außerdem gilt $\mathcal{P}_{P_2} := \{y \in \mathbb{R}^{mn} \mid \sum_{i=1}^n y_{ij} = 1 \forall j \text{ und } \sum_{j=1}^m y_{ij} = 1 \forall i \text{ und } 0 \leq y_{ij} \leq 1 \forall i, j\}$.

Die Bedingungen $\sum_{i=1}^n y_{ij} = 1 \forall j$ und $0 \leq y_{ij} \forall i, j$ bewirken die Redundanz der Bedingungen $y_{ij} \leq 1 \forall i, j$. Mit $m = n$ gilt deshalb $\mathcal{P}_{P_2} = \mathcal{A}$ und damit $\dim(\mathcal{P}_{P_2}) = \dim(\mathcal{A}) = n^2 - 2n + 1$. □

Obwohl sich die Größe des Problems (P2) hinsichtlich der Anzahl an Nebenbedingungen und Variablen im Vergleich zu (P1) nicht verändert hat, sind die Komplexitäten der beiden Probleme nicht vergleichbar. Während (P1) ein lineares Zuordnungsproblem ist, das in polynomieller Zeit gelöst werden kann, ist (P2) das klassische quadratische Zuordnungsproblem (QAP). Dieses wurde erstmals 1957 im Zusammenhang mit der optimalen Wahl von Fabrik-Standorten beschrieben (siehe [Koopmans u. Beckmann, 1957]). Eine frühe Beschreibung der Suche nach optimalen Tastenbelegungen für Schreibmaschinen mittels quadratischer Zuordnungsprobleme findet sich in [Burkard u. Offermann, 1977]. Das QAP, zählt zur Klasse der NP-harten Probleme (siehe [Sahni u. Gonzalez, 1976]), die nicht in polynomieller Zeit gelöst werden können (außer P=NP).

In diesem Fall ist man häufig an einer Linearisierung des Problems interessiert (siehe bspw. [Adams et al., 2004]), um die zahlreichen bekannten Techniken für eine Verbesserung der Lösungsdauer eines ganzzahligen, linearen Problems anwenden und kombinieren zu können.

Ein erster Schritt auf dem Weg zur Linearisierung von (P2) ist die Einführung zusätzlicher Variablen $x_{ijkl} \in \{0, 1\}$. Sie ermöglichen zunächst die Formulierung eines zu (P2) äquivalenten Problems mit linearer Zielfunktion und nichtlinearen Nebenbedingungen (siehe Problem (P3)), welche im Anschluss in Abschnitt 3.2.2 linearisiert werden und so auf das ganzzahlige, lineare Optimierungsproblem (P4) führen.

Die neuen Variablen x_{ijkl} geben an, ob der Buchstabe $i \in [n]$ auf der Taste $j \in [m]$ und der Buchstabe $k \in [n]$ auf der Taste $l \in [m]$ liegt ($x_{ijkl} = 1$) oder nicht ($x_{ijkl} = 0$). Dabei muss ihre Binarität nicht explizit gefordert werden, sie folgt implizit aus der Beziehung, die zwischen den x - und y -Variablen gilt und als neue Nebenbedingung in das Problem eingeht. Sie lautet:

$$x_{ijkl} = y_{ij} \cdot y_{kl} \quad \forall i, k \in [n], j, l \in [m] \quad (3.6)$$

Damit ergibt sich folgendes Problem:

$$\begin{aligned} \min & (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} \\ \text{u.d.N.} & \sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \\ & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \\ & x_{ijkl} = y_{ij} \cdot y_{kl} \quad \forall i, k \in [n], j, l \in [m] \\ & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \end{aligned} \quad (P3)$$

Das Problem zur Erzeugung optimaler Tastenbelegungen ist im Vergleich zu (P2) durch die Einführung der Variablen x_{ijkl} und der zusätzlichen Nebenbedingungen deutlich größer gewor-

3 Distanzminimierung für vertikale Anwendungen

den. (P3) besitzt $m^2 \cdot n^2 + m \cdot n$ Variablen und $m^2 \cdot n^2 + m + n$ Nebenbedingungen (ohne die Binärbedingungen). Für das Optimierungsproblem (P3) gelten folgende Aussagen.

Lemma 3.6. *Es gilt: $(y, x) \in \mathcal{F}_{P3} \Rightarrow y \in \mathcal{F}_{P2}$.*

Beweis.

Sei $(y, x) \in \mathcal{F}_{P3}$. Angenommen es gilt $y \notin \mathcal{F}_{P2}$, dann existiert mindestens eine Nebenbedingung in (P2), die von y verletzt wird. Da die Nebenbedingungen aus (P2) eine Teilmenge der Nebenbedingungen aus (P3) sind, verletzt (y, x) diese Nebenbedingung auch in (P3). Damit ist $(y, x) \notin \mathcal{F}_{P3}$, also kein zulässiger Punkt für (P3), was ein Widerspruch zur Voraussetzung ist. \square

Lemma 3.7. *Sei $y \in \mathbb{R}^{mn}$, $x \in \mathbb{R}^{m^2n^2}$, dann gilt:*

$$x_{ijkl} = y_{ij} \cdot y_{kl} \quad \forall i, k \in [n], j, l \in [m] \Leftrightarrow [y \in \mathcal{F}_{P2} \Rightarrow (y, x) \in \mathcal{F}_{P3}].$$

Beweis. Sei $y \in \mathbb{R}^{mn}$, $x \in \mathbb{R}^{m^2n^2}$.

" \Rightarrow " Sei $x_{ijkl} = y_{ij} \cdot y_{kl} \quad \forall i, j, k, l$ und $y \in \mathcal{F}_{P2}$. Dann erfüllt (y, x) alle Nebenbedingungen aus (P3) und es folgt $(y, x) \in \mathcal{F}_{P3}$.

" \Leftarrow " Sei $y \in \mathcal{F}_{P2}$ und es gelte $[y \in \mathcal{F}_{P2} \Rightarrow (y, x) \in \mathcal{F}_{P3}]$. Dann bleibt zu zeigen, dass (y, x) den Zusammenhang $[x_{ijkl} = y_{ij} \cdot y_{kl} \quad \forall i, j, k, l]$ erfüllt. Angenommen $[x_{ijkl} = y_{ij} \cdot y_{kl} \quad \forall i, j, k, l]$ gilt nicht, dann existieren $\tilde{i}, \tilde{k} \in [n]$ und $\tilde{j}, \tilde{l} \in [m]$ mit $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \neq y_{\tilde{i}\tilde{j}} \cdot y_{\tilde{k}\tilde{l}}$ und (y, x) erfüllt nicht alle Nebenbedingungen von (P3). Damit gilt $(y, x) \notin \mathcal{F}_{P3}$, was ein Widerspruch zu den Voraussetzungen ist. \square

Korollar 3.8. *Es gilt: $\mathcal{F}_{P3} \neq \emptyset$.*

Beweis.

Sei $\tilde{y} \in \mathcal{F}_{P2}$ (Existenz wg. Korollar 3.4, Aussage 1). Wähle $\tilde{x}_{ijkl} = \tilde{y}_{ij} \cdot \tilde{y}_{kl} \quad \forall i, k \in [n], j, l \in [m]$, dann folgt mit Lemma 3.7, dass $(\tilde{y}, \tilde{x}) \in \mathcal{F}_{P3}$ und damit $\mathcal{F}_{P3} \neq \emptyset$. \square

Die Existenz zulässiger Punkte beantwortet allerdings nicht die Frage ob das Problem (P3) tatsächlich optimale Tastenbelegungen liefert. Dieser Frage widmen sich die folgenden Lemmata.

Lemma 3.9. *Es gilt: $(y^*, x^*) \in \mathcal{L}_{P3} \Rightarrow y^* \in \mathcal{L}_{P2}$.*

Beweis.

$$\text{Sei } (y^*, x^*) \in \mathcal{L}_{P3} \Rightarrow \begin{cases} (i) & (y^*, x^*) \in \mathcal{F}_{P3} \\ (ii) & f_{P3}(y^*, x^*) \leq f_{P3}(y, x) \quad \forall (y, x) \in \mathcal{F}_{P3} \end{cases}$$

Aus (i) und Lemma 3.6 folgt, dass $y^* \in \mathcal{F}_{P2}$.

Angenommen $y^* \notin \mathcal{L}_{P2}$, dann existiert ein $\hat{y} \in \mathcal{F}_{P2}$ mit $f_{P2}(\hat{y}) < f_{P2}(y^*)$.

Wähle \hat{x} so, dass $\hat{x}_{ijkl} = \hat{y}_{ij} \cdot \hat{y}_{kl} \forall i, j, k, l$, dann folgt mit Lemma 3.7, dass $(\hat{y}, \hat{x}) \in \mathcal{F}_{P3}$.
 Außerdem gilt: $f_{P3}(\hat{y}, \hat{x}) = f_{P2}(\hat{y}) < f_{P2}(y^*) = f_{P3}(y^*, x^*)$, was ein Widerspruch zu (ii) ist. \square

Lemma 3.10. Sei $y \in \mathbb{R}^{mn}$, $x \in \mathbb{R}^{m^2n^2}$, dann gilt:

$$x_{ijkl}^* = y_{ij}^* \cdot y_{kl}^* \forall i, k \in [n], j, l \in [m] \Leftrightarrow [y^* \in \mathcal{L}_{P2} \Rightarrow (y^*, x^*) \in \mathcal{L}_{P3}].$$

Beweis.

" \Rightarrow " Sei $x_{ijkl}^* = y_{ij}^* \cdot y_{kl}^* \forall i, j, k, l$ und $y^* \in \mathcal{L}_{P2}$. Zu zeigen: $(y^*, x^*) \in \mathcal{L}_{P3}$.

$$\text{Aus } y^* \in \mathcal{L}_{P2} \text{ folgt, dass } \begin{cases} (i) & y^* \in \mathcal{F}_{P2} \\ (ii) & f_{P2}(y^*) \leq f_{P2}(y) \forall y \in \mathcal{F}_{P2} \end{cases}$$

Aus (i) und Lemma 3.7 folgt, dass $\Rightarrow (y^*, x^*) \in \mathcal{F}_{P3}$.

Außerdem gilt mit (ii): $f_{P3}(y^*, x^*) = f_{P2}(y^*) \leq f_{P2}(y) = f_{P3}(y, x) \forall (y, x) \in \mathcal{F}_{P3}$,
 weshalb (y^*, x^*) Lösung von (P3) ist. Also gilt: $(y^*, x^*) \in \mathcal{L}_{P3}$.

" \Leftarrow " Es gelte $[y^* \in \mathcal{L}_{P2} \Rightarrow (y^*, x^*) \in \mathcal{L}_{P3}]$ (*).

$$\text{Sei außerdem } y^* \in \mathcal{L}_{P2}, \text{ dann folgt: } \begin{cases} (i) & y^* \in \mathcal{F}_{P2} \\ (ii) & f_{P2}(y^*) \leq f_{P2}(y) \forall y \in \mathcal{F}_{P2} \end{cases}$$

und zusammen mit (*), dass $(y^*, x^*) \in \mathcal{L}_{P3}$.

Angenommen $[x_{ijkl}^* = y_{ij}^* \cdot y_{kl}^* \forall i, j, k, l]$ gilt nicht, dann existieren $\tilde{i}, \tilde{k} \in [n]$

und $\tilde{j}, \tilde{l} \in [m]$ mit $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* \neq y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^*$. Also verletzt (y^*, x^*) eine Nebenbedingung
 aus (P3) und es folgt: $(y^*, x^*) \notin \mathcal{F}_{P3}$. Damit ist (y^*, x^*) keine Lösung von (P3),
 was einen Widerspruch zu $(y^*, x^*) \in \mathcal{L}_{P3}$ darstellt. \square

Lemma 3.11. Es gilt: $\mathcal{L}_{P3} \neq \emptyset$.

Der Beweis könnte analog zum Beweis von Lemma 3.3 über die Existenz zulässiger Punkte und die Beschränktheit von (P3) geführt werden. Eine alternative Möglichkeit ist folgender Beweis.

Beweis.

Sei $y^* \in \mathcal{L}_{P2}$ (Existenz dank Aussage 4 aus Lemma 3.4) und wähle $x^* \in \mathbb{R}^{m^2n^2}$ so, dass
 $x_{ijkl}^* = y_{ij}^* \cdot y_{kl}^* \forall i, k \in [n], j, l \in [m]$, dann gilt nach Lemma 3.10, dass $(y^*, x^*) \in \mathcal{L}_{P3}$. \square

3.2.2 Linearisierung des Problems

Mathematische Formulierung

Um ausgehend von (P3) ein lineares, ganzzahliges Optimierungsproblem zu erhalten, müssen die nichtlineare Nebenbedingungen (3.6), welche die x -Variablen an die y -Variablen koppeln,

linearisiert werden. Dies gelingt mittels drei neuen Gruppen von Nebenbedingungen. Die Erste dient als untere Schranke an x_{ijkl} und lautet:

$$y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall i, k \in [n], j, l \in [m] \quad (3.7)$$

Die zweite Gruppe beschränkt x_{ijkl} von oben und lautet:

$$2 \cdot x_{ijkl} \leq y_{ij} + y_{kl} \quad \forall i, k \in [n], j, l \in [m] \quad (3.8)$$

Die dritte Gruppe fordert die Ganzzahligkeit der x_{ijkl} :

$$x_{ijkl} \in \{0, 1\} \quad \forall i, k \in [n], j, l \in [m] \quad (3.9)$$

Zusammengefasst ergibt sich folgendes lineares, ganzzahliges Optimierungsproblem:

$$\begin{aligned} & \min (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} \\ \text{u.d.N.} & \sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \\ & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \\ & y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall i, k \in [n], j, l \in [m] \\ & 2 \cdot x_{ijkl} \leq y_{ij} + y_{kl} \quad \forall i, k \in [n], j, l \in [m] \\ & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \\ & x_{ijkl} \in \{0, 1\} \quad \forall i, k \in [n], j, l \in [m] \end{aligned} \quad (P4)$$

Das Problem (P4) liefert ganzzahlige Lösungen und besteht aus $m^2n^2 + mn$ Variablen und $2m^2n^2 + m + n$ Nebenbedingungen. Darüberhinaus besitzt es folgende Eigenschaften.

Korollar 3.12. *Es gilt:*

1. $\mathcal{F}_{P4} \neq \emptyset$.
2. Für $m = n = 1$ gilt: $\mathcal{F}_{P4} = \mathcal{P}_{P4} = \{(1, 1)\}$.
3. Für $m = n \geq 2$ gilt: $\mathcal{F}_{P4} \subset \mathcal{P}_{P4}$.

Beweis.

Zu 1.: Betrachte (\tilde{y}, \tilde{x}) mit $\tilde{x}_{ijkl} = \tilde{y}_{ij} + \tilde{y}_{kl} - 1 \quad \forall i, j, k, l$ und $\tilde{y}_{ij} = 1 \quad \forall i = j$ und $\tilde{y}_{ij} = 0 \quad \forall i \neq j$.
Dann gilt $2 \cdot \tilde{x}_{ijkl} = \tilde{y}_{ij} + \tilde{y}_{kl} + \underbrace{\tilde{y}_{ij} + \tilde{y}_{kl} - 2}_{\leq 0} \leq \tilde{y}_{ij} + \tilde{y}_{kl}$ und $\sum_{i=1}^n \tilde{y}_{ij} = 1 \quad \forall j$ und $\sum_{j=1}^m \tilde{y}_{ij} = 1 \quad \forall i$.

Somit sind alle Nebenbedingungen von (P4) erfüllt und es gilt: $(\tilde{y}, \tilde{x}) \in \mathcal{F}_{P4}$.

Zu 2.: Sei $m = n = 1$, dann lauten die Bedingungen von (P4) (ohne Binärbedingungen): $y_{11} = 1$ und $y_{11} + y_{11} - 1 \leq x_{1111}$ und $2 \cdot x_{1111} \leq y_{11} + y_{11}$. Aus ihnen folgt direkt die Behauptung.

Zu 3.: Sei $m = n \geq 2$. Zu zeigen: $\exists(\hat{y}, \hat{x}) \in \mathcal{P}_{P4}$ mit $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P4}$. Betrachte (\hat{y}, \hat{x}) mit $\hat{y}_{ij} = \frac{1}{m} \forall i, j$ und $\hat{x}_{ijkl} = 0 \forall i, j, k, l$. Dann gilt $\sum_{i=1}^n \hat{y}_{ij} = 1 \forall j$ und $\sum_{j=1}^m \hat{y}_{ij} = 1 \forall i$, desweiteren gilt $\hat{y}_{ij} + \hat{y}_{kl} - 1 = \frac{2}{m} - 1 \leq 0 = \hat{x}_{ijkl}$ und $2 \cdot \hat{x}_{ijkl} = 0 \leq \frac{2}{m} = \hat{y}_{ij} + \hat{y}_{kl}$. Damit ist $(\hat{y}, \hat{x}) \in \mathcal{P}_{P4}$, aber $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P4}$, da $\hat{y}_{ij} \notin \{0, 1\}$. □

Lemma 3.13. Sei $(y, x) \in \mathbb{R}^{m^2n^2+mn}$ und erfülle (y, x) die Bedingungen (3.7),(3.8) und (3.9), dann gilt: $x_{ijkl} = y_{ij} \cdot y_{kl} \forall i, k \in [n], j, l \in [m]$.

Beweis. Sei $(y, x) \in \mathbb{R}^{m^2n^2+mn}$ und erfülle (y, x) die Bedingungen (3.7),(3.8) und (3.9). Angenommen $[x_{ijkl} = y_{ij} \cdot y_{kl} \forall i, j, k, l]$ gilt nicht, dann existieren $\tilde{i}, \tilde{k} \in [n]$ und $\tilde{j}, \tilde{l} \in [m]$ mit $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \neq y_{\tilde{i}\tilde{j}} \cdot y_{\tilde{k}\tilde{l}}$ und man unterscheidet folgende Fälle:

Fall 1: $y_{\tilde{i}\tilde{j}} = y_{\tilde{k}\tilde{l}} = 0$

Dann lauten die Nebenbedingungen $2 \cdot x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \leq 0$ und $-1 \leq x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}$ und $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \in \{0, 1\}$. Damit folgt $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0$, was ein Widerspruch zur Annahme ist.

Fall 2: $y_{\tilde{i}\tilde{j}} = 0$ und $y_{\tilde{k}\tilde{l}} = 1$ oder vice versa

Dann lauten die Nebenbedingungen $2 \cdot x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \leq 1$ und $0 \leq x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}$ und $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \in \{0, 1\}$. Damit folgt $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0$, was ein Widerspruch zur Annahme ist.

Fall 3: $y_{\tilde{i}\tilde{j}} = y_{\tilde{k}\tilde{l}} = 1$

Dann lauten die Nebenbedingungen $2 \cdot x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \leq 2$ und $1 \leq x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}$ und $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \in \{0, 1\}$. Es folgt, dass $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 1$, was ein Widerspruch zur Annahme ist. □

Lemma 3.14. Es gilt: $(y, x) \in \mathcal{F}_{P3} \Leftrightarrow (y, x) \in \mathcal{F}_{P4}$.

Beweis.

3 Distanzminimierung für vertikale Anwendungen

" \Rightarrow " Sei $(y, x) \in \mathcal{F}_{P3}$, dann gilt $x_{ijkl} = y_{ij} \cdot y_{kl} \forall i, j, k, l$ und mit Lemma 3.6, dass $y \in \mathcal{F}_{P2}$.

Es bleibt zu zeigen, dass $\forall i, j, k, l$ gilt, dass:
$$\begin{cases} (i) & y_{ij} + y_{kl} - 1 \leq x_{ijkl} \\ (ii) & 2 \cdot x_{ijkl} \leq y_{ij} + y_{kl} \\ (iii) & x_{ijkl} \in \{0, 1\} \end{cases}$$

Zu (i): Es gilt: $y_{ij} + y_{kl} - 1 \leq x_{ijkl} \Leftrightarrow y_{ij} + y_{kl} - 1 \leq y_{ij} \cdot y_{kl}$

$\Leftrightarrow 0 \leq y_{ij} \cdot y_{kl} - y_{ij} - y_{kl} + 1$. Dies ist erfüllt, da $y_{ij} \in \{0, 1\}$ (wg. $y \in \mathcal{F}_{P2}$).

Zu (ii): Es gilt: $2 \cdot x_{ijkl} \leq y_{ij} + y_{kl} \Leftrightarrow 0 \leq y_{ij} + y_{kl} - 2 \cdot x_{ijkl}$

$\Leftrightarrow 0 \leq y_{ij} + y_{kl} - 2 \cdot y_{ij} \cdot y_{kl} \Leftrightarrow 0 \leq \underbrace{y_{ij}}_{\geq 0} \cdot \underbrace{(1 - y_{kl})}_{\geq 0} + \underbrace{y_{kl}}_{\geq 0} \cdot \underbrace{(1 - y_{ij})}_{\geq 0}$

Zu (iii): Aus $y_{ij} \in \{0, 1\}$ und $x_{ijkl} = y_{ij} \cdot y_{kl} \forall i, j, k, l$ folgt offensichtlich $x_{ijkl} \in \{0, 1\}$.

" \Leftarrow " Sei $(y, x) \in \mathcal{F}_{P4}$, dann erfüllt (y, x) die Bedingungen (3.7),(3.8),(3.9) und es gilt $y \in \mathcal{F}_{P2}$ (da die Nebenbedingungen von (P2) eine Teilmenge der Bedingungen aus (P4) sind). Mit Lemma 3.13 folgt, dass $x_{ijkl} = y_{ij} \cdot y_{kl} \forall i, j, k, l$ weshalb gilt, dass $(y, x) \in \mathcal{F}_{P3}$. □

Lemma 3.15. Es gilt: $(y^*, x^*) \in \mathcal{L}_{P3} \Leftrightarrow (y^*, x^*) \in \mathcal{L}_{P4}$.

Beweis.

" \Rightarrow " Sei $(y^*, x^*) \in \mathcal{L}_{P3}$, dann gilt:
$$\begin{cases} (i) & (y^*, x^*) \in \mathcal{F}_{P3} \\ (ii) & f_{P3}(y^*, x^*) \leq f_{P3}(y, x) \forall (y, x) \in \mathcal{F}_{P3} \end{cases}$$

Aus (i) und Lemma 3.14 folgt $(y^*, x^*) \in \mathcal{F}_{P4}$.

Angenommen $(y^*, x^*) \notin \mathcal{L}_{P4}$, dann existiert $(\hat{y}, \hat{x}) \in \mathcal{F}_{P4}$ mit $f_{P4}(\hat{y}, \hat{x}) < f_{P4}(y^*, x^*)$.

Mit Lemma 3.14 gilt $(\hat{y}, \hat{x}) \in \mathcal{F}_{P3}$ und aufgrund von $f_{P3} = f_{P4}$ gilt, dass

$f_{P3}(\hat{y}, \hat{x}) < f_{P3}(y^*, x^*)$. Dies ist ein Widerspruch zu (ii).

" \Leftarrow " Der Beweis funktioniert analog zur Hinrichtung. □

Korollar 3.16. Es gilt: $\mathcal{L}_{P4} \neq \emptyset$.

Beweis.

Die Behauptung folgt aus der Existenz einer Lösung von (P3) (Lemma 3.11) und Lemma 3.15. □

Das Optimierungsproblem (P3), welches eine nichtlineare Nebenbedingung enthält und das binäre, lineare Problem (P4) besitzen also den gleichen Zulässigkeitsbereich, sowie die gleiche Lösungsmenge. Dank Lemma 3.9 gilt somit, dass die y -Komponenten jeder Lösung von (P4) auch das quadratische Zuordnungsproblem (P2) lösen. Dies rechtfertigt die Verwendung von

(P4) zur Bestimmung optimaler Tastenbelegungen im weiteren Verlauf. Dabei können die resultierenden Tastenbelegungen (analog zu (P3)) sowohl aus den x - als auch aus den y -Variablen rekonstruiert werden.

Qualitative Ergebnisse der Optimierung

Übergibt man das Problem (P4) an einen Solver für lineare, ganzzahlige Optimierungsprobleme, stellt man fest, dass diese Formulierung nicht dafür geeignet ist eine optimale Tastatur für den Praxiseinsatz zu ermitteln. Da wir es mit einem linearisierten QAP zu tun haben, ist es nicht verwunderlich, dass es sich um ein schwieriges Problem handelt und ein dramatischer Anstieg der Laufzeit für eine wachsende Zahl an Buchstaben und Tasten mit einhergeht. Tastenbelegungen in der Größenordnung von $m = n \leq 10$ lassen sich in der Regel noch berechnen, darüber stößt man bereits an eine Grenze.

Im konkreten Fall wurde das Optimierungsproblem (P4) in *Xpress IVE* implementiert (siehe Abschnitt 2.4.6) und anhand von Test-Instanz 1 (siehe Anhang A.2.5) getestet. Um verlässliche Aussagen treffen zu können, wurden eingebaute Presolving- und Cuttingstrategien, sowie Heuristiken ausgeschaltet. Abbildung 13 zeigt den Verlauf der Optimierung. Man erkennt, dass der Solver nach etwa 30 Sekunden eine optimale Lösung errechnet hat. Davon benötigt die Suche nach einem ersten zulässigen Punkt und damit einer oberen Schranke an die Optimallösung etwa 5 Sekunden, die restlichen 25 Sekunden entfallen auf die Optimierung in Form des Abarbeitens des *Branch&Bound*-Baums.

Betrachtet man die untere Kurve, welche das Anwachsen der besten unteren Schranke im zeitlichen Verlauf visualisiert, fällt auf, dass diese nur sehr langsam ansteigt. Dies hat zur Folge, dass man im Baum weniger Äste abschneiden kann als das mit einer schärferen unteren Schranke der Fall wäre, da nicht ausgeschlossen werden kann, dass in ihnen noch Lösungen auftreten. Dies wiederum hat zur Folge, dass man noch in Ästen nach Lösungen sucht, von denen sich im Nachhinein herausstellt, dass in ihnen keine besseren Lösungen mehr zu finden waren. Könnte man die untere Schranke verbessern und somit dem Solver im Voraus mitteilen, dass in bestimmten Ästen keine besseren Lösungen mehr auftreten, würden diese abgeschnitten. Die Konsequenz wäre, dass man sich in diesen Ästen die Suche nach Lösungen spart, was eine Reduktion der Laufzeit mit sich bringt.

Es stellt sich die Frage, was es heißt, die größte untere Schranke zu verbessern. Die untere Schranke in jedem Knoten des Baums ist der Zielfunktionswert des zugehörigen relaxierten linearen Minimierungsproblems. Eine Verbesserung der unteren Schranke entspricht der Vergrößerung der Zielfunktionswerte der LP-Relaxierungen. Um dies zu erreichen, können wir zusätzliche Nebenbedingungen, sogenannte Cuts, zum Problem hinzufügen. Diese Nebenbedingungen dürfen dabei keine potentiellen Lösungen, also keine zulässigen Punkte des ursprünglichen linearen ganzzahligen Problems abschneiden (Zulässigkeit). Anschaulich entspricht dieses

3 Distanzminimierung für vertikale Anwendungen

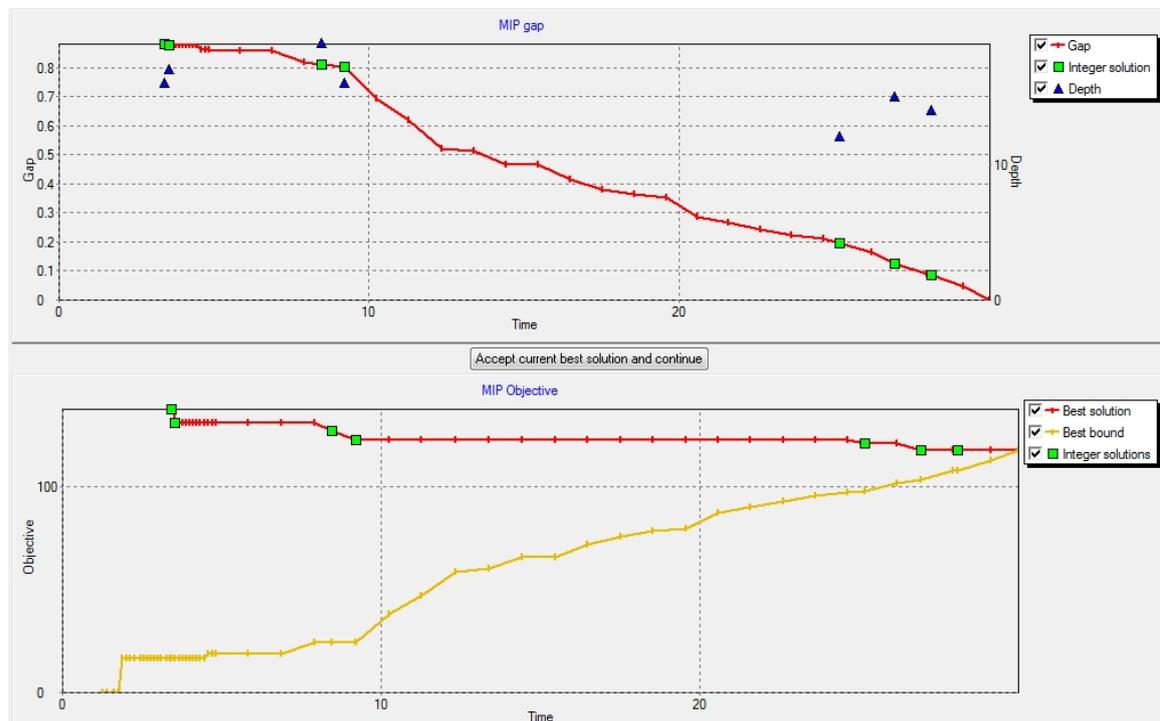


Abbildung 13: Grafische Darstellung (in IVE XPress) des Verlaufs der Optimierung von Test-Instanz 1 bei ausgeschaltetem Presolving, Heuristiken und Cutstrategien unter Verwendung von *VirtualKeyboardOptimization.mos*.

Vorgehen der Verkleinerung des Polytops ohne eine der darin liegenden potentiellen Lösungen wegzuschneiden (siehe Abb. 14).

Diese und andere mögliche Verbesserungen werden im folgenden Abschnitt erarbeitet, um einer optimalen Tastenbelegung einen Schritt näher zu kommen.

3.2.3 Ansätze zur Verbesserung der Lösungsdauer des linearisierten Problems

Im folgenden Abschnitt stehen Ansätze und Ideen zur Verbesserung der Lösbarkeit von Problem (P4) im Mittelpunkt. Dabei wird jede potentielle Verbesserung separat betrachtet. Eine Evaluation und die Entscheidung, welche Ansätze letztendlich in das Optimierungsproblem aufgenommen werden, findet sich am Ende dieses Abschnitts.

Hinzufügen von Schnittebenen - CUTS

Die grundsätzliche Idee hinter dem Hinzufügen von Schnittebenen ist es, die Beschreibung der konvexen Hülle der zulässigen Punkte von (P4) (sog. integer-hull) durch das Polytop zu verbessern (siehe Abb. 14) und so die Lösbarkeit des Problems aus zeitlicher Sicht zu erreichen.

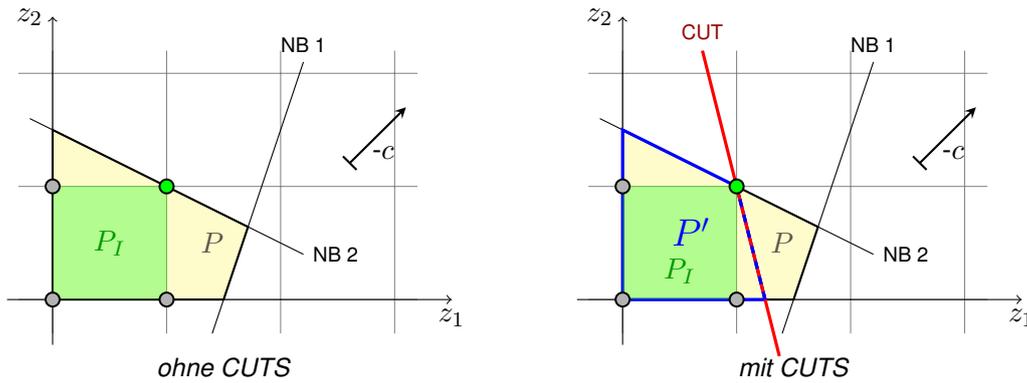


Abbildung 14: Schematische Darstellung der grundlegenden Idee des Ansatzes CUTS. Dabei sind P und P' das ursprüngliche bzw. neue, durch die (rote) Nebenbedingung entstehende Polytop und P_I die konvexe Hülle der zulässigen Punkte (grau) und der Lösung (grün).

Zu diesem Zweck ist es notwendig, die potentiellen neuen Nebenbedingungen auf ihre Zulässigkeit zu prüfen. Außerdem kann man prüfen, ob sie nützlich sind in dem Sinne, dass sie wirklich Teile des Polytops abschneiden. Im folgenden Unterabschnitt werden einige neue Nebenbedingungen untersucht.

Wir beginnen mit der Forderung, dass für jedes Buchstabenpaar $(i, k) \in [n] \times [n]$ genau ein Tastenpaar $(j, l) \in [m] \times [m]$ existiert, auf dem es liegt. Mathematisch ausgedrückt lautet dies:

$$\sum_{j=1}^m \sum_{l=1}^m x_{ijkl} = 1 \quad \forall i, k \in [n] \quad (3.10)$$

Dass diese n^2 Nebenbedingungen ihre oben genannte Aufgabe erfüllen, zeigen die folgenden Aussagen.

Lemma 3.17. Die Gleichungen (3.10) sind zulässige Cuts für das Problem (P4).

Beweis.

Zu zeigen: $(y, x) \in \mathcal{F}_{P4} \Rightarrow (y, x)$ erfüllt (3.10).

Sei $(y, x) \in \mathcal{F}_{P4}$ und angenommen (y, x) erfüllt (3.10) nicht, dann existieren $\tilde{i}, \tilde{k} \in [n]$ mit

$$\sum_{j,l=1}^m x_{\tilde{i}\tilde{j}\tilde{k}l} \neq 1. \text{ Man unterscheidet:}$$

Fall 1: $\sum_{j,l=1}^m x_{\tilde{i}\tilde{j}\tilde{k}l} < 1$

Wegen $x_{ijkl} \in \{0, 1\} \quad \forall i, j, k, l$ gilt dann $x_{\tilde{i}\tilde{j}\tilde{k}l} = 0 \quad \forall j, l$. Aus $y_{\tilde{i}j} + y_{\tilde{k}l} - 1 \leq 0 \quad \forall j, l$ folgt, dass

$y_{\tilde{i}j} = 0 \vee y_{\tilde{k}l} = 0 \quad \forall j, l$. Aber wegen $\sum_{j=1}^m y_{\tilde{i}j} = 1$ und $\sum_{l=1}^m y_{\tilde{k}l} = 1$ existieren $\tilde{j} \in [m]$ und $\tilde{l} \in [m]$

mit $y_{\tilde{i}\tilde{j}} = y_{\tilde{k}\tilde{l}} = 1$, was ein Widerspruch ist.

3 Distanzminimierung für vertikale Anwendungen

Fall 2: $\sum_{j,l=1}^m x_{\tilde{i}j\tilde{k}l} > 1$

Wegen $x_{ijkl} \in \{0, 1\} \forall i, j, k, l$ existieren $(j_1, l_1), (j_2, l_2) \in [m] \times [m]$ mit $(j_1, l_1) \neq (j_2, l_2)$ und $x_{\tilde{i}j_1\tilde{k}l_1} = x_{\tilde{i}j_2\tilde{k}l_2} = 1$. Aus den Nebenbedingungen $2 = 2 \cdot x_{\tilde{i}j_1\tilde{k}l_1} \leq y_{\tilde{i}j_1} + y_{\tilde{k}l_1}$ und $2 = 2 \cdot x_{\tilde{i}j_2\tilde{k}l_2} \leq y_{\tilde{i}j_2} + y_{\tilde{k}l_2}$ folgt, dass $y_{\tilde{i}j_1} = y_{\tilde{k}l_1} = y_{\tilde{i}j_2} = y_{\tilde{k}l_2} = 1$. Dies ist ein Widerspruch zur Nebenbedingung $\sum_{j=1}^m y_{\tilde{i}j} = 1$. □

Korollar 3.18. Sei $m = n \geq 2$, dann beschneiden die Cuts (3.10) das Polytop \mathcal{P}_{P_4} .

Beweis.

Sei $m = n \geq 2$. Zu zeigen: $\exists(\hat{y}, \hat{x}) \in \mathcal{P}_{P_4}$, welches (3.10) verletzt. Genauer gesagt ist zu zeigen, dass $\exists(\hat{y}, \hat{x}) \in \mathcal{P}_{P_4} \exists \tilde{i}, \tilde{k} \in [n] : \sum_{j,l=1}^m \hat{x}_{\tilde{i}j\tilde{k}l} \neq 1$. Man unterscheidet:

Fall 1: Wähle $\hat{y}_{ij} = \frac{1}{m} \forall i, j$ und $\hat{x}_{ijkl} = 0 \forall i, j, k, l$. Dann gilt: $\sum_{j=1}^m \hat{y}_{ij} = 1 \forall i$ und $\sum_{i=1}^n \hat{y}_{ij} = 1 \forall j$ und $\hat{y}_{ij} + \hat{y}_{kl} - 1 = \frac{2}{m} - 1 \leq 0 = \hat{x}_{ijkl}$ und $2 \cdot \hat{x}_{ijkl} = 0 \leq \frac{2}{m} = \hat{y}_{ij} + \hat{y}_{kl}$. Damit sind alle Nebenbedingungen (bis auf die Ganzzahligkeitsbedingung) erfüllt, weshalb folgt, dass $(\hat{y}, \hat{x}) \in \mathcal{P}_{P_4}$. Aber: $\sum_{j,l} \hat{x}_{ijkl} = 0 < 1 \forall i, k$, weshalb (\hat{y}, \hat{x}) von (3.10) abgeschnitten wird.

Fall 2: Wähle $\hat{y}_{ij} = \frac{1}{m} \forall i, j$ und $\hat{x}_{ijkl} = \frac{1}{m} \forall i, j, k, l$. Dann gilt: $\sum_{j=1}^m \hat{y}_{ij} = 1 \forall i$ und $\sum_{i=1}^n \hat{y}_{ij} = 1 \forall j$ und $\hat{y}_{ij} + \hat{y}_{kl} - 1 = \frac{2}{m} - 1 \leq \frac{1}{m} = \hat{x}_{ijkl}$ und $2 \cdot \hat{x}_{ijkl} = \frac{2}{m} = \hat{y}_{ij} + \hat{y}_{kl}$. Damit sind alle Nebenbedingungen (bis auf die Ganzzahligkeitsbedingung) erfüllt, weshalb folgt, dass $(\hat{y}, \hat{x}) \in \mathcal{P}_{P_4}$. Aber: $\sum_{j,l} \hat{x}_{ijkl} = m > 1 \forall i, k$, weshalb (\hat{y}, \hat{x}) von (3.10) abgeschnitten wird. □

Die zweite Gruppe von potentiellen neuen Nebenbedingungen stellt sicher, dass jedes Tastenpaar $(j, l) \in [m] \times [m]$ genau ein Buchstabenpaar $(i, k) \in [n] \times [n]$ beherbergt. Also:

$$\sum_{i=1}^n \sum_{k=1}^n x_{ijkl} = 1 \quad \forall j, l \in [m] \quad (3.11)$$

Die m^2 Nebenbedingungen haben folgende Eigenschaften.

Lemma 3.19. Die Gleichungen (3.11) sind zulässige Cuts für das Problem (P4).

Beweis.

Der Beweis funktioniert analog zum Beweis von Lemma 3.17. □

Korollar 3.20. Sei $m = n \geq 2$, dann beschneiden die Cuts (3.11) das Polytop \mathcal{P}_{P_4} .

Beweis.

Der Beweis funktioniert analog zum Beweis von Korollar 3.18. □

Um die Beschreibung der konvexen Hülle der zulässigen Punkte durch das Polytop weiter zu verbessern, kann man eine zusätzliche Information in Form einer weiteren Nebenbedingung an unser Optimierungsproblem übergeben. Betrachtet man die ursprüngliche Definition der x -Variablen, so stellt man fest, dass diese paarweise gleich sind. Es soll also gelten:

$$x_{ijkl} = x_{klij} \quad \forall i, k \in [n], j, l \in [m] \quad (3.12)$$

Bei genauem Hinsehen bemerkt man, dass jede der $m^2 n^2$ Nebenbedingungen doppelt erzeugt wird. Es genügt deshalb, nur jene Indexkombinationen zu betrachten, für die zusätzlich $i \leq k$ und $(i, j) \neq (k, l)$ gilt. Damit reduziert sich die Anzahl der Bedingungen auf $\frac{m^2}{2}(n^2 + n) - mn$.

Lemma 3.21. Die Gleichungen (3.12) sind zulässige Cuts für das Problem (P4).

Beweis.

Zu zeigen: $(y, x) \in \mathcal{F}_{P_4} \Rightarrow (y, x)$ erfüllt (3.12).

Sei $(y, x) \in \mathcal{F}_{P_4}$ und angenommen (y, x) erfüllt (3.12) nicht, dann existieren $\tilde{i}, \tilde{k} \in [n]$ und $\tilde{j}, \tilde{l} \in [m]$ mit $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \neq x_{\tilde{k}\tilde{l}\tilde{i}\tilde{j}}$. Sei o.B.d.A. $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} < x_{\tilde{k}\tilde{l}\tilde{i}\tilde{j}}$ (der Beweis für ">" funktioniert analog), dann gilt: $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0 \wedge x_{\tilde{k}\tilde{l}\tilde{i}\tilde{j}} = 1$. Wegen $2 = 2 \cdot x_{\tilde{k}\tilde{l}\tilde{i}\tilde{j}} \leq y_{\tilde{k}\tilde{l}} + y_{\tilde{i}\tilde{j}}$ und $y_{ij} \in \{0, 1\} \quad \forall i, j$ folgt $y_{\tilde{k}\tilde{l}} = y_{\tilde{i}\tilde{j}} = 1$ und damit $1 = y_{\tilde{i}\tilde{j}} + y_{\tilde{k}\tilde{l}} - 1 \leq x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0$. Dies ist ein Widerspruch. □

Korollar 3.22. Sei $m = n \geq 2$, dann beschneiden die Cuts (3.12) das Polytop \mathcal{P}_{P_4} .

Beweis.

Sei $m = n \geq 2$. Zu zeigen: $\exists(\hat{y}, \hat{x}) \in \mathcal{P}_{P_4}$, welches (3.12) verletzt. Genauer gesagt ist zu zeigen, dass $\exists(\hat{y}, \hat{x}) \in \mathcal{P}_{P_4} \exists \tilde{i}, \tilde{k} \in [n], \tilde{j}, \tilde{l} \in [m] : \hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \neq \hat{x}_{\tilde{k}\tilde{l}\tilde{i}\tilde{j}}$.

Wähle $\hat{y}_{ij} = \frac{1}{m} \quad \forall i, j$, dann gilt $\sum_{j=1}^m \hat{y}_{ij} = 1 \quad \forall i$ und $\sum_{i=1}^n \hat{y}_{ij} = 1 \quad \forall j$. Wähle außerdem eine beliebige

Indexkombination $(\tilde{i}, \tilde{j}, \tilde{k}, \tilde{l}) \in [n] \times [m] \times [n] \times [m]$ und die dazugehörige Variable $\hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = \frac{1}{m}$. Desweiteren wähle $\hat{x}_{ijkl} = 0 \quad \forall (i, j, k, l) \neq (\tilde{i}, \tilde{j}, \tilde{k}, \tilde{l})$, dann gilt $2 \cdot \hat{x}_{ijkl} \leq \frac{2}{m} = \hat{y}_{ij} + \hat{y}_{kl} \quad \forall i, j, k, l$ und $\hat{y}_{ij} + \hat{y}_{kl} - 1 = \frac{2}{m} - 1 \leq 0 \leq \hat{x}_{ijkl} \quad \forall i, j, k, l$. Damit sind alle Nebenbedingungen (außer der Ganzzahligkeitsbedingung) erfüllt, weshalb folgt, dass $(\hat{y}, \hat{x}) \in \mathcal{P}_{P_4}$. Aber: $\hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} > \hat{x}_{\tilde{k}\tilde{l}\tilde{i}\tilde{j}}$, weshalb (\hat{y}, \hat{x}) von (3.12) abgeschnitten wird (der Beweis für "<" funktioniert analog). □

3 Distanzminimierung für vertikale Anwendungen

Um weitere Cuts zu finden, kann man sich die Einführung der x -Variablen als Produkt der zugehörigen y -Variablen zurück ins Gedächtnis rufen (siehe Gleichung (3.6)). Für $(i, j) = (k, l)$ ergibt sich daraus $x_{ijij} = y_{ij}^2$, was sich dank $y_{ij} \in \{0, 1\}$ reduzieren lässt auf die lineare Nebenbedingung:

$$x_{ijij} = y_{ij} \quad \forall i \in [n], j \in [m] \quad (3.13)$$

Unser Problem würde im Falle des Hinzufügens dieser Information um mn Nebenbedingungen wachsen.

Lemma 3.23. *Die Gleichungen (3.13) sind zulässige Cuts für das Problem (P4).*

Beweis.

Zu zeigen: $(y, x) \in \mathcal{F}_{P4} \Rightarrow (y, x)$ erfüllt (3.13).

Sei $(y, x) \in \mathcal{F}_{P4}$, dann folgt der Zusammenhang $x_{ijij} \leq y_{ij} \forall i, j$ direkt aus den Nebenbedingungen $2 \cdot x_{ijkl} \leq y_{ij} + y_{kl} \forall i, j, k, l$ für $(i, k) = (j, l)$. Damit bleibt zu zeigen, dass die Bedingung $x_{ijij} \geq y_{ij}$ ein zulässiger Cut ist. Angenommen (y, x) erfüllt $[x_{ijij} \geq y_{ij} \forall i, j]$ nicht, dann $\exists \tilde{i} \in [n], \tilde{j} \in [m] : x_{\tilde{i}\tilde{j}\tilde{i}\tilde{j}} < y_{\tilde{i}\tilde{j}}$. Daraus folgt, dass $x_{\tilde{i}\tilde{j}\tilde{i}\tilde{j}} = 0 \wedge y_{\tilde{i}\tilde{j}} = 1$, was dazu führt, dass (y, x) eine Nebenbedingung von (P4) verletzt, da $y_{\tilde{i}\tilde{j}} + y_{\tilde{i}\tilde{j}} - 1 = 1 \leq x_{\tilde{i}\tilde{j}\tilde{i}\tilde{j}} = 0$. Damit ist $(y, x) \notin \mathcal{F}_{P4}$, was ein Widerspruch zur Voraussetzung ist. □

Die Ungleichungen der Form $x_{ijij} \leq y_{ij}$ verbessern die Beschreibung des Polytops nicht, da sie aufgrund der Nebenbedingungen $2 \cdot x_{ijkl} \leq y_{ij} + y_{kl}$ redundant sind. Für die Cuts $x_{ijij} \geq y_{ij}$ gilt folgendes Lemma.

Korollar 3.24. *Sei $m = n \geq 2$, dann beschneiden die Cuts $x_{ijij} \geq y_{ij}$ das Polytop \mathcal{P}_{P4} .*

Beweis.

Sei $m = n \geq 2$. Zu zeigen: $\exists (\hat{y}, \hat{x}) \in \mathcal{P}_{P4}$, für das gilt: (\hat{y}, \hat{x}) erfüllt $[x_{ijij} \geq y_{ij} \forall i, j]$ nicht.

Das heißt, es ist zu zeigen, dass $\exists (\hat{y}, \hat{x}) \in \mathcal{P}_{P4} \exists \tilde{i} \in [n], \tilde{j} \in [m] : \hat{x}_{\tilde{i}\tilde{j}\tilde{i}\tilde{j}} < \hat{y}_{\tilde{i}\tilde{j}}$.

Wähle $\hat{y}_{ij} = \frac{1}{m} \forall i, j$ und $\hat{x}_{ijkl} = 0 \forall i, j, k, l$, dann gilt $\hat{y}_{ij} + \hat{y}_{kl} - 1 = \frac{2}{m} - 1 \leq 0 = \hat{x}_{ijkl}$ und $2 \cdot \hat{x}_{ijkl} = 0 \leq \frac{2}{m} = \hat{y}_{ij} + \hat{y}_{kl}$, sowie $\sum_{j=1}^m \hat{y}_{ij} = 1 \forall i$ und $\sum_{i=1}^n \hat{y}_{ij} = 1 \forall j$. Damit sind alle Nebenbedingungen (außer der Ganzzahligkeitsbedingung) erfüllt, weshalb folgt $(\hat{y}, \hat{x}) \in \mathcal{P}_{P4}$.

Aber: $\hat{x}_{ijij} < \hat{y}_{ij} \forall i, j$, weshalb (\hat{y}, \hat{x}) von der Bedingung $x_{ijij} \geq y_{ij}$ abgeschnitten wird. □

Erinnert man sich an die zulässigen Nebenbedingungen (3.10) und (3.11), dann drängt sich die Frage auf, ob es nicht möglich ist, statt der Buchstabenpaare beziehungsweise Tastenpaare ein Buchstaben-Taste-Paar zu wählen und für die zugehörigen x -Variablen gewisse Eigenschaften

zu fordern. Man stellt fest, dass mit festem Buchstaben-Taste-Paar $(k, l) \in [n] \times [m]$ für die Summe der x_{ijkl} -Variablen folgendes gilt:

$$\sum_{i=1}^n \sum_{j=1}^m x_{ijkl} = n \cdot y_{kl} \quad \forall k \in [n], l \in [m] \quad (3.14)$$

Folgende Lemmata rechtfertigen die potentielle Verwendung dieser mn neuen Cuts.

Lemma 3.25. Die Gleichungen (3.14) sind zulässige Cuts für das Problem (P4).

Beweis.

Zu zeigen: $(y, x) \in \mathcal{F}_{P4} \Rightarrow (y, x)$ erfüllt (3.14).

Sei $(y, x) \in \mathcal{F}_{P4}$ und angenommen (y, x) erfüllt (3.14) nicht, dann existieren $\tilde{k} \in [n]$ und $\tilde{l} \in [m]$ mit $\sum_{i=1}^n \sum_{j=1}^m x_{ij\tilde{k}\tilde{l}} \neq n \cdot y_{\tilde{k}\tilde{l}}$. Man unterscheidet:

Fall 1: $\sum_{i=1}^n \sum_{j=1}^m x_{ij\tilde{k}\tilde{l}} < n \cdot y_{\tilde{k}\tilde{l}}$

Für $y_{\tilde{k}\tilde{l}} = 0$ folgt der Widerspruch aus den Bedingungen $x_{ijkl} \in \{0, 1\} \forall i, j, k, l$.

Für $y_{\tilde{k}\tilde{l}} = 1$ gilt: $y_{ij} + y_{\tilde{k}\tilde{l}} - 1 = y_{ij} \leq x_{ij\tilde{k}\tilde{l}}$ und damit $\sum_{i=1}^n \sum_{j=1}^m y_{ij} = \sum_{i=1}^n 1 = n \leq \sum_{i=1}^n \sum_{j=1}^m x_{ij\tilde{k}\tilde{l}}$.

Letzteres ist ein Widerspruch zu den Voraussetzungen.

Fall 2: $\sum_{i=1}^n \sum_{j=1}^m x_{ij\tilde{k}\tilde{l}} > n \cdot y_{\tilde{k}\tilde{l}}$

Für $y_{\tilde{k}\tilde{l}} = 0$ gilt: $\sum_{i=1}^n \sum_{j=1}^m x_{ij\tilde{k}\tilde{l}} > 0$. Daraus folgt, dass $\exists \tilde{i} \in [n], \tilde{j} \in [m] : x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 1$ und damit $2 = 2 \cdot x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \leq y_{\tilde{i}\tilde{j}} + y_{\tilde{k}\tilde{l}} \leq 1$, was ein Widerspruch ist.

Für $y_{\tilde{k}\tilde{l}} = 1$ gilt $2 \cdot x_{ij\tilde{k}\tilde{l}} \leq y_{ij} + y_{\tilde{k}\tilde{l}} \leq y_{ij} + 1$. Mit $x_{ij\tilde{k}\tilde{l}} \in \{0, 1\} \forall i, j$ folgt $x_{ij\tilde{k}\tilde{l}} \leq y_{ij} \forall i, j$.

Daraus folgt, dass $\sum_{i=1}^n \sum_{j=1}^m x_{ij\tilde{k}\tilde{l}} \leq \sum_{i=1}^n \sum_{j=1}^m y_{ij} = n$, was ein Widerspruch ist zu $\sum_{i=1}^n \sum_{j=1}^m x_{ij\tilde{k}\tilde{l}} > n$. \square

Korollar 3.26. Sei $m = n \geq 2$, dann beschneiden die Cuts (3.14) das Polytop \mathcal{P}_{P4} .

Beweis.

Sei $m = n \geq 2$. Zu zeigen: $\exists (\hat{y}, \hat{x}) \in \mathcal{P}_{P4}$, für das gilt: (\hat{y}, \hat{x}) verletzt (3.14). Also ist zu zeigen, dass $\exists (\hat{y}, \hat{x}) \in \mathcal{P}_{P4} \exists \tilde{k} \in [n], \tilde{l} \in [m] : \sum_{i=1}^n \sum_{j=1}^m \hat{x}_{ij\tilde{k}\tilde{l}} \neq n \cdot \hat{y}_{\tilde{k}\tilde{l}}$. Man unterscheidet:

Fall 1: Wähle $\hat{y}_{ij} = \frac{1}{m} \forall i, j$, dann gilt: $\sum_{j=1}^m \hat{y}_{ij} = 1 \forall i$ und $\sum_{i=1}^n \hat{y}_{ij} = 1 \forall j$. Wähle außerdem $\hat{x}_{ijkl} = 0 \forall i, j, k, l$, dann gilt zusätzlich $\hat{y}_{ij} + \hat{y}_{kl} - 1 = \frac{2}{m} - 1 \leq 0 = \hat{x}_{ijkl}$ und

3 Distanzminimierung für vertikale Anwendungen

$2 \cdot \hat{x}_{ijkl} = 0 \leq \frac{2}{m} = \hat{y}_{ij} + \hat{y}_{kl}$. Damit sind alle Nebenbedingungen (bis auf die Ganzzahligkeitsbed.) erfüllt, weshalb gilt: $(\hat{y}, \hat{x}) \in \mathcal{P}_{P_4}$. Aber: $\sum_{i=1}^n \sum_{j=1}^m \hat{x}_{ijkl} = 0 < n \cdot \hat{y}_{kl} \forall k, l$, weshalb (\hat{y}, \hat{x}) von (3.14) abgeschnitten wird.

Fall 2: Wähle $\hat{y}_{ij} = 1 \forall i = j$ und $\hat{y}_{ij} = 0 \forall i \neq j$, dann gilt $\sum_{j=1}^m \hat{y}_{ij} = 1 \forall i$ und $\sum_{i=1}^n \hat{y}_{ij} = 1 \forall j$. Wähle außerdem $\hat{x}_{ijkl} = 1 \forall i = j, k = l$ und $\hat{x}_{ijkl} = 0,5 \forall i = j, k \neq l$ und alle restlichen $\hat{x}_{ijkl} = 0$. Dann gilt: $\hat{y}_{ij} + \hat{y}_{kl} - 1 \leq \hat{x}_{ijkl} \forall i, j, k, l$ und $2 \cdot \hat{x}_{ijkl} \leq \hat{y}_{ij} + \hat{y}_{kl} \forall i, j, k, l$. Damit folgt, dass $(\hat{y}, \hat{x}) \in \mathcal{P}_{P_4}$, da alle Bedingungen (bis auf die Ganzzahligkeitsbed.) erfüllt sind. Aber: $\sum_{i=1}^n \sum_{j=1}^m \hat{x}_{ijkl} = 0,5 \cdot n > 0 = n \cdot \hat{y}_{kl} \forall k \neq l$, weshalb (\hat{y}, \hat{x}) von (3.14) abgeschnitten wird. □

Eliminierung von Symmetrien - SYMM

Ein Blick auf die Zielfunktion von (P4) offenbart, dass es außer der distanzminimierenden Positionierung von Buchstabenpaaren kein Kriterium für die Auswahl einer Tastenbelegung gibt. Dies hat zur Folge, dass viele Symmetrien innerhalb des Problems auftreten, weshalb die Menge der zulässigen Punkte eine große Teilmenge an äquivalenten Optimallösungen enthält (siehe Abb. 15). Der Grund, warum die Existenz vieler symmetrischer Lösungen problematisch sein kann, ist, dass im Lösungsvorgang unter Umständen all diese Lösungen betrachtet werden müssen. Das kostet Zeit und ließe sich durch eine Reduktion der Anzahl an symmetrischen Lösungen verhindern.

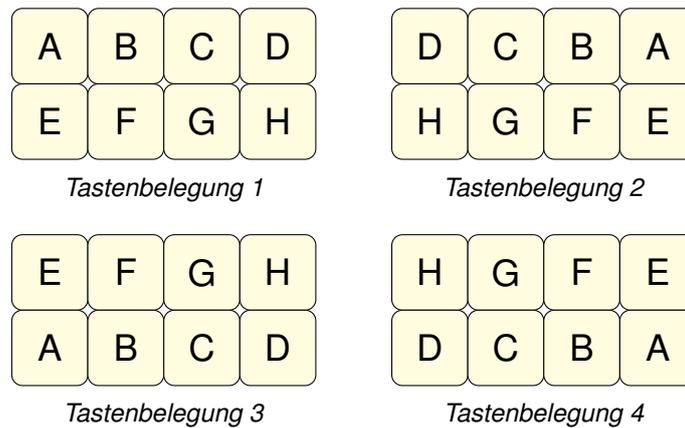


Abbildung 15: Vier symmetrische Tastenbelegungen (bezüglich der Zielfunktion (3.4) und im geometrischen Sinne).

Nebenbei ist an dieser Stelle zu bemerken, dass viele der im obigen Sinne symmetrischen Lösungen auch im klassischen, geometrischen Sinne symmetrisch sind (siehe Abb. 15). So sind für ein rechteckiges Tastaturlayout beispielsweise alle Spiegelbilder bezüglich der Koordinaten-

achsen und dem Mittelpunkt einer optimalen Tastatur auch optimale Lösungen des Problems. Um manche der Symmetrien zu vermeiden und somit die Anzahl der möglichen Lösungen zu reduzieren bietet es sich an, einen bestimmten Buchstaben $\tilde{i} \in [n]$ nur auf einem Teilbereich $\emptyset \neq T \subseteq [m]$ der Tastatur zuzulassen. Formuliert als potentielle neue Nebenbedingung sieht der Ansatz SYMM(\tilde{i}, T) wie folgt aus:

$$\sum_{j \in T} y_{\tilde{i}j} = 1 \tag{3.15}$$

Es ist anzumerken, dass (3.15) die Redundanz der Nebenbedingung bewirkt, welche ursprünglich sichergestellt hat, dass der Buchstabe \tilde{i} auf der gesamten Tastatur genau einmal vorkommt (siehe Gleichung (3.2)).

Die Idee hinter der Fixierung eines Buchstabens auf einen Teilbereich der Tastatur ist der Wunsch, positiven Einfluss auf die Größe des *Branch&Bound*-Baums zu nehmen. Durch das Hinzufügen der neuen Nebenbedingung reduziert man abhängig von T die Menge an zulässigen Punkten, indem man Teile des Polytops \mathcal{P}_{P_4} abschneidet. Damit reduziert man bei geeigneter Wahl von T auch die Menge der äquivalenten Optimallösungen, da manche der ursprünglichen Lösungen durch die neue Nebenbedingung unzulässig werden. Im Idealfall verbleiben nur wenige der Lösungen mit gleichem minimalen Zielfunktionswert (siehe Abb. 16) und man kann dank dieser Reduktion der Lösungsmenge auf eine Verbesserung der Laufzeit hoffen.

Wird der Tastaturbereich T zu klein oder falsch gewählt, so kann dies dazu führen, dass alle ursprünglichen Optimallösungen von (P4) abgeschnitten werden (siehe Abb. 16). In diesem Fall würde man als Lösung einen ursprünglich suboptimalen zulässigen Punkt erhalten. Eine richtige Wahl von T ist deshalb entscheidend für die globale Optimalität der erhaltenen Lösung. Da ein Großteil der Tastaturlayouts in Theorie und Praxis und insbesondere die Layouts aus Abschnitt 2.4.3 symmetrisch bezüglich ihrer vertikalen Achse sind und die Zielfunktion von (P4) nur die Lage der Buchstaben zueinander optimiert, aber keine Präferenz bezüglich der Positionierung eines Buchstabens auf der Tastatur enthält, kann als Zielbereich T eine Tastaturhälfte H gewählt werden. Mit dieser Wahl ist sichergestellt, dass nicht alle Optimallösungen abgeschnitten werden, da aus jedem äquivalenten (bzgl. der Zielfunktion aus (P4)), spiegelbildlich identischen (bzgl. der vertikalen Achse) Tastenbelegungspaar eine der beiden Belegungen im Zulässigkeitsbereich verbleibt. Anschaulich halbiert die Wahl $T = H$ das Polytop, die Menge an zulässigen Punkten und die Menge der symmetrischen Lösungen.

Unter bestimmten zusätzlichen Voraussetzungen wie der horizontalen Symmetrie des Layouts, könnte als Zielbereich sogar ein Tastaturviertel V gewählt werden, da aus jeder Gruppe von vier äquivalenten (bzgl. der Zielfunktion aus (P4)), spiegelbildlich identischen Tastenbelegungen, durch die Nebenbedingung (3.15) drei Belegungen unzulässig werden. Die vierte Belegung, welche (3.15) erfüllt ist nach wie vor zulässig und eine potentielle Optimallösung.

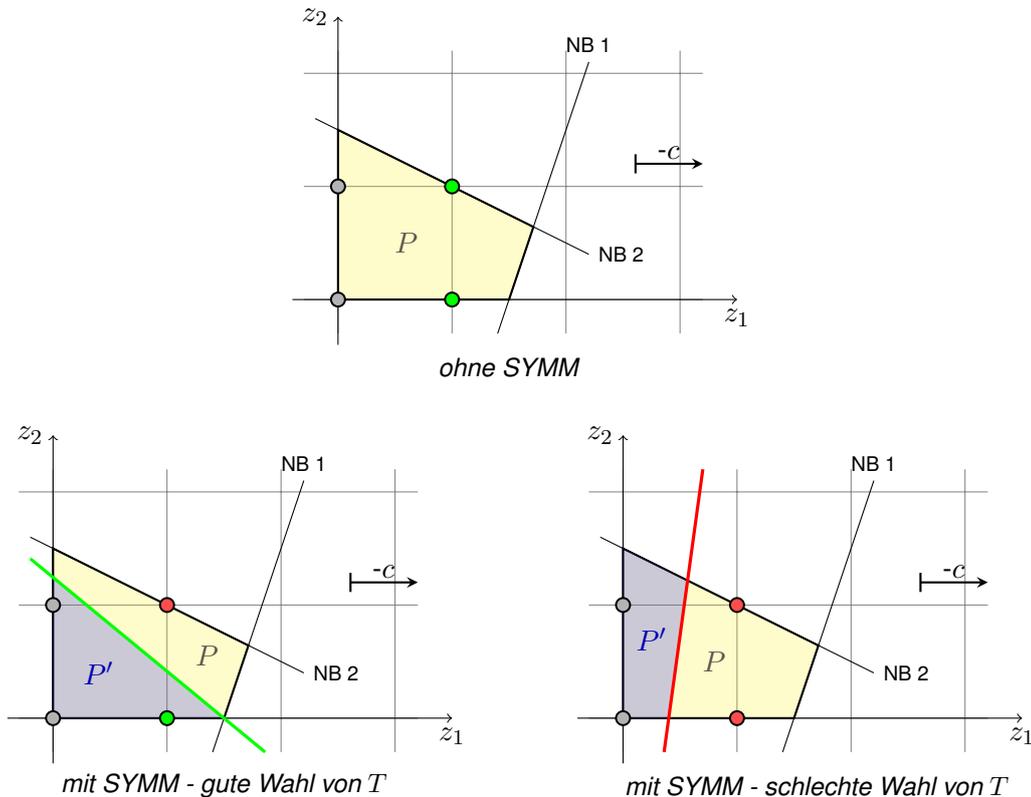


Abbildung 16: Schematische Darstellung der grundlegenden Idee des Ansatzes SYMM und Einfluss der Wahl von T auf die zusätzlichen Nebenbedingungen (rot oder grün) und die ursprüngliche Lösungsmenge (grüne Punkte). Dabei sind P und P' das ursprüngliche bzw. neue Polytop und rote Punkte ehemalige Lösungen.

Um den Ansatz $\text{SYMM}(\tilde{i}, T)$ auf alle in 2.4.3 gewählten Layouts anwenden zu können, fällt die Entscheidung auf die Fixierung eines Buchstabens \tilde{i} auf einer Tastaturhälfte. Für Layouts mit ungerader Spaltenanzahl, deren Hälften bei mittiger Teilung halbe Tasten enthalten würden, erfolgt die Teilung zugunsten einer Seite und die Fixierung von \tilde{i} erfolgt auf der größeren Hälfte. Analog zur Teilung der Layouts im horizontalen Fall (siehe Abschnitt 2.3.1) sei dies die rechte Seite, weshalb zukünftig gilt $T = H_R$.

Obige Aussagen zur möglichen Wahl von T sind auf die Verwendung der Zielfunktion aus (P4) zugeschnitten und lassen sich nicht generalisieren. Für alternative Zielfunktionen, deren äquivalente Lösungen eine andere Struktur haben und nicht den spiegelbildlichen Zusammenhang aufweisen, muss über die Wahl von T neu nachgedacht werden.

Variation der Zielfunktion - OBJFCT

Neben der Vermeidung von Symmetrien mittels einer Beschneidung des Polytops (bspw. durch SYMM) gibt es eine weitere Möglichkeit weniger oder keine symmetrischen Lösungen zu erhal-

ten. Sie basiert auf der Eigenschaft, dass der Zielfunktionsvektor senkrecht auf der konvexen Hülle der symmetrischen Lösungen steht.

Durch eine Variation des Zielfunktionsvektors versucht man diese Eigenschaft aufzuheben und zu erreichen, dass sich der Zielfunktionswert mancher zuvor optimaler Lösungen verschlechtert, sodass diese danach nicht mehr optimal sind. Anschaulich wird also ein neues Kriterium hinzugefügt, welches im besten Fall die Lösungsmenge des Problems verkleinert, indem es manche der symmetrischen Lösungen bevorzugt, andere disqualifiziert (siehe Abb. 17). In der Regel birgt dieses Vorgehen eine Gefahr, da man unter Umständen Punkte zur Lösungsmenge hinzufügt, die zuvor keine Optimallösungen waren. Um diese Problematik möglichst zu vermeiden, variiert man die Werte des Zielfunktionsvektors nur minimal, was anschaulich einer sehr kleinen Richtungsänderung entspricht.

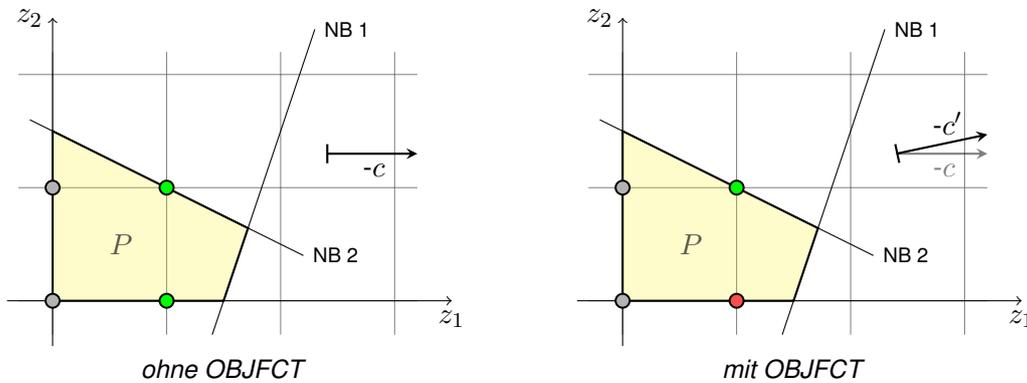


Abbildung 17: Schematische Darstellung der grundlegenden Idee des Ansatzes OBJFCT und Einfluss der Variation des Zielfunktionsvektors $(-c)$ auf die ursprüngliche Lösungsmenge (grün). P und P' sind das ursprüngliche bzw. neue Polytop und rote Punkte ehemalige Lösungen.

Im aktuellen Fall jedoch bietet sich eine wundervolle Gelegenheit diese Methode ohne Probleme anzuwenden. erinnert man sich zurück an den Beginn des Abschnitts 3.2.1. Dort wurde zunächst nur die Gesamtstrecke zwischen den Buchstabenpaaren modelliert. Es wäre aufgrund der Vergleichbarkeit zu den Ergebnissen aus Abschnitt 3.1 aber wünschenswert, die Texteingabe auch am Startpunkt beginnen und enden zu lassen. Damit muss am Anfang eines zu schreibenden Textes zuerst der Weg vom Startpunkt zum ersten Buchstaben und am Ende der Weg vom letzten Buchstaben zum Startpunkt zurückgelegt werden.

Da wir keinen konkreten Text für die Optimierung verwenden, sondern mit Häufigkeiten arbeiten und somit den ersten und letzten Buchstaben nicht kennen, errechnen wir die Länge des erwarteten Weges. Pro Text der durchschnittlichen Textlänge B muss diese Strecke zweimal (Hin- und Rückweg) zurückgelegt werden. Damit erhält man als neue Zielfunktion:

$$(B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m d_j \cdot p_i \cdot y_{ij} \quad (3.16)$$

Es bleibt festzuhalten, dass der Einfluss des zweiten Summenterms, welcher im Vergleich zur Zielfunktion von (P4) neu hinzugekommen ist, auf den Wert der Zielfunktion mit größer werdendem B abnimmt. Dieses Verhalten stimmt mit seiner intuitiven Funktion der Modellierung des Hin- und Rückwegs überein, da die beiden Wege für länger werdende Texte einen immer kleineren Teil der zurückzulegenden Gesamtstrecke einnehmen. Der Anteil des ersten Terms am Zielfunktionswert wird mit wachsendem B sinnvollerweise größer, da er die Gesamtstrecke für die Übertragung der $B - 1$ Bigramme modelliert.

Die Wahl des Startpunktes und die Wahl des Tastaturbereichs T , auf dem ein Buchstabe \tilde{i} fixiert wird, entscheiden über die Kompatibilität der Ansätze $\text{SYMM}(\tilde{i}, T)$ und OBJFCT . Der Grund dafür ist, dass die Qualität einer Tastenbelegung bei Verwendung der neuen Zielfunktion (3.16) an den Startpunkt des Fingers gekoppelt ist, während die ursprüngliche Zielfunktion (aus Problem (P4)) nur die Lage von Bigrammen untereinander optimiert, nicht jedoch deren Position auf der Tastatur. So kann es vorkommen, dass eine bestimmte Wahl von T , die für (P4) geeignet war, die ursprünglich äquivalenten Lösungen abschneidet, die neuerdings von (3.16) bevorzugt werden. Dies würde zu einer suboptimalen Tastenbelegung führen. Die Problematik wird deutlich, wenn in Abbildung 16 statt dem Zielfunktionsvektor $(1, 0)^T$ aufgrund von OBJFCT der Vektor $(1, \epsilon)^T$ verwendet wird. Dann wäre der Punkt $(1, 1)$ die Lösung des Problems (ohne SYMM , mit OBJFCT), welcher durch die zusätzliche Anwendung von SYMM jedoch in beiden Fällen abgeschnitten wird. Insgesamt existieren in (P4) mehr symmetrische Lösungen und die Wahl von T ist deutlich weniger eingeschränkt, als dies bei Verwendung von (3.16) als neue Zielfunktion der Fall ist.

Für die konkrete Wahl des vertikalen Startpunktes (mittig, am oberen Rand - siehe Abschnitt 2.4.4) und die Fixierung auf eine Tastaturhälfte H sind die Ansätze OBJFCT und $\text{SYMM}(\tilde{i}, T)$ kompatibel. Der Grund dafür ist, dass OBJFCT bei mittiger Startpunktwahl nicht alle äquivalenten Lösungen aus (P4) eliminiert, sondern ein Tastenbelegungspaar in der Lösungsmenge verbleibt, das symmetrisch (bzgl. vertikalen Achse) ist. An diesem Punkt setzt $\text{SYMM}(\tilde{i}, H)$ an und eliminiert eine der beiden Belegungen.

Die Vorteile der potentiellen Einführung der neuen Zielfunktion (3.16) liegen auf der Hand. Eine exaktere Modellierung der realistischen Schreibweise, sowie eine erhoffte Verbesserung der Laufzeit durch eine Reduktion der Anzahl an symmetrischen Lösungen.

Relaxierung der oberen-Schranken-Bedingungen - UPBND

Wie zuvor bereits erwähnt, ist man an einer möglichst genauen Beschreibung der konvexen Hülle der zulässigen Punkte von (P4) durch das Polytop \mathcal{P}_{P4} interessiert. Besonders wichtig ist dabei eine gute Beschreibung in Zielfunktionsrichtung, weshalb man insbesondere an Nebenbedingungen interessiert ist, die Teile des Polytops in Zielfunktionsrichtung abschneiden. Aus diesem Grund kann man über eine Reduktion des Problems um jene Nebenbedingungen

nachdenken, welche an anderer Stelle ansetzen (siehe Abb. 18).

Dies wäre in mehrfacher Hinsicht eine Hilfestellung für das Problem (P4). Einerseits schrumpft durch das Vernachlässigen von Nebenbedingungen die Größe des Problems. Andererseits vergrößern wir den Zulässigkeitsbereich des Problems, was die Suche nach einem zulässigen Punkt, dessen Funktionswert als obere Schranke an die Optimallösung im *Branch&Bound*-Verfahren dient, vereinfachen sollte. Darüber hinaus findet eine Vergrößerung des Polytops statt, sodass die untere Schranke in jedem Knoten des Branch&Bound-Verfahrens schneller bestimmt werden kann. Der Grund dafür ist, dass der Algorithmus, der das relaxierte Problem in jedem Knoten löst, schneller einen Punkt im Polytop findet, der als Ausgangspunkt der Optimierung dient.

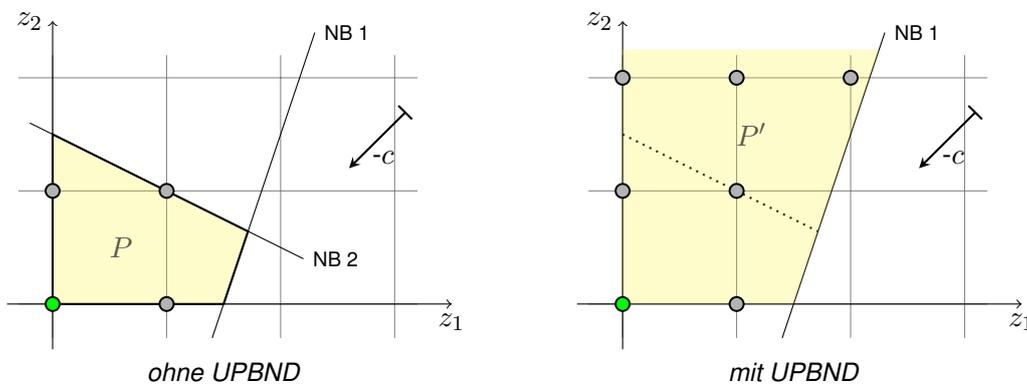


Abbildung 18: Schematische Darstellung der grundlegenden Idee des Ansatzes UPBND. P' ist das neue Polytop, welches aus dem ursprünglichen Polytop P durch die Relaxierung (von NB 2) entsteht. Zulässige Punkte sind grau, Lösungen grün.

In unserem Fall sind die Nebenbedingungen, die das Polytop nicht in Zielfunktionsrichtung beschränken, leicht zu identifizieren. Es sind die Ungleichungen der Form $2 \cdot x_{ijkl} \leq y_{ij} + y_{kl}$, welche x_{ijkl} von oben beschränken. Ihr Vernachlässigen führt zu folgendem linearen, ganzzahligen Optimierungsproblem, welches eine Relaxierung von (P4) darstellt:

$$\begin{aligned}
 & \min (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} \\
 \text{u.d.N.} & \sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \\
 & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \\
 & y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall i, k \in [n], j, l \in [m] \\
 & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \\
 & x_{ijkl} \in \{0, 1\} \quad \forall i, k \in [n], j, l \in [m]
 \end{aligned} \tag{P5}$$

Die Anzahl der Nebenbedingung hat sich im Vergleich zu (P4) reduziert auf $m^2n^2 + m + n$, die Anzahl der Variablen hat sich nicht verändert.

Über die Beziehung der Zulässigkeitsbereiche der Probleme (P4) und (P5) und ihrer Polytope gibt folgendes Lemma Auskunft.

Lemma 3.27. *Es gilt:*

1. $\mathcal{F}_{P_4} \subseteq \mathcal{F}_{P_5}$ und $\mathcal{P}_{P_4} \subseteq \mathcal{P}_{P_5}$.
2. $\mathcal{F}_{P_5} \neq \emptyset$.
3. Für $m = n = 1$ gilt: $\mathcal{F}_{P_5} = \mathcal{P}_{P_5} = \{(1, 1)\}$.
4. Für $m = n \geq 2$ gilt: $\mathcal{F}_{P_4} \subset \mathcal{F}_{P_5}$ und $\mathcal{P}_{P_4} \subset \mathcal{P}_{P_5}$.

Beweis.

Zu 1.: Der Zusammenhang $\mathcal{F}_{P_4} \subseteq \mathcal{F}_{P_5}$ folgt aus der Tatsache, dass die Menge der Nebenbedingungen von Problem (P5) eine Teilmenge der Nebenbedingungen aus (P4) ist. Denn sei $(y, x) \in \mathcal{F}_{P_4}$ und angenommen $(y, x) \notin \mathcal{F}_{P_5}$, dann existiert eine Nebenbedingung in (P5), welche von (y, x) verletzt wird. Diese Bedingung wäre auch in (P4) verletzt, weshalb $(y, x) \notin \mathcal{F}_{P_4}$, was ein Widerspruch zur Voraussetzung ist.

Analoges gilt für die Menge der Nebenbedingungen der LP-relaxierten Probleme, weshalb $\mathcal{P}_{P_4} \subseteq \mathcal{P}_{P_5}$ gilt.

Zu 2.: Die Behauptung folgt aus $\mathcal{F}_{P_4} \neq \emptyset$ (Aussage 1 in Korollar 3.12) und Aussage 1 dieses Lemmas.

Zu 3.: Die Aussage erhält man durch das Aufstellen der Nebenbedingungen von (P5) oder seiner LP-Relaxierung (für $m = n = 1$).

Zu 4.: Sei $m = n \geq 2$, dann bleibt dank Aussage 1 dieses Lemmas zu zeigen, dass:

(i) $\exists(\hat{y}, \hat{x}) \in \mathcal{F}_{P_5}$ mit $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P_4}$ und (ii) $\exists(\hat{y}, \hat{x}) \in \mathcal{P}_{P_5}$ mit $(\hat{y}, \hat{x}) \notin \mathcal{P}_{P_4}$.

Betrachte (\hat{y}, \hat{x}) mit $\hat{y}_{ij} = 1 \forall i = j$ und $\hat{y}_{ij} = 0 \forall i \neq j$ und $\hat{x}_{ijkl} = 1 \forall i, j, k, l$. Dann gilt $\hat{y}_{ij} + \hat{y}_{kl} - 1 \leq \hat{x}_{ijkl} \forall i, j, k, l$ und $\sum_{j=1}^m \hat{y}_{ij} = 1 \forall i$ und $\sum_{i=1}^n \hat{y}_{ij} = 1 \forall j$. Daraus folgt: $(\hat{y}, \hat{x}) \in \mathcal{F}_{P_5}$.

Wähle $\tilde{i}, \tilde{k} \in [n]$ und $\tilde{l} \in [m]$ mit $\tilde{k} \neq \tilde{l}$ (Existenz wg. $m = n \geq 2$), dann gilt, dass

$2 \cdot \hat{x}_{\tilde{i}\tilde{k}\tilde{l}} = 2 > 1 + 0 = \hat{y}_{\tilde{i}\tilde{i}} + \hat{y}_{\tilde{k}\tilde{l}}$ (*) und somit $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P_4}$. Außerdem gilt: $(\hat{y}, \hat{x}) \in \mathcal{F}_{P_5} \subseteq \mathcal{P}_{P_5}$ und mit (*), dass $(\hat{y}, \hat{x}) \notin \mathcal{P}_{P_4}$. Insgesamt erfüllt (\hat{y}, \hat{x}) also die Behauptungen (i) und (ii). □

Lemma 3.28. *Es gilt: $(y^*, x^*) \in \mathcal{L}_{P_4} \Rightarrow (y^*, x^*) \in \mathcal{L}_{P_5}$.*

Beweis.

Sei $(y^*, x^*) \in \mathcal{L}_{P_4}$, dann gilt:
$$\begin{cases} (i) & (y^*, x^*) \in \mathcal{F}_{P_4} \\ (ii) & f_{P_4}(y^*, x^*) \leq f_{P_4}(y, x) \quad \forall (y, x) \in \mathcal{F}_{P_4} \end{cases}$$

Mit (i) und Aussage 1 aus Lemma 3.27 folgt, dass $(y^*, x^*) \in \mathcal{F}_{P_5}$.

Angenommen $(y^*, x^*) \notin \mathcal{L}_{P_5}$, dann $\exists(\hat{y}, \hat{x}) \in \mathcal{F}_{P_5} : f_{P_5}(\hat{y}, \hat{x}) < f_{P_5}(y^*, x^*)$.

Wegen $f_{P_4} = f_{P_5}$ gilt dann aber, dass $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P_4}$ (sonst wäre $(y^*, x^*) \notin \mathcal{L}_{P_4}$).

Daraus folgt, dass $\tilde{i}, \tilde{k} \in [n]$ und $\tilde{j}, \tilde{l} \in [m]$ existieren mit $2 \cdot \hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} > \hat{y}_{\tilde{i}\tilde{j}} + \hat{y}_{\tilde{k}\tilde{l}}$.

Wegen $\hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \in \{0, 1\}$, $\hat{y}_{\tilde{i}\tilde{j}} \in \{0, 1\}$, $\hat{y}_{\tilde{k}\tilde{l}} \in \{0, 1\}$ gilt $\hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 1$ und $\hat{y}_{\tilde{i}\tilde{j}} + \hat{y}_{\tilde{k}\tilde{l}} \leq 1$.

Wähle (\tilde{y}, \tilde{x}) mit $\tilde{y}_{ij} = \hat{y}_{ij} \quad \forall i, j$ und $\tilde{x}_{ijkl} = \hat{x}_{ijkl} \quad \forall (i, j, k, l) \neq (\tilde{i}, \tilde{j}, \tilde{k}, \tilde{l})$ und $\tilde{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0$. Dann gilt $\tilde{y}_{\tilde{i}\tilde{j}} + \tilde{y}_{\tilde{k}\tilde{l}} - 1 = \hat{y}_{\tilde{i}\tilde{j}} + \hat{y}_{\tilde{k}\tilde{l}} - 1 \leq 0 = \tilde{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}$ und $2 \cdot \tilde{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0 \leq \tilde{y}_{\tilde{i}\tilde{j}} + \tilde{y}_{\tilde{k}\tilde{l}}$, weshalb $(\tilde{y}, \tilde{x}) \in \mathcal{F}_{P_4}$.

Mit $\tilde{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0 < 1 = \hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}$ gilt $f_{P_5}(\tilde{y}, \tilde{x}) = \sum_{ijkl} \underbrace{p_{ik} \cdot d_{jl}}_{\geq 0} \cdot \tilde{x}_{ijkl} \leq \sum_{ijkl} p_{ik} \cdot d_{jl} \cdot \hat{x}_{ijkl} = f_{P_5}(\hat{y}, \hat{x})$.

Daraus folgt $f_{P_5}(\tilde{y}, \tilde{x}) < f_{P_5}(y^*, x^*)$. Mit $f_{P_4} = f_{P_5}$ folgt der Widerspruch zu (ii). □

Korollar 3.29. Es gilt: $\mathcal{L}_{P_5} \neq \emptyset$.

Beweis.

Die Behauptung folgt aus $\mathcal{L}_{P_4} \neq \emptyset$ (Korollar 3.16) und Lemma 3.28. □

Um die Verwendung von (P5) für die Suche nach einer optimalen Tastenbelegung zu rechtfertigen, ist man neben der Existenz einer Lösung für (P5) an der Rückrichtung von Lemma 3.28 interessiert. Diese gilt jedoch im Allgemeinen nicht, aber es kann folgende Eigenschaft der Lösungsmenge des Problems (P5) festgestellt werden.

Lemma 3.30. Sei $d_{jl} \cdot p_{ik} > 0 \quad \forall i, k \in [n], j, l \in [m]$, dann gilt: $(y^*, x^*) \in \mathcal{L}_{P_5} \Rightarrow (y^*, x^*) \in \mathcal{L}_{P_4}$.

Beweis.

Sei $d_{jl} \cdot p_{ik} > 0 \quad \forall i, k \in [n], j, l \in [m]$ und $(y^*, x^*) \in \mathcal{L}_{P_5}$.

Aus Letzterem folgt, dass
$$\begin{cases} (i) & (y^*, x^*) \in \mathcal{F}_{P_5} \\ (ii) & f_{P_5}(y^*, x^*) \leq f_{P_5}(y, x) \quad \forall (y, x) \in \mathcal{F}_{P_5} \end{cases}$$

Angenommen (y^*, x^*) ist keine Lösung von (P4), dann unterscheide:

Fall 1: $(y^*, x^*) \in \mathcal{F}_{P_4}$, aber $\exists(\hat{y}, \hat{x}) \in \mathcal{F}_{P_4}$ mit $f_{P_4}(\hat{y}, \hat{x}) < f_{P_4}(y^*, x^*)$.

Aus Lemma 3.27 folgt, dass $(\hat{y}, \hat{x}) \in \mathcal{F}_{P_5}$. Mit $f_{P_4} = f_{P_5}$ folgt der Widerspruch zu (ii).

Fall 2: $(y^*, x^*) \notin \mathcal{F}_{P_4}$.

Dann existieren $\tilde{i}, \tilde{k} \in [n]$, $\tilde{j}, \tilde{l} \in [m] : 2 \cdot x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* > y_{\tilde{i}\tilde{j}}^* + y_{\tilde{k}\tilde{l}}^*$.

Analog zum Beweis von Lemma 3.28 lässt sich ein $(\tilde{y}, \tilde{x}) \in \mathcal{F}_{P_4}$ konstruieren, für das gilt

3 Distanzminimierung für vertikale Anwendungen

$$f_{P5}(\tilde{y}, \tilde{x}) = \sum_{ijkl} \underbrace{p_{ik} \cdot d_{jl}}_{>0} \cdot \tilde{x}_{ijkl} < \sum_{ijkl} p_{ik} \cdot d_{jl} \cdot x_{ijkl}^* = f_{P5}(y^*, x^*) \Rightarrow \text{Widerspruch zu (ii)}. \quad \square$$

Betrachtet man die Erkenntnisse, die man bislang für den Ansatz OBJFCT sammeln konnte, so stellt man fest, dass das Problem (P5) eine Lösung des Problems (P4) und damit eine optimale Tastenbelegung liefert, wenn die Koeffizienten aller Summanden in der Zielfunktion echt größer Null sind (siehe Lemma 3.30). Anschaulich bedeutet das, dass die obere Schranke (Bedingung (3.8)) keine Lösungen von (P5) abschneidet, wenn alle x -Variablen in die Zielfunktion einfließen. Diese Eigenschaft ist jedoch nicht erfüllt, da die Entfernungen $d_{jj} = 0 \forall j \in [m]$ sind, weshalb es weiterhin herauszufinden gilt, ob dieser potentielle Verbesserungsansatz für die Suche nach einer optimalen Tastenbelegung geeignet ist.

Lemma 3.31. *Bezeichne mit f_{P4} und f_{P5} die identischen Zielfunktionen der Probleme (P4) und (P5) und definiere $f := f_{P4} = f_{P5}$. Sei $(y_1^*, x_1^*) \in \mathcal{L}_{P4}$ und $(y_2^*, x_2^*) \in \mathcal{L}_{P5}$, dann gilt $f(y_1^*, x_1^*) = f(y_2^*, x_2^*)$*

Beweis.

Sei $(y_1^*, x_1^*) \in \mathcal{L}_{P4}$ und $(y_2^*, x_2^*) \in \mathcal{L}_{P5}$, dann gilt $(y_1^*, x_1^*) \in \mathcal{F}_{P4}$ und $(y_2^*, x_2^*) \in \mathcal{F}_{P5}$. Wegen $\mathcal{F}_{P4} \subseteq \mathcal{F}_{P5}$ (Lemma 3.27) gilt $(y_1^*, x_1^*) \in \mathcal{F}_{P5}$ und $f(y_1^*, x_1^*) \geq f(y_2^*, x_2^*)$. Bleibt zu zeigen, dass $f(y_1^*, x_1^*) \leq f(y_2^*, x_2^*)$. Angenommen $f(y_1^*, x_1^*) > f(y_2^*, x_2^*)$, dann ist $(y_1^*, x_1^*) \notin \mathcal{L}_{P5}$, was im Widerspruch zu Lemma 3.28 steht. □

Dank Lemma 3.31 weiß man, dass das resultierende Problem (P5) in jedem Fall eine Lösung mit identischem Zielfunktionswert (wie (P4)) liefert. Es stellt sich die Frage, ob und wie aus einer Lösung von (P5) die zugehörige Tastenbelegung ausgelesen werden kann.

Lemma 3.32. *Sei $(y^*, x^*) \in \mathcal{L}_{P5}$, dann gilt: $x_{ijkl}^* \geq y_{ij}^* \cdot y_{kl}^* \forall i, k \in [n], j, l \in [m]$.*

Beweis.

Sei (y^*, x^*) Lösung von (P5). Angenommen es gilt nicht $x_{ijkl}^* \geq y_{ij}^* \cdot y_{kl}^* \forall i, j, k, l$, dann existieren $\tilde{i}, \tilde{k} \in [n], \tilde{j}, \tilde{l} \in [m]$ mit $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* < y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^*$ und man unterscheidet:

Fall 1: $y_{\tilde{i}\tilde{j}}^* = 0 \vee y_{\tilde{k}\tilde{l}}^* = 0$. Dann folgt $y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^* = 0$ und somit $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* < 0$, was ein Widerspruch zu $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* \in \{0, 1\}$ ist.

Fall 2: $y_{\tilde{i}\tilde{j}}^* = 1 \wedge y_{\tilde{k}\tilde{l}}^* = 1$. Dann folgt $y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^* = 1$ und somit $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* < 1 = y_{\tilde{i}\tilde{j}}^* + y_{\tilde{k}\tilde{l}}^* - 1$. Damit verletzt (y^*, x^*) eine Nebenbedingung von (P5), was ein Widerspruch zur Voraussetzung ist. □

Lemma 3.33. *Sei $(y^*, x^*) \in \mathcal{L}_{P5}$. Für $\tilde{i}, \tilde{k} \in [n], \tilde{j}, \tilde{l} \in [m]$ mit $d_{\tilde{j}\tilde{l}} \cdot p_{\tilde{i}\tilde{k}} > 0$ gilt: $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* = y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^*$.*

Beweis.

Sei $(y^*, x^*) \in \mathcal{L}_{P5}$ und für $\tilde{i}, \tilde{k} \in [n]$, $\tilde{j}, \tilde{l} \in [m]$ gelte $d_{\tilde{j}\tilde{l}} \cdot p_{\tilde{i}\tilde{k}} > 0$. Mit Lemma 3.32 bleibt zu zeigen, dass dann gilt: $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* \leq y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^*$. Angenommen es gilt $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* > y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^*$, dann folgt mit $y_{ij} \in \{0, 1\}$ und $x_{ijkl} \leq 1$, dass $y_{\tilde{i}\tilde{j}}^* = 0 \vee y_{\tilde{k}\tilde{l}}^* = 0$. Daraus folgt $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* > 0$ und $y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^* = 0$.

Wähle (\hat{y}, \hat{x}) mit $\hat{y}_{ij} = y_{ij}^* \forall i, j$ und $\hat{x}_{ijkl} = x_{ijkl}^* \forall (i, j, k, l) \neq (\tilde{i}, \tilde{j}, \tilde{k}, \tilde{l})$ und $\hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^* = 0$. Dann folgt $\hat{y}_{\tilde{i}\tilde{j}} + \hat{y}_{\tilde{k}\tilde{l}} - 1 = y_{\tilde{i}\tilde{j}}^* + y_{\tilde{k}\tilde{l}}^* - 1 \leq 0 = \hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}$ und damit $(\hat{y}, \hat{x}) \in \mathcal{F}_{P5}$.

Außerdem gilt mit $\hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0 < x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^*$ und $d_{\tilde{j}\tilde{l}} \cdot p_{\tilde{i}\tilde{k}} > 0$, dass $d_{\tilde{j}\tilde{l}} \cdot p_{\tilde{i}\tilde{k}} \cdot \hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} < d_{\tilde{j}\tilde{l}} \cdot p_{\tilde{i}\tilde{k}} \cdot x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^*$ und damit $f_{P5}(\hat{y}, \hat{x}) < f_{P5}(y^*, x^*)$. Dies ist ein Widerspruch zu $(y^*, x^*) \in \mathcal{L}_{P5}$. □

Korollar 3.34. Sei $(y^*, x^*) \in \mathcal{L}_{P5}$ und $\tilde{i}, \tilde{k} \in [n]$, $\tilde{j}, \tilde{l} \in [m]$ mit $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* > y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^*$, dann gilt: $d_{\tilde{j}\tilde{l}} \cdot p_{\tilde{i}\tilde{k}} = 0$.

Beweis.

Das Korollar ist die Kontraposition der Aussage aus Lemma 3.33. □

Dank Lemma 3.13 weiß man, dass für alle zulässigen Punkte aus (P4) und damit auch für alle Lösungen der Zusammenhang $x_{ijkl} = y_{ij} \cdot y_{kl}$ gilt. Aus Lemma 3.33 weiß man, dass derselbe Zusammenhang in (P5) für alle x -Variablen gilt, deren Lösungswert den Zielfunktionswert beeinflusst. Insofern überrascht die Aussage, dass die Zielfunktionswerte der Probleme (P4) und (P5) übereinstimmen (siehe Lemma 3.31), nicht.

Mit dem Wissen, dass $x_{ijkl} = y_{ij} \cdot y_{kl} \forall i, j, k, l$ gelten muss, um die Tastenbelegung aus den x -Variablen auslesen zu können (siehe (3.6)), gelangt man mithilfe der Aussagen aus Lemma 3.32, Lemma 3.33 und Korollar 3.34 zu folgender Erkenntnis: x -Variablen, deren Lösungswert den Zielfunktionswert beeinflusst, geben auch bei Verwendung von UPBND in der Lösung die korrekte Belegung eines Tastenpaars wieder. Für x_{ijkl} deren Zielfunktionskoeffizienten gleich Null sind, ist dies nicht notwendigerweise der Fall (im Gegensatz zu (P3) und (P4)). Ist man sich dieses Umstandes bewusst und rekonstruiert in der Folge die Tastenbelegung aus den y -Variablen, so steht einer Verwendung von UPBND nichts im Weg.

Bisher wurde die Zulässigkeit und die Nützlichkeit aller Cuts nur für das Problem (P4) bewiesen. Nun stellt sich die Frage, ob diese auch für das Problem (P5) zulässig und hilfreich im Bezug auf die bessere Beschreibung der konvexen Hülle der zulässigen Punkte sind. Da das relaxierte Problem (P5) einen größeren Zulässigkeitsbereich hat als seine ursprüngliche Formulierung (P4), sind die Cuts nicht automatisch zulässig. Folgende Lemmata widmen sich der Zulässigkeit der Cuts für Problem (P5).

Lemma 3.35. Folgende Nebenbedingungen sind zulässig für das Problem (P5):

3 Distanzminimierung für vertikale Anwendungen

1. $\sum_{i,k=1}^n x_{ijkl} \geq 1 \quad \forall j, l \in [m]$
2. $\sum_{j,l=1}^m x_{ijkl} \geq 1 \quad \forall i, k \in [n]$
3. $x_{ijij} \geq y_{ij} \quad \forall i \in [n], j \in [m]$
4. $\sum_{i=1}^n \sum_{j=1}^m x_{ijkl} \geq n \cdot y_{kl} \quad \forall k \in [n], l \in [m]$

Beweis.

Die Beweise funktionieren analog zu den Zulässigkeitsbeweisen der jeweiligen Cuts für Problem (P4) (siehe Beweise 3.17, 3.19, 3.23 und 3.25).

□

Korollar 3.36. Sei $m = n \geq 2$, dann beschneiden die Cuts aus Lemma 3.35 das Polytop \mathcal{P}_{P5} .

Beweis.

Die Aussage folgt aus Punkt 1 von Lemma 3.27 und den Korollaren 3.18, 3.20, 3.24 und 3.26.

□

Nun stellt sich die Frage, welchen Einfluss die Relaxierung auf die Eigenschaften der verbleibenden Cuts hatte. Darüber informiert folgendes Lemma.

Lemma 3.37. Sei $m, n \geq 2$, dann sind folgende Nebenbedingungen nicht zulässig für das Problem (P5):

1. $\sum_{i,k=1}^n x_{ijkl} \leq 1 \quad \forall j, l \in [m]$
2. $\sum_{j,l=1}^m x_{ijkl} \leq 1 \quad \forall i, k \in [n]$
3. $x_{ijij} \leq y_{ij} \quad \forall i \in [n], j \in [m]$
4. $x_{ijkl} = x_{klij} \quad \forall i, k \in [n], j, l \in [m]$
5. $\sum_{i=1}^n \sum_{j=1}^m x_{ijkl} \leq n \cdot y_{kl} \quad \forall k \in [n], l \in [m]$

Beweis.

Zu 1.: Zu zeigen: $\exists (\hat{y}, \hat{x}) \in \mathcal{F}_{P5} \exists \tilde{j}, \tilde{l} \in [m] : \sum_{i,k=1}^n x_{i\tilde{j}k\tilde{l}} > 1$.

Wähle $\hat{y}_{ij} = 1 \forall i = j$ und $\hat{y}_{ij} = 0 \forall i \neq j$ und $\hat{x}_{ijkl} = 1 \forall i, j, k, l$, dann gilt $\hat{y}_{ij} \in \{0, 1\} \forall i, j$ und

$$\hat{x}_{ijkl} \in \{0, 1\} \quad \forall i, j, k, l \quad \text{und} \quad \sum_{i=1}^n \hat{y}_{ij} = 1 \quad \forall j \quad \text{und} \quad \sum_{j=1}^m \hat{y}_{ij} = 1 \quad \forall i \quad \text{und} \quad \hat{y}_{ij} + \hat{y}_{kl} - 1 \leq \hat{x}_{ijkl} \quad \forall i, j, k, l.$$

Damit ist $(\hat{y}, \hat{x}) \in \mathcal{F}_{P5}$ und für beliebige $\tilde{j}, \tilde{l} \in [m]$ gilt $\sum_{i,k=1}^n x_{i\tilde{j}k\tilde{l}} = n^2 > 1$.

Zu 2.: Zu zeigen: $\exists (\hat{y}, \hat{x}) \in \mathcal{F}_{P5} \exists \tilde{i}, \tilde{k} \in [n] : \sum_{j,l=1}^m x_{i\tilde{j}k\tilde{l}} > 1$.

Wähle (\hat{y}, \hat{x}) wie in 1., dann gilt $(\hat{y}, \hat{x}) \in \mathcal{F}_{P5}$ und für beliebige $\tilde{i}, \tilde{k} \in [n]$ gilt $\sum_{j,l=1}^m x_{i\tilde{j}k\tilde{l}} = n^2 > 1$.

Zu 3.: Zu zeigen: $\exists (\hat{y}, \hat{x}) \in \mathcal{F}_{P5} \exists \tilde{i} \in [n], \tilde{j} \in [m] : \hat{x}_{i\tilde{j}\tilde{i}\tilde{j}} > \hat{y}_{i\tilde{j}}$.

Wähle (\hat{y}, \hat{x}) wie in 1., dann gilt $(\hat{y}, \hat{x}) \in \mathcal{F}_{P5}$ und $1 = \hat{x}_{i\tilde{j}\tilde{i}\tilde{j}} > \hat{y}_{i\tilde{j}} = 0$ für alle $\tilde{i} \neq \tilde{j}$.

Zu 4.: Zu zeigen: $\exists (\hat{y}, \hat{x}) \in \mathcal{F}_{P5} \exists \tilde{i}, \tilde{k} \in [n], \tilde{j}, \tilde{l} \in [m] : x_{i\tilde{j}\tilde{k}\tilde{l}} \neq x_{k\tilde{l}\tilde{i}\tilde{j}}$.

Wähle \hat{y} wie in 1. und $\hat{x}_{ijkl} = 1 \quad \forall i = j, k = l$. Alle \hat{x}_{ijkl} mit $i \neq j$ oder $k \neq l$ wähle so, dass $\hat{x}_{ijkl} = 1 \quad \forall i \leq k$ und $\hat{x}_{ijkl} = 0 \quad \forall i > k$, dann folgt $(\hat{y}, \hat{x}) \in \mathcal{F}_{P5}$. Wähle $\tilde{i}, \tilde{k} \in [n]$ mit $\tilde{i} < \tilde{k}$, dann gilt $1 = \hat{x}_{i\tilde{j}\tilde{k}\tilde{l}} > \hat{x}_{k\tilde{l}\tilde{i}\tilde{j}} = 0$ (der Beweis für " $<$ " funktioniert analog).

Zu 5.: Zu zeigen: $\exists (\hat{y}, \hat{x}) \in \mathcal{F}_{P5} \exists \tilde{k} \in [n], \tilde{l} \in [m] : \sum_{i=1}^n \sum_{j=1}^m x_{i\tilde{j}k\tilde{l}} > n \cdot y_{k\tilde{l}}$.

Wähle (\hat{y}, \hat{x}) wie in 1., dann ist $(\hat{y}, \hat{x}) \in \mathcal{F}_{P5}$. Wähle außerdem beliebige $\tilde{k} \in [n], \tilde{l} \in [m]$, dann gilt $\sum_{i=1}^n \sum_{j=1}^m x_{i\tilde{j}k\tilde{l}} = n^2 > n \cdot y_{k\tilde{l}}$.

□

Eine interessante Erkenntnis in diesem Zusammenhang ist, dass selbst die unzulässigen Cuts aus Lemma 3.37 nie alle Lösungen von (P5) abschneiden. Dies wird in folgendem Lemma deutlich.

Lemma 3.38. Sei $a \in \mathbb{R}^{m^2n^2+mn}, \beta \in \mathbb{R}$ und $\langle a, (y, x) \rangle \leq \beta$ zulässiger Cut für das Problem (P4), dann gilt: Es existiert eine Lösung (y^*, x^*) von (P5) für die $\langle a, (y^*, x^*) \rangle \leq \beta$ gilt.

Beweis.

Sei $a \in \mathbb{R}^{m^2n^2+mn}, \beta \in \mathbb{R}$ und $\langle a, (y, x) \rangle \leq \beta$ zulässiger Cut für das Problem (P4).

Angenommen es existiert keine Lösung $(\tilde{y}^*, \tilde{x}^*)$ von (P5) mit $\langle a, (\tilde{y}^*, \tilde{x}^*) \rangle \leq \beta$. Dann gilt für alle Lösungen (y^*, x^*) von (P5): $\langle a, (y^*, x^*) \rangle > \beta$. Sei $(\hat{y}^*, \hat{x}^*) \in \mathcal{L}_{P4}$ (Existenz wg Lemma 3.16), dann gilt nach Lemma 3.28, dass $(\hat{y}^*, \hat{x}^*) \in \mathcal{L}_{P5}$ und damit $\langle a, (\hat{y}^*, \hat{x}^*) \rangle > \beta \Rightarrow$ Widerspruch zu $\langle a, (y, x) \rangle \leq \beta$ zulässiger Cut für (P4).

□

Aufgrund dieser Aussage kann man also all jene Cuts, welche für (P4) zulässig sind, auch zu Problem (P5) hinzufügen, ohne dass alle Lösungen abgeschnitten werden (siehe Abb. 19).

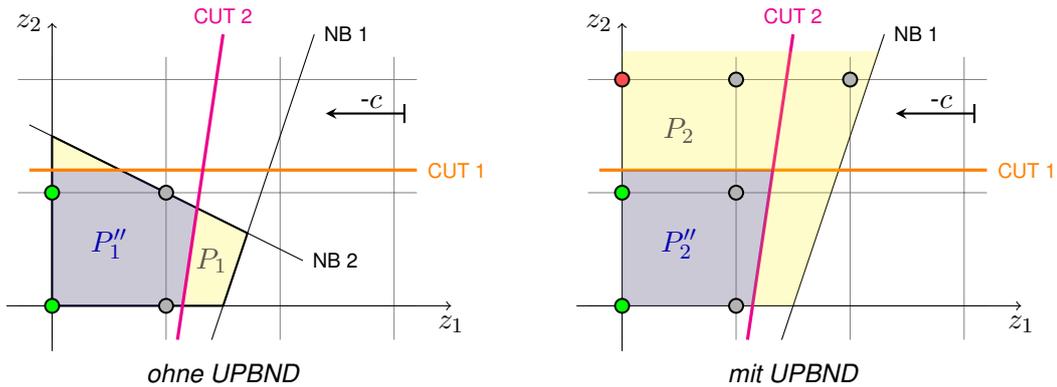


Abbildung 19: Schematische Darstellung der Kompatibilität der Ansätze CUTS (magenta, orange) und UPBND anhand der Auswirkungen von CUTS auf (P4) (links) und (P5) (rechts). Die ursprünglichen Lösungen (grün) bleiben bei Verwendung von CUTS und/oder UPBND optimale Punkte. Die Punkte, welche durch UPBND zu Lösungen wurden (rot), werden durch CUTS potentiell abgeschnitten.

Ob eine Kombination der Ansätze UPBND und CUTS Verwendung findet und wenn ja, welche Cuts zugunsten der Lösbarkeit tatsächlich in unsere Problemformulierung aufgenommen werden, entscheidet sich in mehreren Evaluationsphasen am Ende des Abschnitts 3.2.3. Die Kombination von UPBND mit den Ansätzen SYMM und OBJFCT ist problemlos möglich.

Relaxierung der Ganzzahligkeitsbedingungen - NOBIN

Während die Rekonstruktion der Tastenbelegung in den Problemen (P3) und (P4) sowohl aus den y - wie auch aus den x -Variablen vollzogen werden konnte, war Letzteres in (P5) nicht mehr möglich. Durch eine Relaxierung der oberen Schranken kam es zu dem Verlust der anschaulichen Bedeutung der x_{ijkl} (Belegung eines Tastenpaars (j, l)), weshalb in Zukunft alle optimalen Tastenbelegungen aus den Lösungswerten der y_{ij} gewonnen werden. Dies führt auf die Frage, ob es in (P4) möglich ist, andere Bedingungen als die oberen Schranken (siehe Bedingungen (3.8)) zu relaxieren. Nach dem Loslösen von der anschaulichen Bedeutung der x -Variablen bieten sich besonders die Binärbedingungen der x_{ijkl} für eine Relaxierung an (siehe Abb. 20). Die zugrundeliegende Motivation ist die Hoffnung auf eine enorme Verbesserung der Laufzeit aufgrund der Reduktion an ganzzahligen Variablen und der damit einhergehenden verringerten Größe des *Branch&Bound*-Baums.

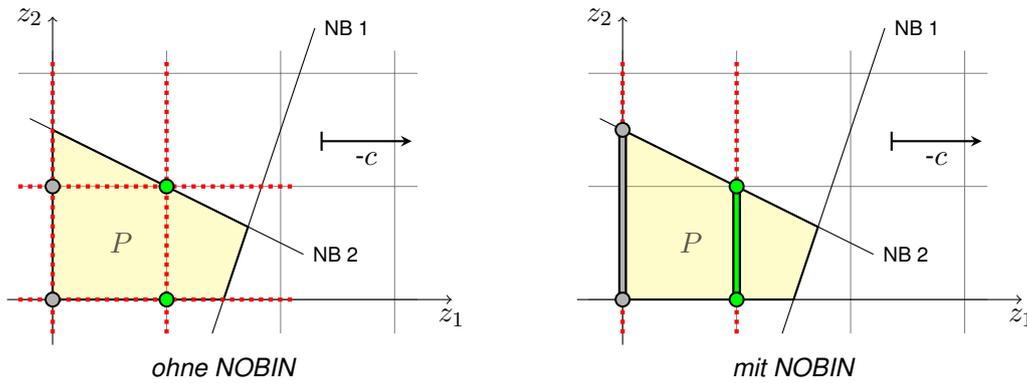


Abbildung 20: Schematische Darstellung der grundlegenden Idee des Ansatzes NOBIN. Die Relaxation der Binärbedingungen (rot) der z_2 -Variable vergrößert den Zulässigkeitsbereich (grau und grün), beeinflusst aber nicht den Wert der z_1 -Komponente von Lösungen (grün).

Das aus (P4) und NOBIN resultierende gemischt-ganzzahlige Problem würde lauten:

$$\begin{aligned}
 & \min (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} \\
 \text{u.d.N. } & \sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \\
 & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \\
 & y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall i, k \in [n], j, l \in [m] \\
 & 2 \cdot x_{ijkl} \leq y_{ij} + y_{kl} \quad \forall i, k \in [n], j, l \in [m] \\
 & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \\
 & 0 \leq x_{ijkl} \leq 1 \quad \forall i, k \in [n], j, l \in [m]
 \end{aligned} \tag{P6}$$

Die Anzahl der Variablen und gewöhnlichen Nebenbedingungen hat sich im Vergleich zu (P4) nicht verändert, jedoch liefert (P6) im Gegensatz dazu aufgrund der Relaxierung nicht mehr notwendigerweise ganzzahlige Lösungen.

Korollar 3.39. Es gilt:

1. $\mathcal{F}_{P4} \subseteq \mathcal{F}_{P6}$ und $\mathcal{P}_{P4} = \mathcal{P}_{P6}$.
2. $\mathcal{F}_{P6} \neq \emptyset$.
3. Für $m = n = 1$ gilt: $\mathcal{F}_{P6} = \mathcal{P}_{P6} = \{(1, 1)\}$.
4. Für $m = n \geq 2$ gilt: $\mathcal{F}_{P4} \subset \mathcal{F}_{P6} \subset \mathcal{P}_{P4} = \mathcal{P}_{P6}$.

Beweis.

Zu 1.: Die Behauptung $\mathcal{F}_{P4} \subseteq \mathcal{F}_{P6}$ folgt aus der Tatsache, dass Problem (P6) eine Relaxierung

3 Distanzminimierung für vertikale Anwendungen

von (P4) ist. $\mathcal{P}_{P4} = \mathcal{P}_{P6}$ folgt aus der Identität der zugehörigen LP-Relaxierungen.

Zu 2.: Die Behauptung folgt aus Aussage 1 dieses Korollars und Korollar 3.12.

Zu 3.: Die Aussage erhält man durch das Aufstellen der Nebenbedingungen von (P6) oder seiner LP-Relaxierung (für $m = n = 1$).

Zu 4.: Sei $m = n \geq 2$. Mit Punkt 3 aus Korollar 3.12 und der 1. Aussage dieses Lemmas bleibt zu zeigen, dass (i) $\exists(\hat{y}, \hat{x}) \in \mathcal{F}_{P6}$ mit $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P4}$ und (ii) $\exists(\tilde{y}, \tilde{x}) \in \mathcal{P}_{P6}$ mit $(\tilde{y}, \tilde{x}) \notin \mathcal{F}_{P6}$.

Betrachte (\hat{y}, \hat{x}) mit $\hat{y}_{ij} = 1 \forall i = j$ und $\hat{y}_{ij} = 0 \forall i \neq j$ und $\hat{x}_{ijkl} = 0, 5 \cdot (\hat{y}_{ij} + \hat{y}_{kl}) \forall i, j, k, l$. Dann gilt $\hat{y}_{ij} + \hat{y}_{kl} - 1 = 0, 5 \cdot (\hat{y}_{ij} + \hat{y}_{kl}) + \underbrace{0, 5 \cdot (\hat{y}_{ij} + \hat{y}_{kl}) - 1}_{\leq 0} \leq 0, 5 \cdot (\hat{y}_{ij} + \hat{y}_{kl}) = \hat{x}_{ijkl}$

und $\sum_{i=1}^n \hat{y}_{ij} = 1 \forall j$ und $\sum_{j=1}^m \hat{y}_{ij} = 1 \forall i$, weshalb gilt: $(\hat{y}, \hat{x}) \in \mathcal{F}_{P6}$. Aber für $\tilde{i}, \tilde{k} \in [n], \tilde{l} \in [m]$ mit $\tilde{k} \neq \tilde{l}$ (Existenz wg. $m = n \geq 2$) gilt $\hat{x}_{\tilde{i}\tilde{k}\tilde{l}} = 0, 5 \cdot (\hat{y}_{\tilde{i}\tilde{k}} + \hat{y}_{\tilde{l}\tilde{l}}) = 0, 5 \notin \{0, 1\}$. Daraus folgt $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P4}$ und damit Behauptung (i).

Betrachte (\tilde{y}, \tilde{x}) mit $\tilde{y}_{ij} = \frac{1}{m} \forall i, j$ und $\tilde{x}_{ijkl} = \frac{1}{m} \forall i, j, k, l$. Dann gilt $2 \cdot \tilde{x}_{ijkl} = \frac{2}{m} = \tilde{y}_{ij} + \tilde{y}_{kl}$ und $\tilde{y}_{ij} + \tilde{y}_{kl} - 1 = \frac{2}{m} - 1 \leq \frac{1}{m} = \tilde{x}_{ijkl}$ und $\sum_{i=1}^n \tilde{y}_{ij} = 1 \forall j$ und $\sum_{j=1}^m \tilde{y}_{ij} = 1 \forall i$, weshalb folgt, dass $(\tilde{y}, \tilde{x}) \in \mathcal{P}_{P6}$. Aber wegen $\tilde{y}_{ij} \notin \{0, 1\}$ gilt $(\tilde{y}, \tilde{x}) \notin \mathcal{F}_{P6}$ und damit Behauptung (ii). □

Lemma 3.40. Es gilt: $(y^*, x^*) \in \mathcal{L}_{P4} \Rightarrow (y^*, x^*) \in \mathcal{L}_{P6}$.

Beweis.

Sei $(y^*, x^*) \in \mathcal{L}_{P4}$, dann gilt $\begin{cases} (i) & (y^*, x^*) \in \mathcal{F}_{P4} \\ (ii) & f_{P4}(y^*, x^*) \leq f_{P4}(y, x) \forall (y, x) \in \mathcal{F}_{P4} \end{cases}$

Mit (i) und Aussage 1 aus Korollar 3.39 folgt, dass $(y^*, x^*) \in \mathcal{F}_{P6}$.

Angenommen $(y^*, x^*) \notin \mathcal{L}_{P6}$, dann $\exists(\hat{y}, \hat{x}) \in \mathcal{F}_{P6} : f_{P6}(\hat{y}, \hat{x}) < f_{P6}(y^*, x^*)$.

Wegen $f_{P4} = f_{P6}$ gilt dann aber, dass $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P4}$, sonst wäre (y^*, x^*) keine Lösung von (P4). Daraus folgt, dass $\tilde{i}, \tilde{k} \in [n]$ und $\tilde{j}, \tilde{l} \in [m]$ existieren mit $\hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \notin \{0, 1\}$. Damit gilt $0 < \hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} < 1$ (wg. $(\hat{y}, \hat{x}) \in \mathcal{F}_{P6}$). Wähle (\tilde{y}, \tilde{x}) mit $\tilde{y}_{ij} = \hat{y}_{ij} \forall i, j$ und $\tilde{x}_{ijkl} = \hat{x}_{ijkl} \forall (i, j, k, l) \neq (\tilde{i}, \tilde{j}, \tilde{k}, \tilde{l})$ und $\tilde{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0$, dann gilt $2 \cdot \tilde{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0 \leq \tilde{y}_{\tilde{i}\tilde{j}} + \tilde{y}_{\tilde{k}\tilde{l}}$ und $\tilde{y}_{\tilde{i}\tilde{j}} + \tilde{y}_{\tilde{k}\tilde{l}} - 1 = \hat{y}_{\tilde{i}\tilde{j}} + \hat{y}_{\tilde{k}\tilde{l}} - 1 \leq \hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} < 1$. Daraus folgt $\tilde{y}_{\tilde{i}\tilde{j}} + \tilde{y}_{\tilde{k}\tilde{l}} - 1 \leq 0 = \tilde{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}$. Insgesamt gilt damit $(\tilde{y}, \tilde{x}) \in \mathcal{F}_{P4}$ und $f_{P6}(\tilde{y}, \tilde{x}) \leq f_{P6}(\hat{y}, \hat{x}) < f_{P6}(y^*, x^*)$. Mit $f_{P4} = f_{P6}$ folgt der Widerspruch zu (ii). □

Korollar 3.41. Es gilt: $\mathcal{L}_{P6} \neq \emptyset$.

Beweis.

Die Aussage folgt aus $\mathcal{L}_{P4} \neq \emptyset$ (3.16) und Lemma 3.40.

□

Folgende Lemmata liefern die Berechtigung der Verwendung von NOBIN.

Lemma 3.42. Sei (y^*, x^*) Lösung von (P6), dann gilt $x_{ijkl}^* \geq y_{ij}^* \cdot y_{kl}^* \forall i, k \in [n], j, l \in [m]$.

Beweis.

Der Beweis funktioniert analog zum Beweis von Lemma 3.32.

□

Lemma 3.43. Sei (y^*, x^*) Lösung von (P6). Für $\tilde{i}, \tilde{k} \in [n], \tilde{j}, \tilde{l} \in [m]$ mit $d_{\tilde{j}\tilde{l}} \cdot p_{\tilde{i}\tilde{k}} > 0$ gilt:
 $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* = y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^*$.

Beweis.

Der Beweis funktioniert analog zum Beweis von Lemma 3.33.

□

Analog zu Lemma 3.31, welches sich dem Zielfunktionswert von (P5) widmet, folgt aus Lemma 3.43, dass eine Lösung von (P4) und eine Lösung von (P6) den gleichen Zielfunktionswert besitzen. Außerdem gilt für (P6) folgendes:

Lemma 3.44. Sei $a \in \mathbb{R}^{m^2n^2+mn}, \beta \in \mathbb{R}$ und $\langle a, (y, x) \rangle \leq \beta$ zulässiger Cut für das Problem (P4), dann gilt: Es existiert eine Lösung (y^*, x^*) von (P6) für die $\langle a, (y^*, x^*) \rangle \leq \beta$ gilt.

Beweis.

Der Beweis funktioniert analog zum Beweis von Lemma 3.38.

□

Damit eignet sich (P6) für die Suche nach optimalen Tastenbelegungen, welche aus den Lösungswerten der y -Variablen rekonstruiert werden können. Um im späteren Verlauf der Arbeit die zuvor präsentierten Ansätze zur Verbesserung der Lösbarkeit von (P4) mit NOBIN kombinieren zu können, wird im Folgenden die Kompatibilität der Ansätze untersucht. Interessant ist an dieser Stelle insbesondere der Ansatz UPBND, da bei der Kombination der beiden Ansätze zwei Relaxierungen an Problem (P4) vorgenommen werden. Zunächst jedoch werden die Folgen einer Verwendung von NOBIN mit CUTS betrachtet. Lemma 3.44 gestattet es, alle zuvor entdeckten, ursprünglich für (P4) gedachten Schnittebenen (siehe Bedingungen (3.10),(3.11),(3.12),(3.13) und (3.14)) auch auf das Problem (P6) anzuwenden. Damit sind die Ansätze CUTS und NOBIN kompatibel, genauso wie die Ansätze SYMM und OBJFCT zusammen mit NOBIN verwendet werden können, da ihnen Ideen zugrundeliegen, die auf den y -Variablen basieren und deshalb von einer Relaxierung der x -Variablen unberührt bleiben.

Bevor Aussagen zur Kompatibilität von UPBND und NOBIN gemacht werden, betrachten wir das zugehörige Problem:

$$\begin{aligned}
 & \min (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} \\
 \text{u.d.N. } & \sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \\
 & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \\
 & y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall i, k \in [n], j, l \in [m] \\
 & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \\
 & 0 \leq x_{ijkl} \leq 1 \quad \forall i, k \in [n], j, l \in [m]
 \end{aligned} \tag{P7}$$

Bis auf die fehlende Ganzzahligkeitsbedingung an die x_{ijkl} ist die Formulierung identisch zu (P5). Neben einer möglicherweise nicht-ganzzahlige Lösung gelten folgende Eigenschaften für Problem (P7).

Korollar 3.45. *Es gilt:*

1. $\mathcal{F}_{P4} \subseteq \mathcal{F}_{P7}$ und $\mathcal{P}_{P4} \subseteq \mathcal{P}_{P7}$.
2. $\mathcal{F}_{P5} \subseteq \mathcal{F}_{P7} \subseteq \mathcal{P}_{P5} = \mathcal{P}_{P7}$.
3. $\mathcal{F}_{P6} \subseteq \mathcal{F}_{P7}$ und $\mathcal{P}_{P6} \subseteq \mathcal{P}_{P7}$.
4. $\mathcal{F}_{P7} \neq \emptyset$.
5. $\mathcal{F}_{P7} = \mathcal{P}_{P7} = \{(1, 1)\}$.
6. Für $m = n \geq 2$ gilt: $\mathcal{F}_{P4} \subset \mathcal{F}_{P7}$ und $\mathcal{P}_{P4} \subset \mathcal{P}_{P7}$.
7. Für $m = n \geq 2$ gilt: $\mathcal{F}_{P5} \subset \mathcal{F}_{P7} \subset \mathcal{P}_{P5} = \mathcal{P}_{P7}$.
8. Für $m = n \geq 2$ gilt: $\mathcal{F}_{P6} \subset \mathcal{F}_{P7}$ und $\mathcal{P}_{P6} \subset \mathcal{P}_{P7}$.

Beweis.

Zu 1. und 3.: Die Behauptungen folgen daraus, dass Problem (P7) eine Relaxierung von (P4) und (P6) darstellt.

Zu 2.: Der Zusammenhang $\mathcal{P}_{P5} = \mathcal{P}_{P7}$ folgt aus der Tatsache, dass die LP-Relaxierungen der Probleme (P5) und (P7) identisch sind. Mit dem Wissen, dass allgemein $\mathcal{F} \subseteq \mathcal{P}$ gilt, bleibt zu zeigen, dass $\mathcal{F}_{P5} \subseteq \mathcal{F}_{P7}$, was daraus folgt, dass (P7) eine Relaxierung von (P5) darstellt.

Zu 4.: Die Behauptung folgt aus Aussage 1 dieses Korollars und Lemma 3.12.

Zu 5.: Die Aussage erhält man durch das Aufstellen der Nebenbedingungen von (P7) oder seiner LP-Relaxation (für $m = n = 1$).

Zu 6.: Die Behauptung folgt aus $\mathcal{F}_{P4} \subset \mathcal{F}_{P6}$ und $\mathcal{P}_{P4} = \mathcal{P}_{P6}$ (Korollar 3.39) und Aussage 8 dieses Korollars.

Zu 7.: Sei $m = n \geq 2$. Dank der 2. Aussage dieses Korollars bleibt zu zeigen, dass:

(i) $\exists(\hat{y}, \hat{x}) \in \mathcal{F}_{P7}$ mit $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P5}$ und (ii) $\exists(\tilde{y}, \tilde{x}) \in \mathcal{P}_{P7}$ mit $(\tilde{y}, \tilde{x}) \notin \mathcal{F}_{P7}$.

Betrachte (\hat{y}, \hat{x}) mit $\hat{y}_{ij} = 1 \forall i = j$ und $\hat{y}_{ij} = 0 \forall i \neq j$ und $\hat{x}_{ijkl} = 1 \forall i = j, k = l$ und $\hat{x}_{ijkl} = 0,5 \forall i \neq j \vee k \neq l$. Dann gilt $(\hat{y}, \hat{x}) \in \mathcal{F}_{P7}$, da $\sum_{i=1}^n \hat{y}_{ij} = 1 \forall j$ und $\sum_{j=1}^m \hat{y}_{ij} = 1 \forall i$ und $\hat{y}_{ij} + \hat{y}_{kl} - 1 \leq \hat{x}_{ijkl} \forall i, j, k, l$ erfüllt ist. Aber $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P5}$, da $\hat{x}_{ijkl} = 0,5 \notin \{0, 1\}$ existieren (wg. $m = n \geq 2$). Damit folgt Behauptung (i).

Betrachte außerdem (\tilde{y}, \tilde{x}) mit $\tilde{y}_{ij} = \frac{1}{m} \forall i, j$ und $\tilde{x}_{ijkl} = 0 \forall i, j, k, l$. Dann gilt $\sum_{i=1}^n \tilde{y}_{ij} = 1 \forall j$ und $\sum_{j=1}^m \tilde{y}_{ij} = 1 \forall i$ und $\tilde{y}_{ij} + \tilde{y}_{kl} - 1 = \frac{2}{m} - 1 \leq 0 = \tilde{x}_{ijkl}$, weshalb gilt $(\tilde{y}, \tilde{x}) \in \mathcal{P}_{P7}$. Aber da $\tilde{y}_{ij} \notin \{0, 1\}$, folgt $(\tilde{y}, \tilde{x}) \notin \mathcal{F}_{P7}$ und damit die Behauptung (ii).

Zu 8.: Sei $m = n \geq 2$, dann bleibt dank Aussage 3 dieses Korollars zu zeigen, dass:

(i) $\exists(\hat{y}, \hat{x}) \in \mathcal{F}_{P7}$ mit $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P6}$ und (ii) $\exists(\hat{y}, \hat{x}) \in \mathcal{P}_{P7}$ mit $(\hat{y}, \hat{x}) \notin \mathcal{P}_{P6}$.

Betrachte (\hat{y}, \hat{x}) mit $\hat{y}_{ij} = 1 \forall i = j$ und $\hat{y}_{ij} = 0 \forall i \neq j$ und $\hat{x}_{ijkl} = 1 \forall i, j, k, l$. Dann gilt $\sum_{i=1}^n \hat{y}_{ij} = 1 \forall j$ und $\sum_{j=1}^m \hat{y}_{ij} = 1 \forall i$ und $\hat{y}_{ij} + \hat{y}_{kl} - 1 \leq \hat{x}_{ijkl} \forall i, j, k, l$, weshalb folgt $(\hat{y}, \hat{x}) \in \mathcal{F}_{P7}$.

Wähle $\tilde{i}, \tilde{k} \in [n]$, $\tilde{l} \in [m]$ mit $\tilde{k} \neq \tilde{l}$ (Existenz wg. $m = n \geq 2$), dann gilt $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P6}$, da $2 \cdot \hat{x}_{\tilde{i}\tilde{k}\tilde{l}} = 2 > 1 + 0 = \hat{y}_{\tilde{i}\tilde{k}} + \hat{y}_{\tilde{k}\tilde{l}}$ (*) gilt. Daraus folgt die Behauptung (i).

Außerdem gilt: $(\hat{y}, \hat{x}) \in \mathcal{F}_{P7} \subseteq \mathcal{P}_{P7}$ und zusammen mit (*), dass $(\hat{y}, \hat{x}) \notin \mathcal{P}_{P6}$, weshalb die Behauptung (ii) gilt.

□

Lemma 3.46. Es gilt: $(y^*, x^*) \in \mathcal{L}_{P4} \Rightarrow (y^*, x^*) \in \mathcal{L}_{P7}$.

Beweis.

Sei $(y^*, x^*) \in \mathcal{L}_{P4}$, dann gilt $\begin{cases} (i) & (y^*, x^*) \in \mathcal{F}_{P4} \\ (ii) & f_{P4}(y^*, x^*) \leq f_{P4}(y, x) \forall (y, x) \in \mathcal{F}_{P4} \end{cases}$

Mit (i) und Aussage 1. aus Korollar 3.45 folgt, dass $(y^*, x^*) \in \mathcal{F}_{P7}$.

Angenommen $(y^*, x^*) \notin \mathcal{L}_{P7}$, dann $\exists(\hat{y}, \hat{x}) \in \mathcal{F}_{P7} : f_{P7}(\hat{y}, \hat{x}) < f_{P7}(y^*, x^*)$.

Wegen $f_{P4} = f_{P7}$ gilt dann aber, dass $(\hat{y}, \hat{x}) \notin \mathcal{F}_{P4}$, sonst wäre (y^*, x^*) keine Lösung von (P4). Damit existieren $\tilde{i}, \tilde{k} \in [n], \tilde{j}, \tilde{l} \in [m] : \hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} \notin \{0, 1\} \vee 2 \cdot \hat{x}_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} > \hat{y}_{\tilde{i}\tilde{j}} + \hat{y}_{\tilde{k}\tilde{l}}$. Beide Fälle lassen sich analog zu den Beweisen der Lemmata 3.28 und 3.40 zum Widerspruch führen. □

Korollar 3.47. *Es gilt: $\mathcal{L}_{P7} \neq \emptyset$.*

Beweis.

Die Behauptung folgt aus $\mathcal{L}_{P4} \neq \emptyset$ (Korollar 3.16) und Lemma 3.46. □

Lemma 3.48. *Sei $(y^*, x^*) \in \mathcal{L}_{P7}$, dann gilt $x_{ijkl}^* \geq y_{ij}^* \cdot y_{kl}^* \forall i, k \in [n], j, l \in [m]$.*

Beweis.

Der Beweis funktioniert analog zum Beweis von Lemma 3.32. □

Lemma 3.49. *Sei $(y^*, x^*) \in \mathcal{L}_{P7}$. Für $\tilde{i}, \tilde{k} \in [n], \tilde{j}, \tilde{l} \in [m]$ mit $d_{\tilde{j}\tilde{l}} \cdot p_{\tilde{i}\tilde{k}} > 0$ gilt: $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}}^* = y_{\tilde{i}\tilde{j}}^* \cdot y_{\tilde{k}\tilde{l}}^*$.*

Beweis.

Der Beweis funktioniert analog zum Beweis von Lemma 3.33. □

Wie schon zuvor (siehe Lemma 3.31) folgt mit Lemma 3.49 die Gleichheit der Zielfunktionswerte von (P4) und (P7).

Lemma 3.50. *Sei $a \in \mathbb{R}^{m^2n^2+mn}, \beta \in \mathbb{R}$ und $\langle a, (y, x) \rangle \leq \beta$ zulässiger Cut für das Problem (P4), dann gilt: Es existiert eine Lösung (y^*, x^*) von (P7) für die $\langle a, (y^*, x^*) \rangle \leq \beta$ gilt.*

Beweis.

Der Beweis funktioniert analog zum Beweis von Lemma 3.38. □

Damit sind die Ansätze UPBND und NOBIN kompatibel und das daraus resultierende Optimierungsproblem (P7), welches eine optimale Tastenbelegung liefert, darf wie schon zuvor (P6) mit den Ideen CUTS, SYMM und OBJFCT kombiniert werden.

Verwendung einer alternativen Linearisierung - AJLIN

Studiert man die außerordentlich umfangreiche Literatur zum Thema quadratische Zuordnungsprobleme beziehungsweise deren Linearisierungen, stößt man unweigerlich auf die Adams-Johnson-Linearisierung (siehe bspw. [Burkard et al., 1998]). Diese wurde 1994 von Warren Adams und Terri Johnson in [Adams u. Johnson, 1994] vorgestellt. Im Vergleich zu anderen

Linearisierungen ist sie, aufgrund ihrer Nebenbedingungen, hinsichtlich der Lösbarkeit zu favorisieren, obwohl es Linearisierungen gibt, die weniger Variablen oder Nebenbedingungen verwenden. Ein Vorteil, den die Adams-Johnson-Nebenbedingungen mit sich bringen ist, dass sich aus ihnen durch Linearkombinationen die Nebenbedingungen aller anderen gängigen Linearisierungen erzeugen lassen (siehe [Adams u. Johnson, 1994]). Eine schematische Darstellung der Grundidee dieses Ansatzes findet sich in Abbildung 21.

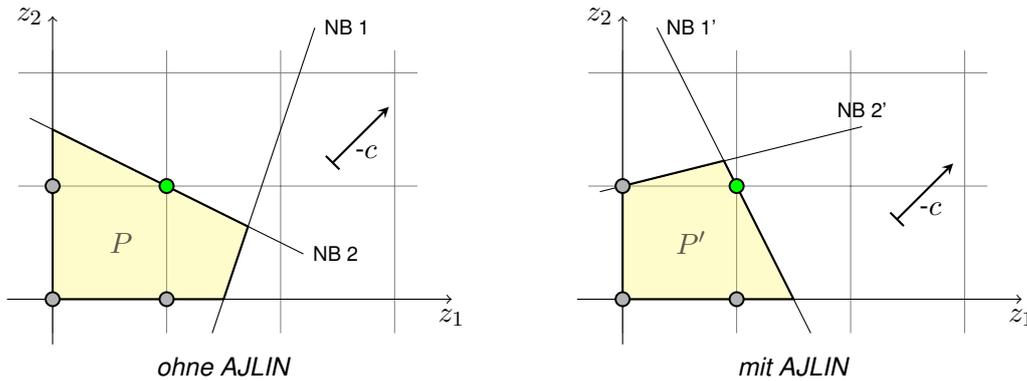


Abbildung 21: Schematische Darstellung der grundlegenden Idee des Ansatzes AJLIN. Dabei sind NB 1' und NB 2' die Nebenbedingungen der neuen Linearisierung, P' das zugehörige Polytop und grüne Punkte Lösungen des Problems.

Angepasst auf unser Problem der Tastaturoptimierung ergibt sich aus der Adams-Johnson-Linearisierung (AJLIN) folgendes gemischt-ganzzahlige, lineare Optimierungsproblem:

$$\begin{aligned}
 & \min (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} \\
 \text{u.d.N.} & \sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \\
 & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \\
 & \sum_{i=1}^n x_{ijkl} = y_{kl} \quad \forall k \in [n], j, l \in [m] \\
 & \sum_{j=1}^m x_{ijkl} = y_{kl} \quad \forall i, k \in [n], l \in [m] \\
 & x_{ijkl} = x_{klij} \quad \forall i, k \in [n], j, l \in [m] \\
 & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \\
 & x_{ijkl} \geq 0 \quad \forall i, k \in [n], j, l \in [m]
 \end{aligned} \tag{P8}$$

Da die Adams-Johnson-Linearisierung genau wie die Linearisierung aus (P4) darauf basiert, dass die x -Variablen das Produkt der zugehörigen y -Variablen repräsentieren, ist die Anzahl

der Variablen in beiden Fällen gleich $m^2n^2 + mn$. Jedoch enthält (P8) $m^2n + n^2m + m + n$ Nebenbedingungen (ohne Ganzzahligkeits- und Positivitätsbedingungen) und damit deutlich weniger als Problem (P4).

Die beiden ersten Gruppen an Nebenbedingungen sind bereits aus zuvor betrachteten Problemen (bspw. (P4)) bekannt, genau wie die paarweise Gleichheit der x -Variablen (siehe Gleichung (3.12)). Die bisher unbekanntenen Nebenbedingungen der dritten Gruppe fordern, dass für jede feste Wahl eines Buchstaben $\tilde{k} \in [n]$ und zweier Tasten $\tilde{j}, \tilde{l} \in [m]$, der Wert der Summe über die $x_{i\tilde{j}\tilde{k}\tilde{l}}$ -Variablen gleich dem der $y_{\tilde{k}\tilde{l}}$ -Variablen ist. Anschaulich lässt sich dieser Zusammenhang so erklären, als dass sicher ein Buchstabe existiert, welcher auf der gewählten Taste \tilde{j} liegt, weshalb die Werte der einzelnen Summanden davon abhängen, wie \tilde{k} und \tilde{l} gewählt wurden. Liegt der Buchstabe \tilde{k} nicht auf Taste \tilde{l} , so sind alle Summanden gleich Null. Ist die Variable $y_{\tilde{k}\tilde{l}}$ gleich Eins, dann ergibt sich zusammen mit den anderen Nebenbedingungen, dass genau einer der Summanden gleich Eins ist (siehe auch Korollar 3.52). Die Nebenbedingungen der vierten Gruppe haben eine ähnliche Struktur wie die eben erläuterten und fordern dieselbe Eigenschaft der x -Variablen für eine gewählte Buchstabe-Buchstabe-Taste-Kombination.

Lemma 3.51. Sei $(y, x) \in \mathcal{F}_{P8}$, dann gilt: $x_{ijkl} = y_{ij} \cdot y_{kl} \forall i, k \in [n], j, l \in [m]$.

Beweis.

Sei $(y, x) \in \mathcal{F}_{P8}$. Für $m = n = 1$ folgt $y_{11} = 1$ und $x_{1111} = 1$ direkt aus den Nebenbedingungen und die Behauptung ist erfüllt. Für $m = n \geq 2$ wähle $\tilde{i}, \tilde{k} \in [n], \tilde{j}, \tilde{l} \in [m]$ und unterscheide:

Fall 1: $y_{\tilde{i}\tilde{j}} = 0$.

Daraus folgt, dass $y_{\tilde{i}\tilde{j}} \cdot y_{\tilde{k}\tilde{l}} = 0$. Außerdem gilt $\sum_{k=1}^n x_{k\tilde{l}\tilde{i}\tilde{j}} = y_{\tilde{i}\tilde{j}} = 0$. Mit $x_{ijkl} \geq 0 \forall i, j, k, l$ folgt $x_{k\tilde{l}\tilde{i}\tilde{j}} = 0 \forall k$ und damit $x_{\tilde{k}\tilde{l}\tilde{i}\tilde{j}} = 0$. Mit $x_{ijkl} = x_{klij} \forall i, j, k, l$ folgt daraus, dass $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0$.

Fall 2: $y_{\tilde{k}\tilde{l}} = 0$.

Daraus folgt, dass $y_{\tilde{i}\tilde{j}} \cdot y_{\tilde{k}\tilde{l}} = 0$. Außerdem gilt $\sum_{i=1}^n x_{i\tilde{j}\tilde{k}\tilde{l}} = y_{\tilde{k}\tilde{l}} = 0$, weshalb $x_{i\tilde{j}\tilde{k}\tilde{l}} = 0 \forall i$ und damit $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0$.

Fall 3: $y_{\tilde{i}\tilde{j}} = y_{\tilde{k}\tilde{l}} = 1$.

Dann gilt $y_{\tilde{k}\tilde{l}} = 1 \wedge y_{\tilde{i}\tilde{j}} = 1$ und $\sum_{i=1}^n x_{i\tilde{j}\tilde{k}\tilde{l}} = x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} + \sum_{[n] \ni i \neq \tilde{i}} x_{i\tilde{j}\tilde{k}\tilde{l}} = y_{\tilde{k}\tilde{l}} = 1$ (*). Wegen $\sum_{i=1}^n y_{i\tilde{j}} = 1$

und $y_{\tilde{i}\tilde{j}} = 1$ gilt, dass $y_{i\tilde{j}} = 0 \forall i \neq \tilde{i}$.

Wähle $\hat{i} \neq \tilde{i}$ (Existenz wg. $m = n \geq 2$), dann gilt $\sum_k x_{k\tilde{l}\tilde{i}\tilde{j}} = y_{\tilde{i}\tilde{j}} = 0 \forall \hat{i} \neq \tilde{i}$ und damit $x_{k\tilde{l}\tilde{i}\tilde{j}} = 0 \forall k, \hat{i} \neq \tilde{i}$. Mit $x_{ijkl} = x_{klij} \forall i, j, k, l$ folgt daraus, dass $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0 \forall k, \hat{i} \neq \tilde{i}$. Damit gilt

$x_{i\tilde{j}k\tilde{l}} = 0 \forall \tilde{i} \neq \tilde{j}$, weshalb $\sum_{i \neq \tilde{i}} x_{i\tilde{j}k\tilde{l}} = 0$. Mit (\star) folgt $x_{i\tilde{j}k\tilde{l}} = 1$.

□

Korollar 3.52. Sei $(y, x) \in \mathcal{F}_{P8}$, dann gilt: $x_{ijkl} \in \{0, 1\} \forall i, k \in [n], j, l \in [m]$.

Beweis.

Sei $(y, x) \in \mathcal{F}_{P8}$, dann gilt $y_{ij} \in \{0, 1\}$. Mit Lemma 3.51 folgt die Behauptung.

□

Analog zu (P4) kann eine optimale Tastenbelegung dank Lemma 3.51 sowohl aus den x - als auch aus den y -Werten der Lösung ausgelesen werden. Darin unterscheidet sich die Adams-Johnson-Linearisierung des QAP von den Problemen (P5),(P6) und (P7), für deren Lösungen die Rekonstruktion der Belegung nur aus den y -Variablen möglich war.

Einen weiteren Unterschied zwischen den gemischt-ganzzahligen Problemen (P6),(P7) und (P8) liefert Korollar 3.52. Es besagt, dass zulässige Punkte und damit insbesondere Lösungen von (P8) im Gegensatz zu (P6) und (P7) immer ganzzahlig sind.

Zusammenfassung und Evaluation der potentiellen Verbesserungen

Tabelle 2 bietet einen Überblick über die in Abschnitt 3.2.3 erarbeiteten Ansätze.

Um die Frage zu beantworten, welche der potentiellen Verbesserungen tatsächlich in die finale Problemformulierung aufgenommen werden, wird ihr Einfluss auf die Optimierung am Beispiel der Testinstanz 2 mit $m = n = 8$ untersucht. Die zugehörigen konkreten Daten sind in Anhang A.2.5 zu finden.

Um das Verbesserungspotential der Ansätze besser abschätzen zu können, werden in der ersten Evaluationsphase zunächst die eingebauten Strategien des Solvers ausgeschaltet. Die aus der Verwendung der einzelnen Ansätze resultierenden Laufzeiten findet man in Tabelle 3. Auf eine Spalte mit der Anzahl der Variablen wird bewusst verzichtet, da sie unabhängig von den verwendeten Ansätzen ist (64 y -Variablen, 4096 x -Variablen). Dies gilt auch für den Zielfunktionswert, der abgesehen von Testfall (g), in welchem die Zielfunktion verändert wird, gleich 116,742 LE ist und die Distanz angibt, die für die Eingabe aller Bigramme zurückgelegt werden muss. Alle Testfälle (Spalte 1) basieren auf Problem (P4), wobei die zweite Spalte über die daran vorgenommenen Änderungen Aufschluss gibt. Die dritte und vierte Spalte listen die Anzahl der Gleichungs- und Ungleichungsnebenbedingungen auf, Spalte 5 liefert die Laufzeit des zugehörigen Testfalls.

Analysiert man die Ergebnisse aus Tabelle 3, so stellt man fest, dass bei unterdrückten Solverstrategien die größte Verbesserung der Laufzeit durch die Verwendung von NOBIN erzielt werden kann. Der Grund dafür ist die Reduktion der Größe des *Branch&Bound*-Baums, die aus der Vernachlässigung der Ganzzahligkeitsbedingungen an die x -Variablen resultiert.

Tabelle 2: Schematischer Überblick über die Ansätze zur Verbesserung der zeitlichen Lösbarkeit von Problem (P4) und welche Nebenbedingungen hinzugefügt (\oplus) oder entfernt (\ominus) werden.

Name des Ansatzes		Unterschiede zu Problem (P4)
CUTS(1)	\oplus	$\sum_{jl} x_{ijkl} = 1$ $\sum_{ik} x_{ijkl} = 1$
CUTS(2)	\oplus	$x_{ijkl} = x_{klij}$
CUTS(3)	\oplus	$x_{ijij} = y_{ij}$
CUTS(4)	\oplus	$\sum_{ij} x_{ijkl} = n \cdot y_{kl}$
SYMM(\tilde{i}, T)	\oplus	$\sum_{j \in T} y_{ij} = 1$
OBJFCT	Zfkt.	$f_{P4} + 2 \cdot \sum_{ij} d_j \cdot p_i \cdot y_{ij}$
UPBND	\ominus	$2 \cdot x_{ijkl} \leq y_{ij} + y_{kl}$
AJLIN		AJCON, CUTS(2), NOBIN
AJCON	\oplus	$\sum_i x_{ijkl} = y_{kl}$ $\sum_j x_{ijkl} = y_{kl}$
NOBIN	\ominus	$x_{ijkl} \in \{0, 1\}$
	\oplus	$0 \leq x_{ijkl} \leq 1$

Ein weiterer Ansatz, der eine enorme Verbesserung mit sich bringt, ist UPBND, der ebenfalls auf einer Relaxierung des Problems basiert. Im Gegensatz zu NOBIN behält er die Forderung nach Ganzzahligkeit bei und vernachlässigt stattdessen die oberen Schranke der x_{ijkl} . Der Grund für die Beschleunigung der Optimierung liegt hier in der Reduktion der zu erfüllenden Ungleichungen.

Auch die Testfälle (f1),(f2),(f3),(g) liefern im Vergleich zum ursprünglichen Modell schneller optimale Lösungen des Problems. Dies lässt darauf schließen, dass die Vermeidung von Symmetrien ein vielversprechender Ansatz zur Verbesserung der Lösbarkeit ist. Die Differenz der Laufzeiten in den Fällen (f1), (f2), (f3) resultiert aus den Unterschieden der Wahrscheinlichkeiten der fixierten Buchstaben. Eine anschauliche Erklärung dafür ist, dass mit steigender Wahrscheinlichkeit für einen Buchstaben auch die Wahrscheinlichkeit für Bigramme, an denen er beteiligt ist, steigt. Somit ist sein Einfluss auf die Zielfunktion größer als der von seltenen Buchstaben. Es macht also Sinn, den häufigen Buchstaben eine gewisse Priorität einzuräumen und sie zu-

Tabelle 3: Phase 1 - Evaluation der erarbeiteten Ansätze mit unterdrückten Solverstrategien.

Testfall	verw. Ansatz	Ungl.NB	Gl.NB	LZ
(a)	-	8192	16	21,8 s
(b)	CUTS(1)	8192	144	46,7 s
(c)	CUTS(2)	8192	2256	>10 min
(d)	CUTS(3)	8192	80	23,1 s
(e)	CUTS(4)	8192	80	43,4 s
(f1)	SYMM(e, H_R)	8192	17	4,3 s
(f2)	SYMM(d, H_R)	8192	17	5,1 s
(f3)	SYMM(f, H_R)	8192	17	7,5 s
(g)	OBJFCT	8192	16	8,1 s
(h)	UPBND	4096	16	3,0 s
(i)	AJCON	8192	1040	29,5 s
(j)	AJLIN	0	3280	163,5 s
(k)	NOBIN	8192	16	2,3 s

erst zu verteilen. Geschieht diese Verteilung zusätzlich auf einem vorbestimmten Teilbereich, dann erleichtert das die Aufgabe. Fixiert man im Gegensatz dazu einen seltenen Buchstaben auf einem Teilbereich, so schränkt man die Platzierung der priorisierten Buchstaben ein, was die Bestimmung einer optimalen Tastenbelegung anspruchsvoller macht.

Darüberhinaus stellt man fest, dass sich die verwendeten Ansätze aus den Fällen (b), (c), (d), (e), (i), (j) im derzeitigen Setting nicht dafür eignen, die Lösbarkeit des Optimierungsproblems zu verbessern. Der Hauptgrund ist die Vervielfachung der Nebenbedingungen, die im Idealfall eine Hilfestellung für den Solver sein sollten, jedoch bei unterdrückten Strategien zu überflüssigen Forderungen werden. Aus diesem Grund erscheint es sinnvoll in einer zweiten Evaluationsphase die Unterdrückung der Solverstrategien aufzuheben.

Auch in der Realität würde man die implementierten Strategien des Solvers in der Regel eingeschaltet lassen, da sie erfahrungsgemäß kürzere Laufzeiten erzielen. Um die Ansätze aus Abschnitt 3.2.3 bestmöglichst bewerten zu können und den Einfluss der Solver-Strategien zu ermitteln, wird die Dauer des Lösungsvorgangs für zwei Szenarien betrachtet (siehe Tabelle 4). Im ersten Szenario (Spalte 5) wird die eingebaute Presolving-Strategie des Solvers verwendet, während Cut-Strategien und heuristische Verfahren ausgeschaltet sind. Im zweiten Szenario (Spalte 6) misst man die Laufzeit unter Verwendung all dieser Strategien. Die Spalten 3 und 4 geben die Anzahl der nach dem Presolving verbleibenden Nebenbedingungen und Variablen an.

Bei der Analyse der Ergebnisse aus Szenario 1 der Phase 2 (eingeschaltetes Presolving - Tabelle 4) stellt man zuallererst fest, dass die ursprüngliche Anzahl der Nebenbedingungen und der Variablen im Vergleich zu Phase 1 (Tabelle 3) in allen Fällen durch das Presolving reduziert werden konnte. Damit geht eine deutliche Reduktion der Laufzeit einher. Der Vergleich der

Tabelle 4: Phase 2 - Evaluation der erarbeiteten Ansätze mit ausgewählten Solverstrategien.

Testfall	verw. Ansatz	NB	Var.	LZ(P)	LZ(P,C,H)
(a)	-	3152	1632	1,8 s	1,8 s
(b)	CUTS(1)	6400	3200	38,6 s	88,6 s
(c)	CUTS(2)	3152	1632	2,0s	2,1 s
(d)	CUTS(3)	3152	1632	1,7 s	1,7 s
(e)	CUTS(4)	6352	3200	17,4 s	148,5 s
(f1)	SYMM(e, H_R)	2760	1432	1,5 s	1,6 s
(f2)	SYMM(d, H_R)	2760	1432	1,6 s	1,7 s
(f3)	SYMM(f, H_R)	2760	1432	2,0 s	2,0 s
(g)	OBJFCT	3152	1632	2,1 s	2,2 s
(h)	UPBND	1668	1716	1,0 s	1,8 s
(i)	AJCON	7632	3200	3,9 s	28,1 s
(j)	AJLIN	912	1632	1,1 s	1,9 s
(k)	NOBIN	5728	2920	3,3 s	4,2 s

Laufzeiten untereinander liefert, dass die Fälle (b) und (e) deutlich mehr Zeit als die anderen Ansätze benötigen, um eine optimale Lösung zu bestimmen. Der Grund dafür ist, dass das Presolving in diesen Fällen weniger erfolgreich ist. Auch (i) und (k) zählen zu den langsameren Fällen, in denen viele Nebenbedingungen und Variablen nach dem Presolving verbleiben. Jedoch profitiert der Ansatz AJCON von der Qualität seiner Nebenbedingungen (siehe Abschnitt AJLIN) und NOBIN von der kleineren Menge ganzzahliger Variablen, sodass sie trotzdem unterhalb der Laufzeiten von (b) und (e) bleiben. Wie schon in Evaluationsphase 1 (siehe Tabelle 3) liefert der Ansatz UPBND in (h) aufgrund seiner relaxierenden Eigenschaft sehr schnell eine Lösung von (P4). Die größte Verbesserung (-99%) durch das Verwenden der eingebauten Presolving-Strategie ergibt sich für Fall (j), womit der Ansatz AJLIN hinsichtlich der Laufzeit zu Fall (h) aufschließt. Ein Blick auf die Fälle (a), (c) und (d) offenbart, dass diese hinsichtlich der Anzahl an Nebenbedingungen und Variablen übereinstimmen, was darauf hinweist, dass sie in gleicher Art und Weise von der Presolving-Strategie profitieren und die Ansätze aus (c) und (d) keine neuen Informationen für das linearisierte Optimierungsproblem (P4) (siehe Fall (a)) liefern. Die Laufzeiten sind dementsprechend sehr ähnlich und unterscheiden sich nur durch die Dauer des Presolvings. Im Gegensatz zur Testreihe mit unterdrückten Strategien (Tabelle 3) kann in Fall (g) kein Laufzeitgewinn gegenüber Fall (a) verzeichnet werden. Die minimale zusätzliche Dauer resultiert vermutlich aus dem Presolving, das letztendlich jedoch ein Problem gleicher Größe liefert. Die Laufzeit der Fälle (f1), (f2), (f3) wächst analog zur vorherigen Testreihe (Tabelle 3) mit sinkender Wahrscheinlichkeit der fixierten Buchstaben. Davon hängt auch ab, ob eine Verbesserung der Laufzeit im Vergleich zu (a) erzielt werden kann.

Beim Vergleich der Laufzeiten des ersten und zweiten Szenarios (Spalte 5 und 6 - Tabelle 4) beobachtet man, dass trotz der Verwendung aller Solver-Strategien im zweiten Szenario die

Laufzeiten meist größer sind als bei der Unterdrückung der Cutting- und Heuristik-Strategien. Erwartet hätte man an dieser Stelle die konträre Entwicklung. Die Gründe hierfür könnten sein, dass die Generierung von Cuts durch den Solver, sowie die Anwendung von heuristischen Verfahren zusätzliche Zeit benötigt, aber der erwünschte Erfolg hinsichtlich der Laufzeit ausbleibt. Vergleicht man die Laufzeiten der einzelnen Fälle in Szenario 2, so erkennt man Tendenzen bezüglich einer Verbesserung oder Verschlechterung der Lösbarkeit, die identisch sind zu denen aus Szenario 1 (siehe Tabelle 3). Gründe für die verschiedenen Verhältnisse in manchen Fällen dürften in der unterschiedlich erfolgreichen Vorgehensweise der Strategien des Solvers liegen.

Insgesamt kann man festhalten, dass für die ursprüngliche Linearisierung (siehe (P4)) insbesondere UPBND einen positiven Einfluss auf die Laufzeit hat. Neben UPBND ist vor allem AJLIN in der Lage, die Suche nach einer optimalen Tastenbelegung zu beschleunigen. Aus diesem Grund werden die Formulierungen (P5) und (P8) die Grundlage für weitere Verbesserungen sein. Außerdem erscheint es sinnvoll, über eine Kombination der Ansätze OBJFCT und AJLIN nachzudenken. Zu diesem Zweck wird AJLIN in seine Komponenten AJCON, CUTS(2) und NOBIN aufgeteilt (siehe Tabelle 2), sodass diese auf (P5) angewendet werden können.

Die Ansätze CUTS(1), CUTS(3) und CUTS(4) verschlechtern die Laufzeit oder liefern keine neuen Informationen, da sie beispielsweise durch Aufsummieren aus den schärferen Nebenbedingungen von AJCON erzeugt werden können. Aufgrund dessen und der fehlenden Laufzeitgewinne werden diese Ansätze nicht weiter verwendet.

Aufgrund der exakteren Modellierung der zurückzulegenden Gesamtstrecke wird im weiteren Verlauf OBJFCT zur Symmetrievermeidung eingesetzt. Jedoch muss noch geklärt werden, ob eine Kombination mit $\text{SYMM}(\tilde{i}, H_R)$ erfolgt, um die Zahl der symmetrischen Lösungen weiter zu reduzieren. Die Erfahrungen aus den ersten beiden Evaluationsphasen haben gezeigt, dass sich häufige Buchstaben besonders als Fixkandidaten eignen, weshalb bei einem Einsatz von SYMM der Buchstaben e fixiert wird, da er die größte Wahrscheinlichkeit hat (siehe Tabelle 24 in Anhang A.1).

Somit verbleiben die Ansätze OBJFCT, $\text{SYMM}(e, H_R)$, UPBND, CUTS(2), AJCON, NOBIN und AJLIN in der Menge der potentiellen Verbesserungen. In einer dritten Phase werden die Auswirkungen von Kombinationen dieser Ansätze auf die Laufzeit untersucht.

Tabelle 5 listet die zugehörigen Resultate auf, wobei sie analog zu Tabelle 4 aufgebaut ist.

Vergleicht man zunächst die Anzahl der Variablen und Nebenbedingungen, die nach dem Presolving verbleiben, so stellt man fest, dass diese durch die Verwendung von $\text{SYMM}(e, H_R)$ in allen Fällen weiter reduziert werden konnten. Vor allem in den Fällen (r1), (s1) und (t1) wirkt sich dies positiv auf die Dauer der Optimierung aus.

Die Verwendung von Heuristiken und Cuttingstrategien bringt in (l1)-(t1) eine Zunahme der Laufzeit mit sich. Damit verhalten sich diese Fälle analog zur Phase 2 (siehe Tabelle 4). Für die Fälle (l2)-(t2) ist eine solche Entwicklung nicht feststellbar, was daran liegt, dass bei Verwen-

Tabelle 5: Phase 3 - Evaluation von Kombinationen ausgewählter Ansätze.

Test	verwendete Ansätze	NB	Var.	LZ(P)	LZ(P,C,H)
(l1)	OBJFCT, UPBND	1668	1716	2,1 s	3,3 s
(m1)	OBJFCT, UPBND, NOBIN	1472	1520	1,7 s	2,3 s
(n1)	OBJFCT, UPBND, AJCON	4048	3200	2,4 s	13,4 s
(o1)	OBJFCT, UPBND, AJCON, NOBIN	4048	3200	0,7 s	1,1 s
(p1)	OBJFCT, UPBND, CUTS(2)	1668	1716	2,1 s	5,1 s
(q1)	OBJFCT, UPBND, CUTS(2), NOBIN	1472	1520	1,7 s	6,6 s
(r1)	OBJFCT, UPBND, AJCON, CUTS(2)	2480	1632	2,0 s	3,3 s
(s1)	OBJFCT, UPBND, AJCON, CUTS(2), NOBIN	2480	1632	1,3 s	2,7 s
(t1)	OBJFCT, AJLIN	912	1632	1,0 s	3,0 s
(l2)	OBJFCT, SYMM(e, H_R), UPBND	1450	1494	10 s	1,6 s
(m2)	OBJFCT, SYMM(e, H_R), UPBND, NOBIN	1276	1320	3,0 s	1,9 s
(n2)	OBJFCT, SYMM(e, H_R), UPBND, AJCON	3600	2999	2,3 s	3,0 s
(o2)	OBJFCT, SYMM(e, H_R), UPBND, AJCON, NOBIN	3600	2999	1,0 s	3,2 s
(p2)	OBJFCT, SYMM(e, H_R), UPBND, CUTS(2)	1450	1494	3,6 s	3,0 s
(q2)	OBJFCT, SYMM(e, H_R), UPBND, CUTS(2), NOBIN	1276	1320	3,4 s	1,5 s
(r2)	OBJFCT, SYMM(e, H_R), UPBND, AJCON, CUTS(2)	2228	1432	1,6 s	0,8 s
(s2)	OBJFCT, SYMM(e, H_R), UPBND, AJCON, CUTS(2), NOBIN	2228	1432	0,8 s	0,6 s
(t2)	OBJFCT, SYMM(e, H_R), AJLIN	856	1432	0,5 s	0,5s

dung aller Solverstrategien alternative Lösungsverfahren angewendet werden und diese gut mit dem Ansatz SYMM(e, H_R) harmonieren. Aus Gründen der Nachvollziehbarkeit der resultierenden Laufzeiten, werden im weiteren Verlauf nur Presolving-Strategien verwendet.

Eine Untersuchung der Laufzeiten zeigt außerdem, dass NOBIN positive Effekte auf die Lösungsdauer hat. Diese Effekte verstärken sich in Kombinationen mit den Ansätzen AJCON und CUTS(2) (siehe (o1), (o2), (q1), (q2), (s1), (s2)), was vor allem an der kleineren Anzahl an Variablen (im Vergleich zu (l1) bzw. (l2)) liegt.

Insgesamt ergibt die Auswertung der Testergebnisse, dass die Optimierungssoftware (siehe Abschnitt 2.4.6) mit eingeschaltetem Presolving für die Berechnung einer optimalen Tastenbelegung der Testinstanz 2 am wenigsten Zeit benötigt, wenn der Ansatz OBJFCT zusammen mit der Adams-Johnson-Linearisierung und SYMM(e, H_R) (siehe Fall (t2)) verwendet wird.

Das aus (P4) und den Verbesserungen aus Abschnitt 3.2.3 resultierende Optimierungsproblem hat folgende allgemeine Form:

$$\begin{aligned}
 & \min (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m d_j \cdot p_i \cdot y_{ij} \\
 \text{u.d.N.} & \sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \\
 & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n], i \neq \tilde{i} \\
 & \sum_{j \in T} y_{ij} = 1 \\
 & \sum_{i=1}^n x_{ijkl} = y_{kl} \quad \forall k \in [n], j, l \in [m] \\
 & \sum_{j=1}^m x_{ijkl} = y_{kl} \quad \forall i, k \in [n], l \in [m] \\
 & x_{ijkl} = x_{klij} \quad \forall i, k \in [n], j, l \in [m] \\
 & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \\
 & x_{ijkl} \geq 0 \quad \forall i, k \in [n], j, l \in [m]
 \end{aligned} \tag{P9}$$

Die Anzahl der Variablen und Nebenbedingungen ist identisch zu Problem (P8).

Der Versuch 26 Buchstaben optimal auf dem *QWERTY-like*-Layout mit 26 Tasten zu verteilen, führt unter Verwendung der obigen Formulierung auf ein Problem mit 457.652 Variablen und 271.805 Nebenbedingungen. Trotz der Reduktion auf 219.668 Variablen und 34.580 Nebenbedingungen durch das Presolving scheitert die Suche nach einer optimalen Tastenbelegung und wird nach einer Rechenzeit von 20 Minuten abgebrochen.

Es muss also festgestellt werden, dass trotz der Bemühungen in Abschnitt 3.2.3 keiner der Ansätze den Durchbruch liefert und das Problem für realistische Größenordnungen aus zeitlicher Sicht nach wie vor unlösbar ist.

3.2.4 Auswege aus der zeitlichen Unlösbarkeit

Implizites Nullsetzen geeigneter x -Variablen - IMPLZERO

Da trotz der Verbesserungen aus Abschnitt 3.2.3 das Problem einer optimalen Tastenbelegung für 26 Buchstaben nicht gelöst werden konnte, wird in diesem Abschnitt ein neuer Ansatz verfolgt. Sein Ziel ist es die Problemgröße deutlich zu reduzieren, indem x_{ijkl} -Variablen, welche keinen Einfluss auf die Zielfunktion haben, per Gleichungsnebenbedingung auf Null gesetzt werden.

Betrachtet man dafür die Zielfunktion des Optimierungsproblems (P9), so stellt man fest, dass dies der Fall ist, wenn $p_{ik} = 0$ oder $d_{jl} = 0$. Anschaulich bedeutet das, dass eine x -Variable auf Null gesetzt werden kann, wenn das zugehörige Bigramm niemals auftritt, oder die dafür

zurückzulegende Distanz gleich Null ist. Letzteres würde bedeuten, dass das Bigramm auf einer Taste liegt.

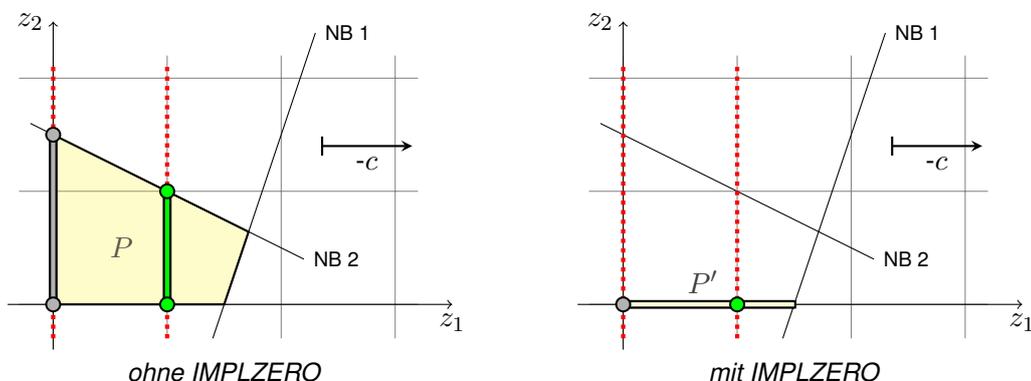


Abbildung 22: Schematische Darstellung der grundlegenden Idee des Ansatzes IMPLZERO. Die nicht-ganzzahlige Variable z_2 wird auf 0 gesetzt, da sie keinen Einfluss auf den Zielfunktionswert hat. Damit verändert sich der Zulässigkeitsbereich (grau und grün), die Menge der Lösungen (grün) und das urspr. Polytop P wird zu P' (Binärbedingungen sind rot).

Zusätzlich zu diesen beiden Fällen, lassen sich dank der festgelegten Eigenschaften der Tastenbelegungen (ein Buchstabe pro Taste, eine Taste pro Buchstabe) bereits vor der Optimierung x -Variablen identifizieren, von denen man weiß, dass sie in jedem zulässigen Punkt und damit in jeder Lösung den Wert 0 annehmen. Dies sind jene x -Variablen, die den gewünschten Tastatureigenschaften widersprechen (siehe Abschnitte 2.3 und 3.2.1). Insgesamt lässt sich damit die Indexmenge I^0 erzeugen. Mit $I := \{(i, k, j, l) \mid i, k \in [n], j, l \in [m]\}$ gilt:

$$I^0 := \{(i, j, k, l) \in I \mid (i = k \wedge j \neq l) \vee (i \neq k \wedge j = l) \vee d_{jl} = 0 \vee p_{ik} = 0\} \quad (3.17)$$

Sie enthält alle (i, j, k, l) -Kombinationen, für die ein Buchstabe auf zwei verschiedenen Tasten liegt, eine Taste zwei verschiedene Buchstaben beherbergt, die Entfernung zweier Tasten gleich Null ist und Kombinationen zweier verschiedener Buchstaben, die niemals im Textkorpus auftreten. Offensichtlich gilt $d_{jl} = 0 \Leftrightarrow j = l$, da zwei verschiedene Tasten immer einen echt positiven Abstand haben. Damit lässt sich I^0 vereinfachen zu:

$$I^0 = \{(i, j, k, l) \in I \mid i = k \vee j = l \vee p_{ik} = 0\} \quad (3.18)$$

Die zu den Indize aus I^0 gehörigen x -Variablen sind jene, für die gefordert wird, dass sie gleich Null sind. So lauten die $|I^0|$ Bedingungen, welche die ausgewählten Variablen fixieren:

$$x_{ijkl} = 0 \quad \forall (i, j, k, l) \in I^0 \quad (3.19)$$

Die Anzahl der binären Variablen, welche nicht durch obige Gleichungsbedingung fixiert wurden, ist demnach $m^2n^2 + mn - |I^0|$. Es gilt:

$$m^2n + mn^2 - mn \leq |I^0| \leq m^2n^2 \quad (3.20)$$

Innerhalb dieser Schranken hängt die Größenordnung von $|I^0|$ stark von den Werten der Wahrscheinlichkeiten p_{ik} ab.

Ein Punkt, den man bei der Verwendung dieses Ansatzes beachten muss, ist, dass aufgrund des Nullsetzens mancher x_{ijkl} das Auslesen der Tastenbelegung aus den x -Variablen, analog zu (P5),(P6) und (P7), aber im Gegensatz zu (P8), nicht mehr möglich ist. Die Tastenbelegung wird also aus den y -Variablen rekonstruiert.

Möchte man die Nebenbedingungen (3.19) direkt zu Optimierungsproblem (P9) hinzufügen, so birgt dies ein großes Risiko. Durch das Nullsetzen von Variablen kann es nämlich vorkommen, dass manche der Nebenbedingungen aus (P9) ihre Gültigkeit verlieren. Um nicht auf ein Optimierungsproblem mit leerem Zulässigkeitsbereich zu stoßen, ist es deshalb zwingend erforderlich, dass alle Nebenbedingungen daraufhin überprüft werden, ob sie mit dem Ansatz kompatibel sind.

Die Nebenbedingungen, welche keine x -Variablen enthalten ($\sum_{i=1}^n y_{ij} = 1$ und $\sum_{j=1}^m y_{ij} = 1$, sowie $\sum_{j \in T} y_{ij}$) sind von IMPLZERO nicht betroffen, weshalb sich an ihrer Korrektheit nichts ändert. Betrachtet man die Gleichungen $x_{ijkl} = x_{klij}$ so stellt man fest, dass deren Richtigkeit nicht mehr gewährleistet ist, da es in manchen Fällen dazu kommen kann, dass die Variable einer Seite (in ihrer Funktion als Repräsentant der Belegung eines Tastenpaars) gleich Eins sein müsste, die andere durch (3.19) jedoch auf Null gesetzt wurde. Als Konsequenz betrachtet man nur noch die Nebenbedingungen, in denen keine der beiden x -Variablen gleich Null gesetzt wurde. Somit gilt:

$$x_{ijkl} = x_{klij} \quad \forall (i, j, k, l) \in I \setminus I^0 \quad (3.21)$$

Die Anzahl der Nebenbedingungen hängt dabei von $|I^0|$ und damit von p_{ik} ab.

Die Richtigkeit der verbleibenden Gleichungsnebenbedingungen der Adams-Johnson-Linearisierung ($\sum_i x_{ijkl} = y_{kl}$ und $\sum_j x_{ijkl} = y_{kl}$), bleibt dann erhalten, wenn sie zu Ungleichungen aufgeweicht werden. Definiere $I_{jkl} := \{(j, k, l) \mid (i, j, k, l) \in I\}$ und I_{ikl} analog, dann lauten die angepassten Nebenbedingungen:

$$\sum_{i=1}^n x_{ijkl} \leq y_{kl} \quad \forall (j, k, l) \in I_{jkl} \quad (3.22)$$

und

$$\sum_{j=1}^m x_{ijkl} \leq y_{kl} \quad \forall (i, k, l) \in I_{ikl} \quad (3.23)$$

Nimmt man in (P9) lediglich die Anpassungen der Nebenbedingungen an IMPLZERO vor (siehe

(3.21),(3.22) und (3.23)), so ergibt sich für die daraus resultierende Formulierung ein weiteres Problem. Da nun außer der Positivitätsbedingung keine unteren Schranken mehr an die x -Variablen existieren, wäre das Ergebnis der Optimierung eine Tastenbelegung, mit erwarteter zurückzulegender Distanz 0, da alle $x_{ijkl} = 0$ gewählt werden würden. Diesem Problem kann man entkommen, indem man untere Schranken für all jene Variablen einführt, die nicht gleich Null gesetzt wurden. Mit den unteren Schranken aus Abschnitt 3.2.2 (Nebenbedingungen (3.7)) erhält man somit folgende zusätzliche Nebenbedingungen:

$$y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall (i, j, k, l) \in I \setminus I^0 \quad (3.24)$$

Dies führt auf das Optimierungsproblem:

$$\begin{aligned} \min & (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m d_j \cdot p_i \cdot y_{ij} \\ \text{u.d.N.} & \sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \\ & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n], i \neq \tilde{i} \\ & \sum_{j \in T} y_{\tilde{i}j} = 1 \\ & y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall (i, j, k, l) \in I \setminus I^0 \\ & \sum_{i=1}^n x_{ijkl} \leq y_{kl} \quad \forall (j, k, l) \in I_{jkl} \\ & \sum_{j=1}^m x_{ijkl} \leq y_{kl} \quad \forall (i, k, l) \in I_{ikl} \\ & x_{ijkl} = x_{klij} \quad \forall (i, j, k, l) \in I \setminus I^0 \\ & x_{ijkl} = 0 \quad \forall (i, j, k, l) \in I^0 \\ & x_{ijkl} \geq 0 \quad \forall (i, k, j, l) \in I \setminus I^0 \\ & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \end{aligned} \quad (\text{P10})$$

Das Problem (P10) besteht aus $2 \cdot |I \setminus I^0| + m^2n + n^2m + m + n$ Nebenbedingungen (ohne die Ganzzahligkeitsbedingungen an y_{ij} und die Positivitäts- beziehungsweise Fixierungsbedingungen an x_{ijkl}) und aus $m^2n^2 + mn - |I^0|$ nicht-fixierten Variablen, wovon für mn viele Ganzzahligkeit gefordert wird.

Neben der Möglichkeit, von (P9) ausgehend die Nebenbedingungen anzupassen und untere Schranken einzuführen, existiert ein weiteres Vorgehen, um das Problem (P10) zu erhalten. So könnte man den Ansatz des impliziten Nullsetzens alternativ auf (P7) anwenden, also zu (P7) die Gleichungen $x_{ijkl} = 0 \quad \forall i, j, k, l \in I^0$ hinzufügen. Wie zuvor wäre es nötig, die ursprüng-

lichen Nebenbedingungen auf den Ansatz anzupassen. Einige Nebenbedingungen der Form $y_{ij} + y_{kl} - 1 \leq x_{ijkl}$ wären nicht mehr richtig (beispielsweise für $y_{\tilde{i}\tilde{j}} = y_{\tilde{k}\tilde{l}} = 1$ und $x_{\tilde{i}\tilde{j}\tilde{k}\tilde{l}} = 0$, aufgrund von $p_{\tilde{i}\tilde{k}} = 0$ und $(\tilde{i}, \tilde{j}) \neq (\tilde{k}, \tilde{l})$), sodass man nur folgende Bedingungen beibehalten würde:

$$y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall (i, j, k, l) \in I \setminus I^0 \quad (3.25)$$

Das soeben erzeugte Optimierungsproblem (aus (P7) + IMPLZERO) beschreibt zwar die Suche nach einer optimalen Tastatur auf korrekte Art und Weise. Es würde also im Gegensatz zu Problem (P9) + IMPLZERO theoretisch immer auch eine sinnvolle Lösung liefern, jedoch nicht in befriedigender Zeit. Der Grund dafür ist die schlechte Qualität der unteren Schranke im *Branch&Bound*-Verfahren. Erinnern wir uns zurück an das Ende des Abschnitts 3.2.2, als wir schon einmal an einem ähnlichen Punkt standen und uns zusätzlicher Cuts bedient haben, um die untere Schranke und damit die Laufzeit zu verbessern. Dieser Idee kann man erneut folgen, muss dabei jedoch vorsichtig sein, damit durch das Hinzufügen von Schnittebenen keine Probleme mit leerem Zulässigkeitsbereich entstehen. Um dies zu vermeiden, wird jede der neuen Bedingungen, welche Teil des Optimierungsproblems werden soll, geprüft und gegebenenfalls angepasst. Aufgrund der Erfolge der Adams-Johnson-Linearisierung in der Evaluationsphase (siehe Tabelle 5) entscheiden wir uns für die dazugehörigen Nebenbedingungen (AJCON, CUTS(2)) als neue Schnittebenen. Die Anpassung auf den Ansatz IMPLZERO liefert wie zuvor die Bedingungen (3.21), (3.22) und (3.23). Integriert man diese Nebenbedingungen in (P7) + IMPLZERO, erhält man wie zuvor das Problem (P10).

Betrachtet man die Menge der aufgeweichten Adams-Johnson-Constraints in (P10) stellt sich die Frage, ob man zu weit gegangen ist und sich einige der aufgeweichten Bedingungen wieder etwas verschärfen lassen. Ist es also möglich, Index-Kombinationen zu identifizieren, für die die ursprünglichen Gleichheitsbedingungen trotz Nullsetzen gelten, also die zugehörigen Bedingungen ihre Richtigkeit durch das Nullsetzen nicht eingebüßt haben. Man stellt fest, dass für manche der (i, k, l) -Kombinationen weiterhin Gleichheit in den Bedingungen der Form $\sum_j x_{ijkl} \leq y_{kl}$ garantiert werden kann, weil in bestimmten Fällen nicht alle Summanden x_{ijkl} implizit gleich Null gesetzt wurden. Sei $I_{ikl}^0 := \{(i, k, l) \mid (i, j, k, l) \in I^0\}$ und definiere analog I_{jkl}^0 . Die ursprünglichen Nebenbedingungen teilen sich dann auf in:

$$\sum_{j=1}^m x_{ijkl} \leq y_{kl} \quad \forall (i, k, l) \in I_{ikl}^0 \quad (3.26)$$

$$\sum_{j=1}^m x_{ijkl} = y_{kl} \quad \forall (i, k, l) \in I_{ikl} \setminus I_{ikl}^0 \quad (3.27)$$

Für die Nebenbedingungen der Form $\sum_i x_{ijkl} \leq y_{kl}$ können wir keine solchen (j, k, l) -Kombinationen

finden. Damit bleibt:

$$\sum_{i=1}^n x_{ijkl} \leq y_{kl} \quad \forall (j, k, l) \in I_{jkl} \quad (3.28)$$

Daraus resultiert folgendes Problem:

$$\begin{aligned} \min & (B-1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m d_j \cdot p_i \cdot y_{ij} \\ \text{u.d.N.} & \sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \\ & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n], i \neq \tilde{i} \\ & \sum_{j \in T} y_{ij} = 1 \\ & y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall (i, j, k, l) \in I \setminus I^0 \\ & \sum_{i=1}^n x_{ijkl} \leq y_{kl} \quad \forall (j, k, l) \in I_{jkl} \\ & \sum_{j=1}^m x_{ijkl} \leq y_{kl} \quad \forall (i, k, l) \in I_{ikl}^0 \\ & \sum_{j=1}^m x_{ijkl} = y_{kl} \quad \forall (i, k, l) \in I_{ikl} \setminus I_{ikl}^0 \\ & x_{ijkl} = x_{klij} \quad \forall (i, j, k, l) \in I \setminus I^0 \\ & x_{ijkl} = 0 \quad \forall (i, j, k, l) \in I^0 \\ & x_{ijkl} \geq 0 \quad \forall (i, k, j, l) \in I \setminus I^0 \\ & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \end{aligned} \quad (P11)$$

Sowohl die Anzahl der Variablen als auch die Anzahl der Nebenbedingungen von Problem (P11) sind identisch zu (P10). Durch die Verschärfung der aufgeweichten Adams-Johnson-Nebenbedingungen (siehe (3.22) und (3.23)) hin zu den Bedingungen (3.26), (3.27) und (3.28) hat sich lediglich das Verhältnis von Ungleichungen und Gleichungen zugunsten Letzterer verschoben.

Verwendet man diese Formulierung, um eine optimale Tastenbelegung für das *QWERTY-like*-Layout ($m = n = 26$) zu berechnen, dann resultiert daraus ein Optimierungsproblem mit 458.354 Nebenbedingungen und 457.652 Variablen. Das Presolving reduziert die Größe auf 194.324 Nebenbedingungen und 166.426 Variablen. Verglichen mit Problem (P9) nach dem Presolving, konnte durch den Ansatz IMPLZERO die Zahl der Variablen um etwa 53.000 verringert werden (ca. -24%), zu dem Preis, dass die Anzahl der Nebenbedingungen um etwa

160.000 gewachsen ist (ca. +460 %). Trotzdem oder vielleicht gerade deshalb schlägt der Versuch eine optimale Belegung von 26 Buchstaben auf 26 Tasten zu finden, fehl.

Reduktion der Einträge der Wahrscheinlichkeitsmatrix

Bei der Analyse des Textkorpus und der Extraktion der Bigrammwahrscheinlichkeiten (siehe Abschnitt 2.4.1) stellt man fest, dass viele der n^2 Bigramme sehr geringe Wahrscheinlichkeiten aufweisen. Diese Beobachtung motiviert die Überlegung, ob einige Einträge der Wahrscheinlichkeitsmatrix aufgrund ihrer geringen Größe vernachlässigt werden können, da sie ohnehin nur einen kleinen Einfluss auf die Zielfunktion haben. An dieser Stelle ist es wichtig, sich klar machen, dass man durch ein solches Vorgehen die Optimalität der Lösung aufgeben würde. Die Rechtfertigung dafür könnte nur eine enorme Verbesserung der Lösbarkeit sein.

Im Folgenden werden verschiedene Möglichkeiten vorgestellt, die alle auf eine Reduktion der Einträge der Wahrscheinlichkeitsmatrix abzielen, um in Kombination mit IMPLZERO die Anzahl der Variablen oder Nebenbedingungen zu verkleinern. Sämtliche präsentierte Methoden finden sich im Programm *VirtualKeyboardOptimization.mos* als wählbare Optionen wieder (siehe Abschnitt 7.3).

Werfen wir zuerst einen Blick auf den bereits angedeuteten Ansatz THRESHOLD (kurz: THR(\tilde{p})), der darauf basiert nicht mehr alle Bigrammwahrscheinlichkeiten zu betrachten, sondern nur noch jene über einer gewissen Schwelle $0 \leq \tilde{p} \leq 1$. Alle Wahrscheinlichkeiten p_{ik} mit $p_{ik} < \tilde{p}$ werden vernachlässigt, indem sie gleich Null gesetzt werden. Dies führt dazu, dass die zugehörigen Variablen keinen Einfluss mehr auf die Zielfunktion haben. Dem Ansatz des impliziten Nullsetzens aus Abschnitt 3.2.4 folgend, lassen sich diese dann per Gleichungsnebenbedingung auf Null fixieren. Die Auswirkungen auf die Laufzeit hängen dabei stark von der Größenordnung von \tilde{p} ab (siehe Tabelle 7). Aus Sicht der Lösbarkeit wären möglichst viele fixierte Variablen und damit möglichst wenige Nebenbedingungen zu favorisieren ($\tilde{p} \gg 0$). Der Wunsch eine Tastatur zu finden, deren Zielfunktionswert möglichst nahe am Optimum ist, forciert aber das konträre Verlangen möglichst wenige Wahrscheinlichkeiten zu unterdrücken (\tilde{p} klein). An dieser Stelle ist es also nötig, einen befriedigenden Kompromiss zwischen Lösbarkeit und Qualität der Lösung zu finden.

Neben THRESHOLD gibt es viele weitere Möglichkeiten, die Anzahl der Einträge der Wahrscheinlichkeitsmatrix (Tabelle 25 in Anhang A.1) zu reduzieren. Denkbar sind Verfahren, wie die Auswahl einer bestimmten Anzahl $N \in \mathbb{N}$ an Einträgen pro Zeile (ENTRIES_PER_ROW, kurz: EPR(N)), die Auswahl der $N \in \mathbb{N}$ besten Einträge (BEST_ENTRIES, kurz: BE(N)) oder Kombinationen der Ansätze untereinander (ENTRIES_PER_ROW + BEST_ENTRIES). Außerdem können die Ansätze mit der Forderung verknüpft werden, dass eine Mindestanzahl N an Einträgen pro Zeile verbleibt (MIN_ENTRIES_PER_ROW, kurz: MIN(N)), oder eine Maximalanzahl nicht überschritten wird (MAX_ENTRIES_PER_ROW, kurz: MAX(N)). Dabei haben

alle Ansätze gemein, dass sie im Rahmen ihrer Anforderungen größere Einträge bevorzugt auswählen.

Eine Konsequenz aller Reduktionsmethoden ist, dass die Summe der neuen Bigrammwahrscheinlichkeiten ungleich Eins ist, weshalb der Zielfunktionswert nicht mehr der tatsächlich zurückzulegenden Gesamtstrecke entspricht. Um den Einfluss des zweiten Terms, welcher den Hin- und Rückweg modelliert, auf den Zielfunktionswert nicht zu verfälschen, skaliert man den ersten Term und erhält:

$$\frac{B-1}{\sum_{i,k=1}^n \tilde{p}_{ik}} \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m \tilde{p}_{ik} \cdot d_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m d_j \cdot p_i \cdot y_{ij} \quad (3.29)$$

Dabei bezeichnet \tilde{p}_{ik} die Einträge der bearbeiteten Bigramm-Wahrscheinlichkeitsmatrix. Der zugehörige Zielfunktionswert gibt die Strecke an, die zurückgelegt werden muss, um einen Text der Länge B schreiben, welcher nur aus den Bigramme besteht, die in die Optimierung einbezogenen wurden. Mit $\tilde{I}^0 := I^0 \cup \{(i, j, k, l) \mid \tilde{p}_{ik} = 0\}$, erhält man das neue Optimierungsproblem:

$$\begin{aligned} \min \quad & \frac{B-1}{\sum_{i,k=1}^n \tilde{p}_{ik}} \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m \tilde{p}_{ik} \cdot d_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m d_j \cdot p_i \cdot y_{ij} \\ \text{u.d.N.} \quad & \sum_{i=1}^n y_{ij} = 1 \quad \forall j \in [m] \\ & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n], i \neq \tilde{i} \\ & \sum_{j \in T} y_{ij} = 1 \\ & x_{ijkl} = 0 \quad \forall i, j, k, l \in \tilde{I}^0 \\ & y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall i, j, k, l \in I \setminus \tilde{I}^0 \\ & \sum_{i=1}^n x_{ijkl} \leq y_{kl} \quad \forall j, k, l \in I_{jkl} \\ & \sum_{j=1}^m x_{ijkl} = y_{kl} \quad \forall i, k, l \in I_{ikl} \setminus \tilde{I}_{ikl}^0 \\ & \sum_{j=1}^m x_{ijkl} \leq y_{kl} \quad \forall i, k, l \in \tilde{I}_{ikl}^0 \\ & x_{ijkl} = x_{klij} \quad \forall i, j, k, l \in I \setminus \tilde{I}^0 \\ & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \\ & x_{ijkl} \geq 0 \quad \forall i, k \in [n], j, l \in [m] \end{aligned} \quad (\text{P12})$$

Obwohl dieses Problem auf den ersten Blick dem Problem (P11) ähnelt, so hat es doch deut-

lich weniger nicht-fixierte Variablen ($m^2n^2 + mn - |\tilde{I}^0|$), sowie weniger Nebenbedingungen ($2 \cdot |I \setminus \tilde{I}^0| + m^2n + n^2m + m + n$), da in der Regel $|\tilde{I}^0| \gg |I^0|$ gilt.

An dieser Stelle gilt es zu entscheiden, welche der zuvor präsentierten Methoden künftig verwendet wird, um die Einträge der Bigramm-Wahrscheinlichkeitsmatrix zu reduzieren. Um diese Frage zu beantworten, muss man neben den speziellen Eigenschaften, die resultierende Laufzeit und einen potentiellen Einfluss auf die Qualität der suboptimalen Lösungen vergleichen (siehe Tabelle 6). Dafür werden in einer Testphase die Parameter so gewählt, dass die Anzahl der in die Optimierung einbezogenen Bigramme gleich ist. Eine Ausnahme stellt die Methode ENTRIES_PER_ROW dar, da in ihrem Fall die Anzahl der verbleibenden Bigramme immer ein Vielfaches der Buchstabenanzahl ist.

Tabelle 6 listet die Ergebnisse dieser Testphase auf. Dabei findet man neben dem Zielfunktionswert (Spalte 6) und der Laufzeit (Spalte 7) auch die Anzahl an nach dem Presolving verbleibenden Nebenbedingungen und Variablen (Spalte 3 und 4). Eine Analyse der bearbeiteten Bigramm-Wahrscheinlichkeitsmatrix liefert die Anzahl der Nicht-Null-Einträge, sowie den minimalen, maximalen und durchschnittlichen Matrixeintrag (Spalte 2).

Tabelle 6: Vergleich der Methoden zur Reduktion der Einträge der Wahrscheinlichkeitsmatrix.

Fall	Methode	Einträge $\neq 0$ (min;max;aver.)	NB	Var.	Zielfkt.wert (LE)	LZ
(u)	EPR(1)	26 (0,02;4;1,00885)	19304	13614	117,372	85,9 s
(v)	THR(0,7)	30 (0,71;4;1,38733)	21138	14352	126,03	43,1 s
(w)	BE(30)	30 (0,71;4;1,38733)	21138	14352	126,03	43,1 s
(x)	BE(30)+MAX(2)	30 (0,42;4;1,25567)	20058	14964	128,063	66,3 s
(y)	THR(1,2)+MIN(1)	30 (0,02;4;1,16333)	20758	15264	128,284	51,4 s
(z)	EPR(1)+BE(4)	30 (0,02;4;1,16333)	20758	15264	128,284	51,4 s

Die Auswertung der Resultate zeigt, dass die Laufzeiten der Methoden THR(0.7) und BE(30) sowie THR(1.2)+MIN(1) und EPR(1)+BE(4) übereinstimmen, was daran liegt, dass sie die gleichen Einträge aus der Bigramm-Wahrscheinlichkeitsmatrix auswählen. Die kürzeste Laufzeit kommt durch die Verwendung von THR(0.7) und BE(30) zustande, da diese Verfahren oftmals viele Werte in einer Matrixzeile auswählen. Die Auswahl der Bigramme bei den anderen Methoden ist gezwungenermaßen auf mehrere Zeilen verstreut, was sich unter anderem in einer geringeren durchschnittlichen Wahrscheinlichkeit der ausgewählten Bigramme bemerkbar macht. Anschaulich führt dies dazu, dass die Positionen einer größeren Menge an Buchstaben aneinander gekoppelt sind, weshalb die Optimierung tendenziell schwieriger wird.

Mit steigender Abhängigkeit der Positionen nimmt auch der Zielfunktionswert zu, da aus Platzgründen nicht mehr jeder Buchstabe auf die Taste gelegt werden kann, die seinen Bigramm-Wahrscheinlichkeiten nach am Besten für eine geringe zurückzulegende Distanz wäre.

Insbesondere die unbeschränkte Auswahl der Einträge, die einen geringeren Zielfunktionswert

nach sich zieht, spricht für die künftige Verwendung von THR oder BE. Darüberhinaus kann durch THR oder BE mit geeigneter Parameterwahl jede beliebige Anzahl von Einträgen an das Optimierungsproblem übergeben werden und nicht wie beispielsweise bei EPR nur Vielfache der Buchstabenanzahl n . Aufgrund der verloren gegangenen Optimalität erscheint es außerdem sinnvoll, sich Gedanken über eine nachgeschaltete Heuristik, wie beispielsweise das *2-opt*-Verfahren, zu machen, um die neue Lösung unter Einbeziehung aller Bigramm-Wahrscheinlichkeiten weiter zu verbessern (siehe Kapitel 10). Hierfür eignen sich bevorzugt Lösungen, die aus Verfahren resultieren, welche weniger Buchstaben aneinanderkoppeln (wie THR oder BE), da die nicht-optimierten Buchstaben untereinander gewinnbringend vertauscht werden können, ohne die optimierte Ordnung der anderen Buchstaben zu beschädigen.

Insgesamt stellt man fest, dass alle Eintrag-Reduktions-Methoden in Kombination mit dem Ansatz IMPLZERO ein mächtiges Instrument sind, da sie die Anzahl der Variablen und der Nebenbedingungen deutlich reduzieren und so die Lösbarkeit des Problems ermöglichen. Die Ergebnisse rechtfertigen den Einsatz von THRESHOLD oder BEST_ENTRIES im weiteren Verlauf der Arbeit. Aufgrund der größeren Anschaulichkeit des Parameters \tilde{p} gegenüber N wird künftig der Ansatz THRESHOLD verwendet. Damit gilt $\tilde{I}^0 := \{(i, j, k, l) \mid (i, j, k, l) \in I^0 \vee \tilde{p}_{ik} < \tilde{p}_j\}$.

Der Erfolg des THRESHOLD-Verfahrens basiert auf einer geeigneten Wahl von \tilde{p} . Wie bereits erwähnt, ist die Wahl immer ein Kompromiss zwischen Laufzeit und Qualität der Lösung. Je kleiner das gewählte \tilde{p} , desto mehr Bigrammwahrscheinlichkeiten fließen in die Berechnung mit ein und die Lage der zugehörigen Buchstaben zueinander wird verbessert. Dies steigert in der Regel die Qualität der resultierenden Tastatur, hat aber zur Folge, dass die Lösungsdauer deutlich zunimmt, da die Anzahl der Variablen, welche nicht fixiert wurden, und die Anzahl der Nebenbedingungen zunimmt.

Tabelle 7 zeigt die Resultate der Testläufe mit unterschiedlicher Wahl von \tilde{p} am Beispiel des *QWERTY-like*-Layouts (siehe Abschnitt 2.4.3). Neben der Anzahl der Nebenbedingungen und Variablen (Spalte 4 und 5), dem Zielfunktionswert (Spalte 6) und der Laufzeit (Spalte 7) von (P12) bei unterschiedlicher Wahl von \tilde{p} liefert sie die Anzahl der einbezogenen und nullgesetzten Einträge (Spalte 2) sowie die Summe der Wahrscheinlichkeiten aller in die Optimierung einbezogenen Bigramme (Spalte 3).

Ein Resultat, das auf den ersten Blick verwundert, ist die Erhöhung des Zielfunktionswertes mit abnehmendem \tilde{p} , da man ja an einer Minimierung der zurückzulegenden Gesamtstrecke interessiert ist. Der Grund dafür ist, dass der Zielfunktionswert nicht die Gesamtstrecke angibt, sondern die Strecke, die zurückzulegen wäre, wenn der Text nur aus den von THR ausgewählten Bigrammen besteht. Er vernachlässigt also die Distanzen, die zwischen den unterdrückten Bigrammen zurückgelegt werden müssen. Damit zieht ein großes \tilde{p} (viele unterdrückte Matrixeinträge) zunächst einen geringeren Zielfunktionswert nach sich, da die wenigen betrachteten Buchstaben nah aneinander positioniert werden können und die übrigen keinen Einfluss haben. In der Auswertung der resultierenden Tastatur, also der Berechnung der zurückzulegen-

Tabelle 7: Eigenschaften und Ergebnisse des Problems (P12) für abnehmendes \tilde{p} am Beispiel des QWERTY-like-Layouts.

\tilde{p} (%)	Einträge ($\neq 0$; 0-gesetzt)	Summe WS. (%)	NB	Var.	Zielfkt.wert (LE)	LZ
4,0	1; 471	4	759	1102	102,724	7,4 s
2,0	4; 468	12,38	3280	2540	102,956	9,6 s
1,5	9; 463	21,55	6538	4752	107,25	10,8 s
1,0	16; 456	30,09	9785	6802	113,024	11,5 s
0,9	20; 452	33,91	13052	9102	117,035	16,5 s
0,8	22; 450	35,61	14174	9664	118,165	24,0 s
0,7	30; 442	41,62	21138	14352	126,03	43,1 s
0,6	38; 434	46,77	25059	17514	131,38	199,0 s
0,55	48; 424	52,41	29319	20902	137,44	678,7 s
0,2	131; 341	79,03	68794	53464	-	-
0,0	472; 0	100	186073	160002	-	-

den Gesamtdistanz, würde es sich jedoch negativ bemerkbar machen, dass nur die Lage von vergleichsweise wenigen Buchstaben zueinander optimiert wurde.

Leider lässt sich keine allgemeine Aussage zum Zielfunktionszuwachs machen, der durch die Verkleinerung von \tilde{p} einhergeht, da dies sehr stark von den verwendeten Bigrammwahrscheinlichkeiten abhängt. Auch über die zurückzulegende Gesamtdistanz, die unter Einbeziehung aller Bigramme berechnet wird, lässt sich nur der folgende Trend prophezeien: Die zurückzulegende Distanz nimmt mit kleiner werdendem \tilde{p} ab, da mit kleinem \tilde{p} die Lage vieler Bigramme zueinander optimiert wird. Die Folgerung, dass Tastenbelegungen mit kleinerer Gesamtstrecke notwendigerweise aus einem kleineren \tilde{p} resultieren, gilt nicht.

Da die Suche nach einer optimalen Tastenbelegung keine sich wiederholende Handlung ist, sondern im Idealfall nur einmal vollzogen wird, legt man den Fokus auf die Qualität der Lösung, das heißt man bevorzugt eine kleine Wahl von \tilde{p} . Um im weiteren Verlauf Ergebnisse präsentieren zu können, wird \tilde{p} aus praktischen Gründen und basierend auf den Erfahrungen aus Tabelle 7 gleich 0,6% gewählt. Anschaulich bedeutet dies, dass nur Bigramme vernachlässigt werden, die im Korpus seltener als jedes 166. Bigramm einmal auftreten.

3.2.5 Ergebnisse der Optimierung

Wir haben uns zu Beginn des Abschnitts 3.2 darauf beschränkt, optimale Tastenbelegungen für den Fall zu bestimmen, dass die Anzahl der zu verteilenden Buchstaben gleich der Anzahl der Tasten ist ($m = n$). Betrachtet man die Tastaturlayouts aus Abschnitt 2.4.3 und erinnert man sich daran, dass die 26 Standardbuchstaben des deutschen Alphabets verteilt werden sollen (siehe Abschnitt 2.3.1), so stellt man fest, dass sich unter diesen Voraussetzungen nur eine optimierte Tastenbelegung für das QWERTY-like-Layout ($m = n = 26$) berechnen lässt.

Die Optimierung für $\tilde{p} = 0,6\%$ mit fixiertem e auf der rechten Tastaturhälfte H_R dauert 181.6 s und liefert folgende Belegung.

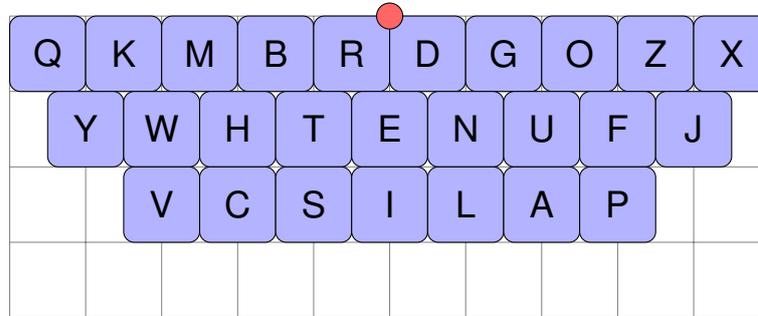


Abbildung 23: Distanz-optimierte Tastenbelegung für das QWERTY-like-Layout bei realistischer Schreibweise im vertikalen Fall.

Für diese Belegung ergibt sich eine zurückzulegende Gesamtdistanz von 220,372 LE (bei einem Zielfunktionswert von 131,38 LE).

Für alle anderen Layouts aus Abschnitt 2.4.3 gilt $m > n$, womit sie nicht die Voraussetzungen des Abschnitts 3.2 erfüllen. Eine Möglichkeit, diese Problematik zu umgehen, wäre es, $m - n$ Dummyzeichen einzuführen, sie zusammen mit den anderen Buchstaben zu verteilen und im Nachhinein zu entfernen. Um den Wert der eigentlichen Lösung nicht zu verfälschen, dürfen die Dummyzeichen $d \in \{n + 1, \dots, m\}$ jedoch nie angesteuert werden. Es müsste also gelten, dass $p_d = p_{id} = p_{di} = 0 \forall d \in \{n + 1, \dots, m\}, i \in [n]$. Ein Nachteil des Hinzufügens von Dummyzeichen ist, dass dieses Vorgehen einer Vergrößerung der Menge an zu verteilenden Zeichen entspricht, was die Dimension des Problems vergrößert und sich negativ auf die Lösungsdauer auswirkt (siehe Tabelle 8). Aus diesem Grund wird die bisher erarbeitete Theorie im folgenden Abschnitt verallgemeinert bevor optimierte Tastenbelegungen für die Layouts 3×9 , 3×10 oder 4×8 präsentiert werden.

3.3 Verallgemeinerung des verbesserten, linearen Problems

Wie man bereits am Beispiel der in Abschnitt 2.4.3 gegebenen Layouts sieht, möchte man in der Realität oftmals eine Belegungen von Tastaturen berechnen, für die statt $m = n$ nur $m \geq n$ gilt. Um diese Anforderungen zu erfüllen ohne wie in Abschnitt 3.2.5 vorgeschlagen zusätzliche Zeichen einführen zu müssen, ist man an einer Verallgemeinerung der bisher erarbeiteten Theorie interessiert, welche für den Spezialfall $m = n$ geschaffen wurde.

3.3.1 Übertragung der Ansätze

Erinnert man sich zurück an das quadratischen Optimierungsproblems (P2) in Abschnitt 3.2.1, welches die Suche nach einer Tastenbelegung mit minimaler zurückzulegender Fingerdistanz für den Fall $m = n$ modelliert. Der Zusammenhang $m = n$ ist auf dem Weg zur Formulierung des QAP lediglich in die Nebenbedingungen der Form $\sum_{i=1}^n y_{ij} = 1$ (siehe (3.5)) eingeflossen. Durch die Verwendung ihrer allgemeineren Form $\sum_{i=1}^n y_{ij} \leq 1$ (siehe (3.3)) erhält man ein *Generalized Quadratic Assignment Problem* (GQAP) [Lee u. Ma, 2004]. Folgt man analog zur Linearisierung des QAP den Ideen aus Abschnitt 3.2.2, so erhält man, ausgehend vom GQAP, ein ganzzahliges, lineares Optimierungsproblem, das theoretisch dafür geeignet ist, Distanz-optimale Tastenbelegungen für den Fall $m \geq n$ zu errechnen. Es ist das verallgemeinerte Analogon zu (P4) und lautet:

$$\begin{aligned}
 & \min (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} \\
 \text{u.d.N. } & \sum_{i=1}^n y_{ij} \leq 1 \quad \forall j \in [m] \\
 & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \\
 & y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall i, k \in [n], j, l \in [m] \\
 & 2 \cdot x_{ijkl} \leq y_{ij} + y_{kl} \quad \forall i, k \in [n], j, l \in [m] \\
 & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \\
 & x_{ijkl} \in \{0, 1\} \quad \forall i, k \in [n], j, l \in [m]
 \end{aligned} \tag{P13}$$

Die Anzahl der Variablen, sowie der Nebenbedingungen bleibt im Vergleich zu (P4) unverändert. Es ändert sich nur das Verhältnis zwischen Gleichungs- und Ungleichungsbedingungen zugunsten letztgenannter. Die Eigenschaften der Beziehung zwischen dem QAP (P2) und Problem (P4) übertragen sich auf die Beziehung zwischen dem GQAP und (P13). Erwartungsgemäß ändert sich auch an der zeitlichen Unlösbarkeit des Problems für realistische Größenordnungen nichts, weshalb obige Formulierung für die Berechnung von Tastenbelegungen in der Praxis nicht genutzt werden kann. Aus diesem Grund sind Verbesserungen nötig, welche die Lösbarkeit sicherstellen. Im Prinzip eignen sich dafür die Ideen aus Abschnitt 3.2.3, die jedoch auf die veränderte Ausgangssituation ($m \geq n$) angepasst werden müssen. Um den Rahmen der Arbeit nicht zu sprengen, betrachtet man die Veränderungen, welche eine Relaxierung der Annahme $m = n$ hin zu $m \geq n$ in den Ansätzen AJLIN und SYMM(e, H) zur Folge hat, denn sie lieferten die besten Ergebnisse in den Evaluationsphasen (siehe Tabelle 5).

Die Zunahme der Tasten im Vergleich zur Buchstabenanzahl hat keinen Einfluss auf die Nebenbedingungen der Form $\sum_j y_{ij} = 1$ und $x_{ijkl} = x_{klij}$, da nach wie vor jeder Buchstaben genau

einmal verteilt wird und die x -Variablen per Konstruktion paarweise gleich sein müssen. Auch die Nebenbedingung $\sum_{j \in T} y_{ij} = 1$ aus SYMM kann unverändert bleiben, da das Verhältnis der Tastenanzahl zur Buchstabenanzahl die Fixierung nicht beeinflusst.

Eine Anpassung ist für die Nebenbedingungen der Form $\sum_i y_{ij} = 1$ nötig, die (analog zu (P13)) durch die Nebenbedingungen $\sum_i y_{ij} \leq 1$ (siehe Ungleichungen (3.3)) ersetzt werden.

Die Nebenbedingungen $\sum_j x_{ijkl} = y_{kl}$ bleiben bestehen, da sich für jede Wahl zweier Buchstaben $\tilde{i}, \tilde{k} \in [n]$ und einer Taste $\tilde{l} \in [m]$ eine weitere Taste $\tilde{j} \in [m]$ finden lässt, auf der \tilde{i} liegt. Damit ist der Wert der Summe $\sum_j x_{i\tilde{j}\tilde{k}\tilde{l}}$ nach wie vor gleich dem Wert von $y_{\tilde{k}\tilde{l}}$.

Im Fall der Nebenbedingungen $\sum_i x_{ijkl} = y_{kl}$ kann man nicht für jede Wahl $\tilde{k} \in [n]$ und $\tilde{j}, \tilde{l} \in [m]$ garantieren, dass es einen Buchstaben $\tilde{i} \in [n]$ gibt, der auf Taste \tilde{j} liegt, da für $m \geq n$ die Taste \tilde{j} möglicherweise leer ist. Die Tasten, welche leer bleiben, lassen sich natürlich erst in der Lösung (also nach der Optimierung) identifizieren (ansonsten wären wir ja wieder im Fall $m = n$), weshalb die Nebenbedingungen folgendermaßen angepasst werden müssen:

$$\sum_{i=1}^n x_{ijkl} \leq y_{kl} \quad \forall k \in [n], j, l \in [m] \quad (3.30)$$

Durch die Anpassung dieser $m^2 n$ Nebenbedingungen ergibt sich eine allgemeine Form von Problem (P9):

$$\begin{aligned} & \min (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot d_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m d_j \cdot p_i \cdot y_{ij} \\ \text{u.d.N. } & \sum_{i=1}^n y_{ij} \leq 1 \quad \forall j \in [m] \\ & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \\ & \sum_{j \in T} y_{ij} = 1 \\ & \sum_{i=1}^n x_{ijkl} \leq y_{kl} \quad \forall k \in [n], j, l \in [m] \\ & \sum_{j=1}^m x_{ijkl} = y_{kl} \quad \forall i, k \in [n], l \in [m] \\ & x_{ijkl} = x_{klij} \quad \forall i, k \in [n], j, l \in [m] \\ & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \\ & x_{ijkl} \geq 0 \quad \forall i, k \in [n], j, l \in [m] \end{aligned} \quad (P14)$$

Ausgehend von (P14) wollen wir das in Abschnitt 3.2.4 entwickelte Verfahren des impliziten Nullsetzens (IMPLZERO) anwenden. Dabei muss geprüft werden, ob die in (P14) verwendeten Nebenbedingungen, welche x -Variablen enthalten, ihre Gültigkeit behalten. Man stellt fest,

dass im Gegensatz zum Fall $m = n$ nur zwei der drei Gruppen von Nebenbedingungen korrigiert werden müssen. Dies liegt daran, dass die ursprünglichen Adams-Johnson-Bedingungen der Form $\sum_i x_{ijkl} = y_{kl}$ zuvor bereits durch die Verallgemeinerung auf $m \geq n$ zu den Ungleichungen (3.30) aufgeweicht wurden. Damit verbleiben die Gleichungen $\sum_j x_{ijkl} = y_{kl}$ und $x_{ijkl} = x_{klij}$, welche an das implizite Nullsetzen angepasst werden müssen. Dies geschieht analog zu Abschnitt 3.2.4.

Eine Reduktion der Einträge der Bigrammwahrscheinlichkeitsmatrix ist auch im allgemeinen Fall nötig, um die Lösbarkeit für realistische Instanzen zu gewährleisten. Dabei wird aufgrund der Erfahrungen aus Abschnitt 3.2.4 der Ansatz THRESHOLD verwendet.

3.3.2 Formulierung eines verallgemeinerten Problems

Mithilfe der in Abschnitt 3.3.1 verallgemeinerten Ansätze erhält man folgendes lineares, ganzzahliges Optimierungsproblem:

$$\begin{aligned}
 \min \quad & \frac{B-1}{\sum_{i,k=1}^n \tilde{p}_{ik}} \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m \tilde{p}_{ik} \cdot d_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m d_j \cdot p_i \cdot y_{ij} \\
 \text{u.d.N.} \quad & \sum_{i=1}^n y_{ij} \leq 1 \quad \forall j \in [m] \\
 & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n], i \neq \tilde{i} \\
 & \sum_{j \in T} y_{\tilde{i}j} = 1 \\
 & x_{ijkl} = 0 \quad \forall i, j, k, l \in \tilde{I}^0 \\
 & y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall i, j, k, l \in I \setminus \tilde{I}^0 \\
 & \sum_{i=1}^n x_{ijkl} \leq y_{kl} \quad \forall j, k, l \in I_{jkl} \\
 & \sum_{j=1}^m x_{ijkl} = y_{kl} \quad \forall i, k, l \in I_{ikl} \setminus \tilde{I}_{ikl}^0 \\
 & \sum_{j=1}^m x_{ijkl} \leq y_{kl} \quad \forall i, k, l \in \tilde{I}_{ikl}^0 \\
 & x_{ijkl} = x_{klij} \quad \forall i, j, k, l \in I \setminus \tilde{I}^0 \\
 & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \\
 & x_{ijkl} \geq 0 \quad \forall i, k \in [n], j, l \in [m]
 \end{aligned} \tag{P15}$$

Dieses Optimierungsproblem mit $m^2n^2 + mn - |\tilde{I}^0|$ Variablen und $2 \cdot |I \setminus \tilde{I}^0| + m^2n + n^2m + m+n$ Nebenbedingungen (ohne die Ganzzahligkeits-, Positivitäts- und Fixierungsbedingungen)

eignet sich für die Suche nach einer optimalen Tastenbelegung im Fall $m \geq n$. Bis auf die Nebenbedingung der Form $\sum_{i=1}^n y_{ij} \leq 1$ ist es identisch zu Problem (P12), in welchem die genannten Nebenbedingungen als Gleichungen auftreten.

3.3.3 Ergebnisse der Optimierung

Die Interpretation der Lösungen von (P15) für die Layouts aus Abschnitt 2.4.3 mit $m > n$ als Tastenbelegungen liefert folgende Ergebnisse (Parameterwahl für THR und SYMM: $\tilde{p} = 0,6\%$, e auf H_R).

Abbildung 24 zeigt die optimierte *QWERTY-like*-Tastenbelegung mit einer zurückzulegenden Gesamtdistanz von 198,884 LE (Zielfunktionswert 131,107 LE), welche man in 114.5 s aus der Optimierung erhält.

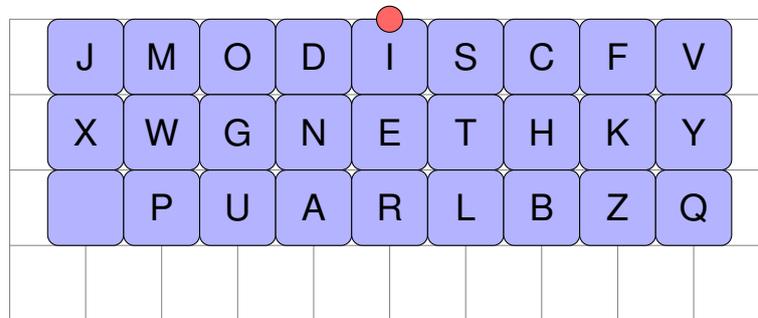


Abbildung 24: Distanz-optimierte Tastenbelegung für das 3x9-Layout bei realistischer Schreibweise im vertikalen Fall.

In 222.8 s erhalten wir eine Tastenbelegung für das 3x9-Layout (siehe Abb. 25) mit einer zurückzulegenden Distanz von 199,199 LE (Zielfunktionswert: 131,139 LE).

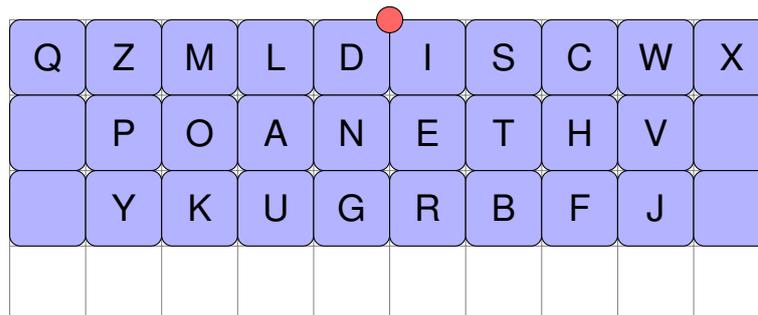


Abbildung 25: Distanz-optimierte Tastenbelegung für das 3x10-Layout bei realistischer Schreibweise im vertikalen Fall.

Die Verwendung der Tastenbelegung aus Abbildung 26 erfordert das Zurücklegen einer Distanz von 185,922 LE. Sie resultiert mit einem Zielfunktionswert von 129,756 LE in 362,2 s aus der Optimierung der Belegung des 4x8-Layouts.

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

Abbildung 26: Distanz-optimierte Tastenbelegung für das 4x8-Layout bei realistischer Schreibweise im vertikalen Fall.

Bei der Betrachtung der aus (P12) und (P15) resultierenden Belegungen für das *QWERTY-like*-Layout (siehe Abb. 23) beziehungsweise das 3x9-, 3x10- und 4x8-Layout (siehe Abb. 24, 25 und 26) lassen sich einige gemeinsame Eigenschaften feststellen.

So sind auf allen Belegungen die Buchstaben *D*, *E*, *I*, *N*, *R*, *S* und *T* sehr nahe zusammen und darüber hinaus auch nahe am Startpunkt. Der Grund dafür ist, dass diese Buchstaben und die dazugehörigen Bigramme jene sind, welche am Häufigsten in deutschen Texten auftreten (siehe Tabellen 24 und 25 in Anhang A.1). Aus ihnen lässt sich bereits eine Vielzahl von häufig benutzten Wörtern wie "er", "in", "der", "die", "ist", "den" übertragen, ohne dass der Finger große Strecken zurücklegen muss, weshalb diese Positionierung intuitiv ist.

Dass nicht alle Bigramm-Wahrscheinlichkeiten in die Optimierung eingeflossen sind, kann man beispielsweise anhand von Buchstabenkombinationen aus *A* und *H* erkennen, da sie auf allen Layouts eher weit auseinander liegen, obwohl beide Bigramme nicht selten sind. Da aber die zugehörigen Wahrscheinlichkeiten knapp unterhalb der Schwelle \tilde{p} liegen, werden sie gleich Null gesetzt und es existiert keine Motivation für eine nahe Positionierung.

Analog zu den Tastenbelegungen aus Abschnitt 3.1.2 sind seltene Buchstaben wie *Q* oder *X* sinnvollerweise auf fernen Taste untergebracht, da keines ihrer Bigramme in die Optimierung einfließt und der Weg vom Startpunkt zur zugehörigen Taste selten zurückgelegt werden muss.

3.4 Zusammenfassung der Ergebnisse im vertikalen Fall

Um künftig eine Unterscheidung zwischen den Fällen $m = n$ und $m > n$ beziehungsweise zwischen den zugehörigen finalen Optimierungsproblemen (siehe (P12) und (P15)) überflüssig zu machen, fassen wir die in Kapitel 3 erarbeitete Theorie zur Suche nach einer distanzminimierenden vertikalen Tastenbelegung in folgendem Optimierungsproblem zusammen:

$$\min \frac{B-1}{\sum_{i,k=1}^n \tilde{p}_{ik}} \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m \tilde{p}_{ik} \cdot d_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m d_j \cdot p_i \cdot y_{ij} \quad (\text{P16})$$

$$\text{u.d.N. } (y, x) \in \mathcal{F}_{P16} := \begin{cases} \mathcal{F}_{P12} & \text{falls } m = n \\ \mathcal{F}_{P15} & \text{falls } m > n \end{cases}$$

Tabelle 8 bietet einen Überblick über die Ergebnisse der Optimierung von Tastenbelegungen im vertikalen Fall. Für die realistische Schreibweise wurde $\tilde{p} = 0,6\%$ gewählt, sowie SYMM(e, H_R) verwendet. Die Zielfunktionswerte in Spalte 6, sowie die Werte der zurückzulegenden Gesamtdistanz (Spalte 7) sind in LE angegeben. Die dritte Spalte gibt an, ob von einer vereinfachten Schreibweise des Benutzers ausgegangen wird, was auf ein lineares Zuordnungsproblem führt (siehe Abschnitt 3.1) oder ob die Suche nach einer optimalen Tastenbelegung auf der realistischen Schreibweise aus Abschnitt 3.2 beruht. Die übrigen Spalten liefern wie gewohnt das Layout, die Anzahl der Nebenbedingungen und Variablen, sowie die Laufzeit der Optimierung.

Tabelle 8: Ergebnisse der Distanzminimierung für vertikale Anwendungen.

Layout	Schreibweise	NB	Var.	Laufzeit	Zfcts.wert	Distanz
QWERTY-like	vereinfacht	52	676	0,08 s	372,414	372,414
QWERTY-like	realistisch	25040	17514	181,6 s	131,393	220,372
3x9-Layout	vereinfacht	53	702	0,08 s	368,661	368,661
3x9 (Dummy)	realistisch	27164	19047	176,8 s	131,107	198,297
3x9-Layout	realistisch	27163	19020	114,5 s	131,107	198,884
3x10-Layout	vereinfacht	56	780	0,09 s	372,685	372,685
3x10 (Dummy)	realistisch	32975	23070	253,3 s	131,139	196,523
3x10-Layout	realistisch	32971	22950	222,8 s	131,139	199,199
4x8-Layout	vereinfacht	58	832	0,1 s	371,037	371,037
4x8 (Dummy)	realistisch	37604	26304	374,1 s	129,756	189,946
4x8-Layout	realistisch	37598	26112	362,2 s	129,756	185,922

4 Distanzminimierung für horizontale Anwendungen

Nachdem wir uns in Kapitel 3 ausgiebig mit der Minimierung der zurückzulegenden Strecke für einen Finger beschäftigt haben, wollen wir unsere Aufmerksamkeit im folgenden Kapitel dem distanzoptimalen Schreiben mit zwei Fingern widmen. Diese Art der Textübertragung wurde insbesondere seit der Integration von Lagesensoren in neue Smartphone-Geräte immer populärer (siehe Abschnitt 1.1 und 1.2). Ein Vorteil der horizontalen Schreibweise ist unter anderem die stabilere Handhabung, da das Gerät mit beiden Händen fixiert wird. Außerdem wird durch die Nutzung beider Hände die Belastung pro Finger reduziert, da im Gegensatz zum vertikalen Fall kein Finger die gesamte Schreibarbeit übernehmen muss. Hinsichtlich der Ermüdung hat dies einen positiven Einfluss. Wie in Abschnitt 2.3 beschrieben, will man als Lösung der Optimierung die Tastenbelegung einer gesplitteten Tastatur erhalten. Im Mittelpunkt steht also die Frage nach einer distanzoptimalen Verteilung der Buchstaben, wenn jedem Daumen ein eindeutiger Tastaturbereich und damit eine eindeutige Menge an Tasten zugeordnet ist.

4.1 Vereinfachte Modellierung der beidhändigen Bedienung

4.1.1 Beschreibung und mathematische Formulierung

Erinnert man sich zurück an die vereinfachte, vertikale Schreibweise aus Abschnitt 3.1 und überträgt diesen Ansatz auf den beidhändigen Fall, so erhält man die sogenannte vereinfachte, horizontale Schreibweise, welche darauf basiert, dass die Finger nach einem getippten Buchstaben zum jeweils zugeordneten Startpunkt zurückkehren (siehe Abb. 27). Man stellt fest, dass für eine erfolgreiche Übertragung lediglich eine Veränderung der Inputdaten notwendig ist, genauer gesagt benötigt man statt d_j (siehe (P1) und Abschnitt 2.4.4) nun die Werte $\tilde{d}_j \geq 0$, die jeweils den Abstand einer Taste $j \in [m]$ zum ihr zugeordneten Startpunkt angeben.

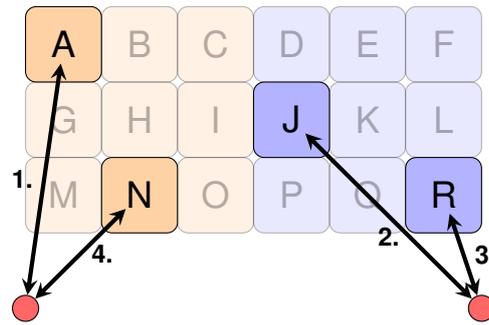


Abbildung 27: Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der vereinfachten Schreibweise in horizontaler Geräteposition.

Nachdem in Abschnitt 2.4.3 eine Aufteilung $T_L \dot{\cup} T_R = [m]$ der Tasten auf die beiden Layout-Seiten gewählt wurde, lassen sich die Werte dank der in Abschnitt 2.4.4 gewählten Startpunkte $SP(S)$ für alle Tastaturlayouts mithilfe der 2-Norm berechnen. Analog zu Abschnitt 2.4.4 gilt demnach:

$$\tilde{d}_j := \|SP(S) - J\|_2 \quad \forall j \in T_S, S \in \{L, R\} \quad (4.1)$$

Abgesehen von den veränderten Input-Daten entspricht das Optimierungsproblem, welches die Suche nach einer optimalen Tastenbelegung für die vereinfachte Schreibweise im horizontalen Fall beschreibt, dem Problem (P1) aus Abschnitt 3.1. Also:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^m 2 \cdot \tilde{d}_j \cdot p_i \cdot y_{ij} \\ \text{u.d.N.} \quad & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n] \\ & \sum_{i=1}^n y_{ij} \leq 1 \quad \forall j \in [m] \\ & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \end{aligned} \quad (\text{P17})$$

Die Eigenschaften der Probleme (P1) und (P17) sind identisch.

4.1.2 Ergebnisse der Optimierung

Übergibt man die Buchstabenwahrscheinlichkeiten aus Tabelle 25 in Anhang A.1, sowie die Informationen über die Entfernungen \tilde{d}_j der Tasten zum jeweiligen zugeordneten Startpunkt (siehe Tabelle 28, 32, 36 und 41 in Anhang A.2) an die Optimierungssoftware, dann besitzt das Optimierungsproblem für das *QWERTY-like*-Layout in 0,06 s folgende Lösung.

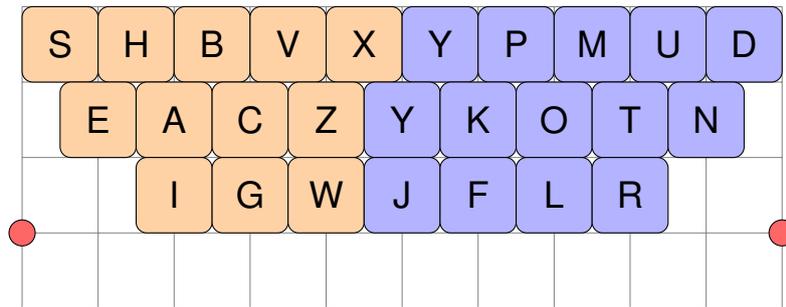


Abbildung 28: Distanz-optimale Tastenbelegung für das QWERTY-like-Layout bei vereinfachter Schreibweise im horizontalen Fall.

Die zurückzulegende Entfernung für die Tastenbelegung aus Abbildung 28 ist gleich 512,27 LE. Einen geringeren Zielfunktionswert von 445,494 LE ergibt die in 0,08 s berechnete Tastenbelegung für das 3x9-Layout (siehe Abb. 29).

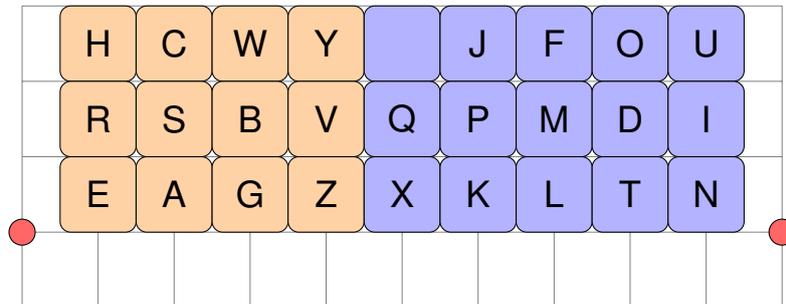


Abbildung 29: Distanz-optimale Tastenbelegung für das 3x9-Layout bei vereinfachter Schreibweise im horizontalen Fall.

Für das 3x10-Layout erhalten wir in 0,09 s eine für die vereinfachte, horizontale Schreibweise optimierte Tastenbelegung (siehe Abbildung 30). Die zurückzulegende Distanz beträgt 372,685 LE.

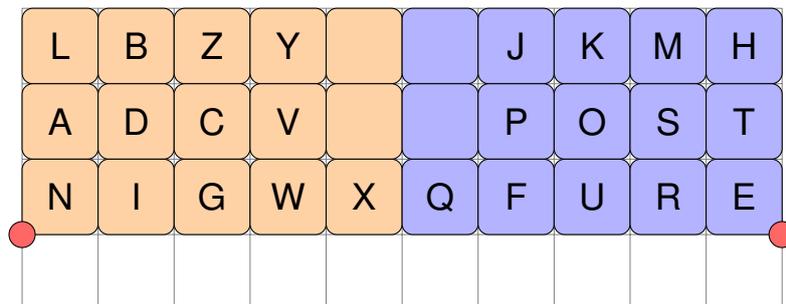


Abbildung 30: Distanz-optimale Tastenbelegung für das 3x10-Layout bei vereinfachter Schreibweise im horizontalen Fall.

Die aus der Optimierung (Laufzeit: 0,11 s) resultierende Tastenbelegung für das 4x8-Layout in

Abbildung 31 erfordert einen Weg von 453 676 LE.

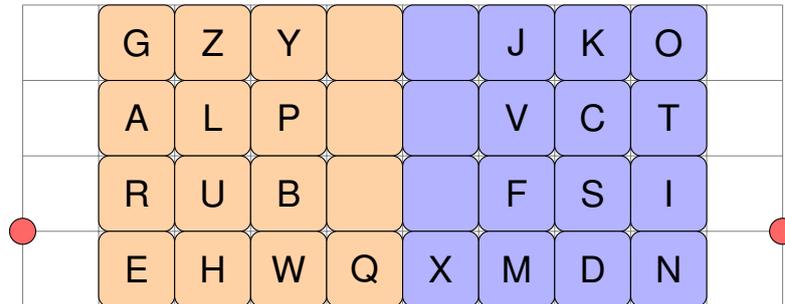


Abbildung 31: Distanz-optimale Tastenbelegung für das 4x8-Layout bei vereinfachter Schreibweise im horizontalen Fall.

Analog zu den Ergebnissen im vertikalen Fall (siehe Abschnitt 3.1.2) stellt man fest, dass unabhängig vom Layout die Buchstaben mit hoher Wahrscheinlichkeit nahe an den Startpunkten positioniert werden, wohingegen seltene und leere Tasten die größte Entfernung aufweisen.

Während die zurückzulegenden Gesamtdistanzen der Layouts bei einhändiger, vereinfachter Bedienung sehr ähnlich sind (siehe Tabelle 8), lassen sich für die vereinfachte Schreibweise mit beiden Händen große Abweichungen in den Zielfunktionswerten ausmachen (bis zu 37,5%). Der Grund dafür ist, dass die Anzahl an Tasten in geringer Entfernung zum Startpunkt, welche in Abschnitt 3.1 als bedeutendes Kriterium für einen geringen Zielfunktionswert identifiziert wurde, stark variiert. So liegt beim *QWERTY-like*-Layout beispielsweise keine Taste im Bereich von 1,6 LE um den Startpunkt, bei den anderen Layouts sind es 2 (*3x9*-Layout), 4 (*4x8*-Layout) oder sogar 6 (*3x10*-Layout) Tasten innerhalb dieser Entfernung. Dementsprechend weist die Tastenbelegung des *QWERTY-like*-Layouts den größten und die des *3x10*-Layouts den geringsten Zielfunktionswert auf.

4.2 Realistische Modellierung der beidhändigen Bedienung

Wie schon bei der einhändigen Bedienung eines Smartphones (siehe Kapitel 3) entspricht das Zurückkehren zum Startpunkt zwischen je zwei Zeichen in Abschnitt 4.1 nicht der allgemein verwendeten beidhändigen Schreibweise. Vielmehr ist es in der Praxis so, dass bei der Verwendung gesplitteter Tastaturen die Strecke zwischen zwei Zeichen, die demselben Finger zugeordnet sind, ohne Umweg über den Startpunkt zurückgelegt wird. Für die Modellierung der zurückzulegenden Strecke eines Bigramms, das von beiden Fingern getippt wird, führt man Wartepunkte $WP(S) \forall S \in \{L, R\}$ ein, auf denen ein temporär nicht benutzter Finger pausieren kann (siehe Abschnitt 2.4.4). In den folgenden Abschnitten werden mehrere Möglichkeiten für die Wahl der Wartepositionen vorgestellt.

4.2.1 Naive Wahl der Wartepositionen

Ein mögliches Bewegungsmuster für die Eingabe von Bigrammen, deren Buchstaben auf verschiedenen Seiten der Tastatur liegen, ist, dass der Finger, welcher das erste Zeichen tippt zum Startpunkt zurückkehrt und der andere Finger sich vom Startpunkt zum zweiten Zeichen bewegt. Der Wartepunkt $WP(S)$ entspricht also dem Startpunkt $SP(S)$, weshalb die Entfernung \hat{d}_j einer Taste j zum ihr zugeordneten Wartepunkt gleich der Entfernung \tilde{d}_j zum ihr zugeordneten Startpunkt ist (siehe auch Abschnitt 2.4.4). Anschaulich resultiert daraus die sogenannte semi-realistische, horizontale Schreibweise in Abbildung 32.

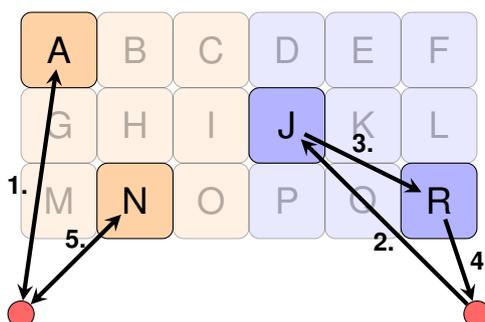


Abbildung 32: Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der semirealistischen Schreibweise in horizontaler Geräteposition. Dabei kehrt der linke Finger für das zweiseitige Bigramm "AJ" auf den (roten) Startpunkt zurück.

Folgender Ausdruck liefert die erwartete, durchschnittlich zurückzulegende Distanz für Bigramme auf der Seite $S \in \{L, R\}$:

$$\sum_{i,k \in [n]} \sum_{j,l \in T_S} p_{ik} \cdot d_{jl} \cdot x_{ijkl} \quad (4.2)$$

Dabei bezeichnet d_{jl} wie gewohnt die Distanz zwischen der Taste $j \in [m]$ und der Taste $l \in [m]$ (siehe (2.1) in Abschnitt 2.4.4). Für Bigramme, deren Buchstaben auf verschiedenen Seiten liegen und die somit beide Finger beschäftigen, ist die zurückzulegende Strecke gegeben durch:

$$\sum_{S \in \{L,R\}} \sum_{i,k \in [n]} \sum_{j \in T_S} \sum_{l \notin T_S} p_{ik} \cdot (\tilde{d}_j + \tilde{d}_l) \cdot x_{ijkl} \quad (4.3)$$

Abermals wird davon ausgegangen, dass sich die Finger am Anfang und am Ende des Textes auf den Startpunkten befinden. Die erwartete Länge des Hin- und Rückwegs für den Finger der Seite $S \in \{L, R\}$ wird durch folgenden Ausdruck modelliert:

$$2 \cdot \sum_{i \in [n]} \sum_{j \in T_S} \tilde{d}_j \cdot p_i \cdot y_{ij} \quad (4.4)$$

Aus den obigen Termen lässt sich die Zielfunktion unseres Optimierungsproblems erzeugen.

Sie lautet:

$$\begin{aligned}
 & (B-1) \cdot \sum_{S \in \{L,R\}} \sum_{i,k \in [n]} \sum_{j,l \in T_S} p_{ik} \cdot d_{jl} \cdot x_{ijkl} \\
 & + (B-1) \cdot \sum_{S \in \{L,R\}} \sum_{i,k \in [n]} \sum_{j \in T_S} \sum_{l \notin T_S} p_{ik} \cdot (\tilde{d}_j + \tilde{d}_l) \cdot x_{ijkl} \\
 & + 2 \cdot \sum_{S \in \{L,R\}} \sum_{i \in [n]} \sum_{j \in T_S} \tilde{d}_j \cdot p_i \cdot y_{ij}
 \end{aligned} \tag{4.5}$$

Mithilfe einer geeigneten Definition von \tilde{d}_{jl} lässt sich die Funktion (4.5) zusammenfassen zu:

$$(B-1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot \tilde{d}_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m \tilde{d}_j \cdot p_i \cdot y_{ij} \tag{4.6}$$

mit

$$\tilde{d}_{jl} := \begin{cases} d_{jl}, & \text{für } j, l \in T_L \vee j, l \in T_R \\ \tilde{d}_j + \tilde{d}_l, & \text{sonst.} \end{cases} \tag{4.7}$$

Damit beschreibt Definition (4.7) die für ein Bigramm zurückzulegende Strecke in Abhängigkeit davon, ob dessen Buchstaben auf einer oder beiden Seiten liegen.

Die Formulierung des Optimierungsproblems verläuft analog zu Abschnitt 3.2.1. Darüberhinaus kann auch die Linearisierung und die Ansätze zur Verbesserung der Lösbarkeit (siehe Abschnitt 3.2.3 und 3.2.4) aus dem vertikalen Fall übernommen werden. Damit entsteht ein Problem, das bis auf die Inputdaten identisch ist zum Optimierungsproblem für die Suche nach einer vertikalen Tastenbelegung (siehe (P16)).

$$\min \frac{B-1}{\sum_{i,k=1}^n \tilde{p}_{ik}} \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m \tilde{p}_{ik} \cdot \tilde{d}_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m \tilde{d}_j \cdot p_i \cdot y_{ij} \tag{P18}$$

$$\text{u.d.N. } (y, x) \in \mathcal{F}_{P16}$$

Während in Abschnitt 4.1 der Zielfunktionswert die tatsächlich zurückzulegende Distanz angibt, verhält es sich in diesem Fall wie in Abschnitt 3.2. Durch die Reduktion der Einträge der Wahrscheinlichkeitsmatrix, die für die Lösbarkeit des Problems aus zeitlicher Sicht von großer Bedeutung ist, gibt der Zielfunktionswert nur die Distanz an, die für eine Übertragung der nicht-unterdrückten Bigramme nötig wäre.

Es wird an dieser Stelle auf eine grafische Präsentation der Ergebnisse verzichtet, da die semi-realistische, horizontale Schreibweise Schwächen im Bezug auf die Realität aufweist, welche im folgenden Abschnitt durch eine bessere Wartepunktwahl korrigiert werden. Eine Auflistung der semi-realistischen Resultate findet sich in Tabelle 9.

4.2.2 Praxisnahe Wahl der Wartepositionen

Trotz der auf den ersten Blick zufriedenstellenden Ergebnisse aus Abschnitt 4.2.1 (siehe Tabelle 9) stellt sich die Frage, ob die zugrundeliegende Annahme $WP(S) = SP(S)$ der Realität entspricht. Also ob der Benutzer für ein zweiseitiges Bigramm tatsächlich mit dem ersten Finger zum zugehörigen Startpunkt zurückkehrt und sich mit dem anderen Finger von dessen Startpunkt aus dem zweiten Buchstaben nähert. Realistisch gesehen ist diese Frage mit Nein zu beantworten, vielmehr ist in der Praxis eine andere Schreibweise zu beobachten (siehe Abb. 33). Bigramme, die auf einer Seite liegen, werden vom Benutzer direkt, ohne Umwege angesteuert und getippt (analog zu Abschnitt 4.2.1). Für Bigramme, die auf verschiedenen Seiten der Tastatur liegen, beobachtet man nach dem Tippvorgang des ersten Fingers ein Zurückziehen auf eine schwebende, mittige Warteposition.

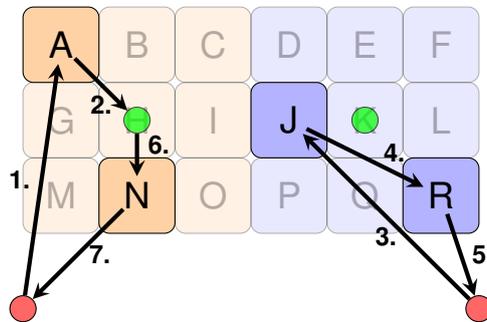


Abbildung 33: Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der realistischen Schreibweise in horizontaler Geräteposition. Zwischen der zweiten und dritten Bewegung wird der linke (grüne) Wartepunkt benutzt, da "A" und "J" auf verschiedenen Seiten liegen.

Um optimierte Tastenbelegungen für diese sogenannte realistische (horizontale) Schreibweise berechnen zu können, benötigt man die Entfernungen $\hat{d}_j \geq 0$ der Tasten zu den jeweiligen Wartepunkten $WP(S) \in \mathbb{R}^2$. Mit der Wahl der Wartepunkte $WP(S)$ als Mittelpunkte $MP(S)$ der jeweiligen Tastaturhälfte der Seite $S \in \{L, R\}$, lassen sich die Distanzen \hat{d}_j durch folgenden Ausdruck bestimmen (siehe (2.4) in Abschnitt 2.4.4):

$$\hat{d}_j := \|WP(S) - J\|_2 = \|MP(S) - J\|_2 \quad \forall j \in T_S, S \in \{L, R\} \quad (4.8)$$

Um die Tasten $j \in [m]$ und $l \in [m]$ aufeinanderfolgend anzutippen, muss bei Verwendung der realistischen, horizontalen Schreibweise folgende Entfernung zurückgelegt werden (vergleiche Definition (4.7)):

$$\hat{d}_{jl} := \begin{cases} d_{jl}, & \text{für } j, l \in T_L \vee j, l \in T_R \\ \hat{d}_j + \hat{d}_l, & \text{sonst.} \end{cases} \quad (4.9)$$

Integriert man diese Neuerung in Optimierungsproblem (P18) aus Abschnitt 4.2.1, dann ergibt

sich:

$$\min \frac{B-1}{\sum_{i,k=1}^n \tilde{p}_{ik}} \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m \tilde{p}_{ik} \cdot \hat{d}_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m \hat{d}_j \cdot p_i \cdot y_{ij} \quad (\text{P19})$$

u.d.N. $(y, x) \in \mathcal{F}_{P16}$

Dieses Problem ist bis auf die veränderten Inputdaten \hat{d}_{jl} identisch zu den Problemen (P16) und (P18) aus dem vertikalen beziehungsweise dem semi-realistischen, horizontalen Fall.

4.2.3 Ergebnisse der Optimierung

Neben den Wahrscheinlichkeiten aus Anhang A.1 benötigt man die Layout-spezifischen Werte der Distanzen \hat{d}_{jl} für alle Tastenpaare $(j, l) \in [m] \times [m]$. Ihre Bestimmung erfolgt automatisch mithilfe der eingelesenen Distanzen \hat{d}_j und d_{jl} aus den Layout-spezifischen Tabellen in Anhang A.2 durch das Tool *VirtualKeyboardOptimization.mos* (siehe Kapitel 7). Folgende Tastenbelegungen sind die Ergebnisse der Optimierung mit $\text{THR}(0,6\%)$ und $\text{SYMM}(e, H_R)$.

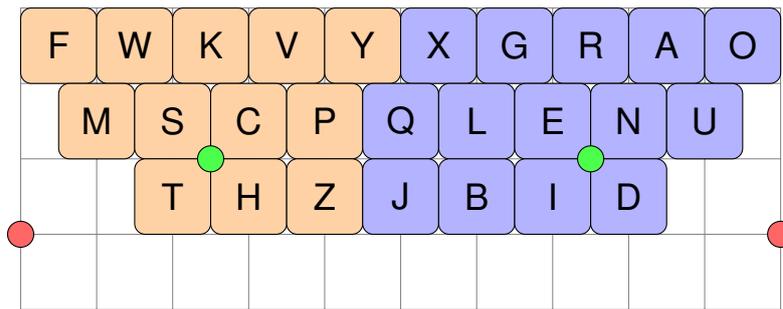


Abbildung 34: Distanz-optimierte Tastenbelegung für das QWERTY-like-Layout bei realistischer Schreibweise im horizontalen Fall.

Die Tastenbelegung (Zielfunktionswert: 128,542 LE) für das QWERTY-like-Layout bei horizontaler Geräteposition und realistischer Schreibweise aus Abbildung 34 erhalten wir in 327,6 s. Die zurückzulegende Gesamtstrecke beträgt 183,791 LE.

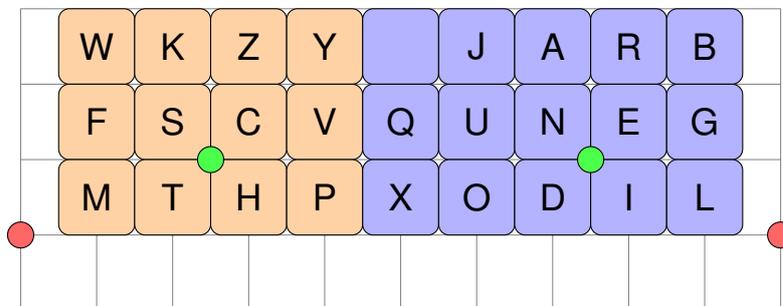


Abbildung 35: Distanz-optimierte Tastenbelegung für das 3x9-Layout bei realistischer Schreibweise im horizontalen Fall.

Im Vergleich zur optimierten *QWERTY-like*-Belegung müssen die Finger auf der optimierten Tastenbelegung des 3x9-Layouts (siehe Abb. 35) mit Zielfunktionswert 127,641 LE weniger Weg zurücklegen, nämlich nur 173,236 LE.

Die Optimierung der Tastenbelegung für das 3x10-Layout benötigt 283,5 s und liefert eine Tastenbelegung mit Zielfunktionswert 121,357 LE und Gesamtdistanz 167,751 LE (siehe Abb. 36).

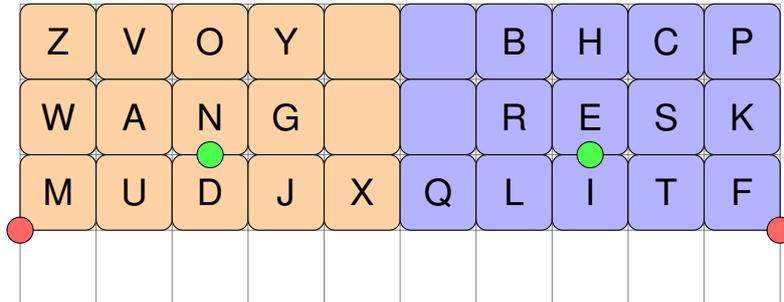


Abbildung 36: Distanz-optimierte Tastenbelegung für das 3x10-Layout bei realistischer Schreibweise im horizontalen Fall.

Eine Strecke von 163,457 LE muss auf der optimierten Tastenbelegung des 4x8-Layouts (siehe Abb. 37) zurückgelegt werden. Sie ist das Resultat der 298,9 s dauernden Optimierung mit Zielfunktionswert 120,874 LE.

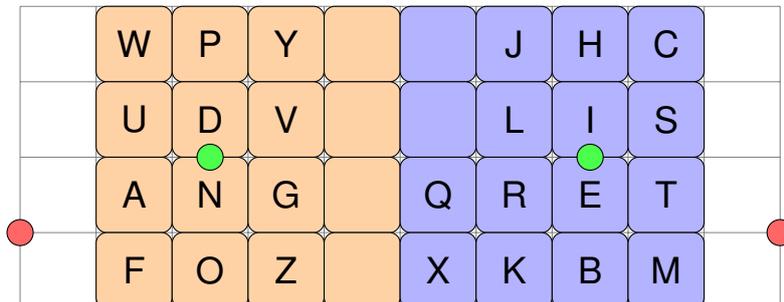


Abbildung 37: Distanz-optimierte Tastenbelegung für das 4x8-Layout bei realistischer Schreibweise im horizontalen Fall.

Vergleicht man die horizontalen Tastenbelegungen mit den Ergebnissen aus dem vertikalen Fall (siehe Abschnitt 3.2.5 und 3.3.3), so stellt man fest, dass die häufigen Buchstaben *D*, *E*, *I*, *N*, *R*, *S* und *T* wieder so positioniert wurden, dass die dazwischenliegenden Entfernungen möglichst gering sind. Die Gründe dafür sind wie zuvor die hohen Bigrammwahrscheinlichkeiten, welche daraus resultieren, dass viele Wörter Kombinationen der genannten Buchstaben enthalten. Im Gegensatz zu den Tastenbelegungen aus Abschnitt 3.3.3 müssen sich die genannten Buchstaben jedoch nicht auf nebeneinanderliegenden Tasten befinden, sondern können dank der Bedienung mit zwei Fingern auch für eine Platzierung auf beiden Seiten des Layouts geringe Entfernungen zueinander aufweisen. Da sich die Daumen zwischen allen zweiseitigen Bigrammen zu den Wartepunkten bewegen, müssen diese Strecken im Schreibvorgang öfter zurück-

gelegt werden als der Hin- und Rückweg. Damit verlieren die Startpunkte im Vergleich zu den Wartepunkten an Bedeutung und die häufigen Buchstaben werden bevorzugt nahe der Wartepunkte positioniert. Für Buchstaben, deren Kombinationen nicht in die Optimierung einfließen, dienen die Startpunkte nach wie vor als einziger Bezugspunkt hinsichtlich ihrer Positionierung auf der Tastatur. Analog zum vertikalen Fall werden sie gemäß ihrer Häufigkeit verteilt was dazu führt, dass leere Tasten und seltene Buchstaben in der Mitte des Layouts platziert werden. An dieser Stelle erkennt man die in Abschnitt 1.1 erwähnten konträren Anforderungen der vertikalen und horizontalen Position an die Tastenbelegungen.

4.3 Zusammenfassung der Ergebnisse im horizontalen Fall

Tabelle 9 listet die Ergebnisse der Distanzminimierung im horizontalen Fall auf. Sie ist dabei aufgebaut wie Tabelle 8 und basiert auf der identischen Wahl der Parameter und Einstellungen ($\tilde{p} = 0,6\%$, e wird auf H_R fixiert).

Tabelle 9: Ergebnisse der Distanzminimierung für horizontale Anwendungen.

Layout	Schreibweise	NB	Var,	Laufzeit	Zfkts,wert	Distanz
QWERTY-like	vereinfacht	51	676	0,06 s	512,27	512,27
QWERTY-like	semi-real.	25040	17514	61,8 s	138,531	263,585
QWERTY-like	realistisch	25040	17514	327,6 s	128,542	183,791
3x9-Layout	vereinfacht	53	702	0,08 s	445,494	445,494
3x9-Layout	semi-real.	27163	19020	130,3 s	133,508	236,318
3x9-Layout	realistisch	27163	19020	321,4 s	127,641	173,236
3x10-Layout	vereinfacht	56	780	0,09 s	372,685	372,685
3x10-Layout	semi-real.	32971	22950	493,7 s	132,701	220,963
3x10-Layout	realistisch	32971	22950	283,5 s	121,357	167,751
4x8-Layout	vereinfacht	58	832	0,11 s	453,676	453,676
4x8-Layout	semi-real.	37598	26112	184,0 s	131,898	230,462
4x8-Layout	realistisch	37598	26112	298,9 s	120,874	163,457

5 Distanzminimierung für Hybrid-Anwendungen

Die meisten Nutzer bedienen ein Smartphone in der Realität sowohl vertikal als auch horizontal. Betrachtet man die Performance bisher präsentierter Tastenbelegungen in Positionen, für welche sie nicht optimiert wurden, dann erhält man alarmierende Ergebnisse (siehe Tabelle 10). In einzelnen Fällen kommt es vor, dass die zurückzulegende Distanz in "falscher" Position um über 60 % größer ist, als die einer Tastenbelegung, welche für diese Position optimiert wurde.

Tabelle 10: Zurückzulegende Distanz (in LE) und Mehraufwand für die Texteingabe mit optimierten Tastenbelegungen in "richtiger" und "falscher" Geräteposition.

Layout + Belegung	Distanz (vert.)	Distanz (hor.)	zusätzl. Aufwand ggü. vert./hor. Optimum		mittlere Distanz
QWERTY-like (vert.opt, Abb.23)	220,372	253,207	+0 %	+38 %	236,7895
QWERTY-like (hor.opt, Abb.34)	305,514	183,791	+39 %	+0 %	244,6525
3x9-Layout (vert.opt, Abb.24)	198,884	245,767	+0 %	+42 %	222,3255
3x9-Layout (hor.opt, Abb.35)	299,024	173,236	+50 %	+0 %	236,13
3x10-Layout (vert.opt, Abb.25)	199,199	237,658	+0 %	+42 %	218,4285
3x10-Layout (hor.opt, Abb.36)	326,815	167,751	+64 %	+0 %	247,283
4x8-Layout (vert.opt, Abb.26)	185,922	246,84	+0 %	+51 %	216,381
4x8-Layout (hor.opt, Abb.37)	312,389	163,457	+68 %	+0 %	237,923

Würden solche Tastaturen in der Realität eingesetzt werden, hätte dies zur Folge, dass es große Vorbehalte gegen einen Wechsel der Geräteposition gäbe und man würde die komplette Textübertragung in der "richtigen" Position vornehmen. Da ein Wechsel der Position aber durchaus Vorteile wie eine stabilere Handhabung oder ein anderes Bildschirmverhältnis mit sich bringt, ist man an sogenannten Hybrid-Tastenbelegungen interessiert, welche für die einhändige und beidhändige Textübertragung gleichermaßen geeignet sind. Wir beschränken uns in diesem Zusammenhang auf die Erzeugung von Tastenbelegungen, welche für realistische Schreibweisen optimiert sind.

5.1 Modellierung

Betrachtet man die finalen Optimierungsprobleme für die realistische Schreibweise im vertikalen beziehungsweise horizontalen Fall (siehe (P16) in Abschnitt 3 bzw. (P19) in Abschnitt 4), so stellt man fest, dass sie in ihrer allgemeinen Form übereinstimmen. Lediglich die Koeffizienten der Zielfunktionen unterscheiden sich, da diesen unterschiedliche Distanzen zugrundeliegen. Anschaulich bedeutet das, dass die Suche nach Tastenbelegungen im gleichen Polytop stattfindet und sich lediglich durch die Richtung des Zielfunktionsvektors unterscheidet. Bei der Formulierung einer neuen Zielfunktion, mit der man in der Lage ist, optimierte Hybrid-Tastenbelegungen für die realistischen Schreibweisen zu berechnen, ist es naheliegend, über eine Verknüpfung der Vektoren, also über eine gewichtete Kombination der ursprünglichen Zielfunktionen (aus (P16) und (P19)) nachzudenken (siehe Abb. 38). Der Zulässigkeitsbereich des neuen Problems, welcher unsere Anforderungen an eine Tastenbelegung wiedergibt und Ansätze zur Verbesserung der zeitlichen Lösbarkeit enthält, ist identisch zu den finalen Problemen aus den Kapiteln 3 und 4.

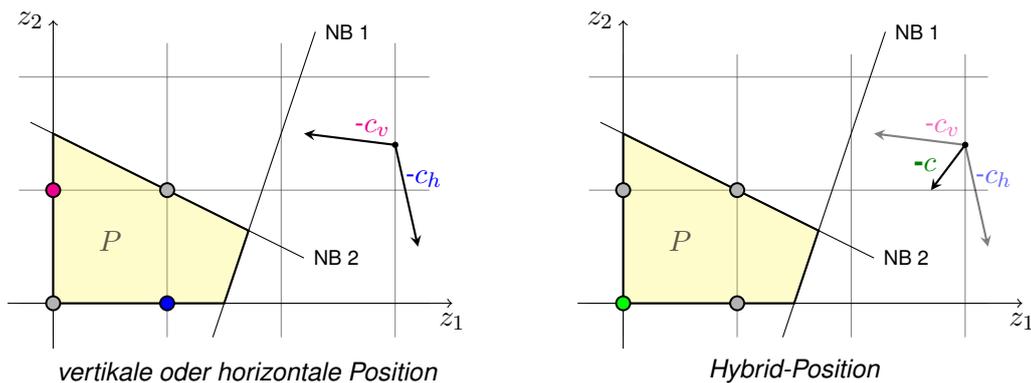


Abbildung 38: Schematische Darstellung der grundlegenden Idee hinter der Modellierung des Hybrid-Falls. Durch eine Kombination der Zielfunktionsvektoren des vertikalen (magenta) und horizontalen (blau) Problems erhält man einen neuen Zielfunktionsvektor (grün), der für die Suche nach einer Hybrid-Tastenbelegung geeignet ist. Die farbigen Punkte sind dabei die Lösungen der jeweiligen Probleme.

Damit erhält man folgendes gemischt-ganzzahliges Optimierungsproblem:

$$\min \frac{B-1}{\sum_{i,k=1}^n \tilde{p}_{ik}} \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m \tilde{p}_{ik} \cdot (\alpha d_{jl} + \beta \hat{d}_{jl}) \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m p_i \cdot (\alpha d_j + \beta \tilde{d}_j) \cdot y_{ij}$$

u.d.N. $(y, x) \in \mathcal{F}_{P16}$ (P20)

Dabei sind $\alpha \geq 0$ und $\beta \geq 0$ Faktoren, die das Verhältnis zwischen vertikaler und horizontaler Bedienung angeben und es gilt $\alpha + \beta = 1$. Die Eigenschaften des Problems (P20) übertragen

sich von (P16).

Abbildung 39 veranschaulicht den Wechsel zwischen realistischer einhändiger und beidhändiger Bedienung, welcher aus einer Veränderung der Geräteposition resultiert.

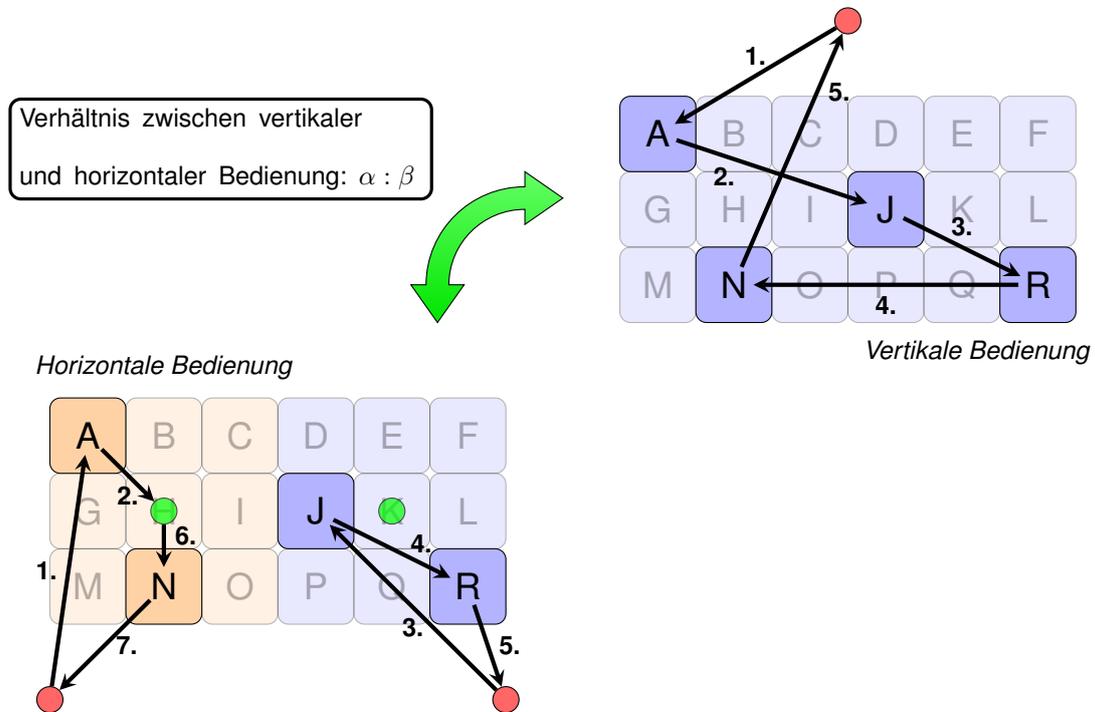


Abbildung 39: Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der realistischen Schreibweisen in vertikaler und horizontaler Geräteposition im Verhältnis $\alpha : \beta$.

5.2 Ergebnisse der Optimierung

Die folgenden Ergebnisse basieren auf der Annahme, dass ein Smartphone-Nutzer sein Gerät in gleichen Teilen einhändig und beidhändig bedient, es gilt also $\alpha = \beta = 0,5$. Eine Diskussion dieser Wahl und ihrer Auswirkungen findet in Abschnitt 9 statt.

Die Inputdaten, die den Ergebnissen zugrundeliegen finden sich in Anhang A und umfassen die Wahrscheinlichkeiten p_i und p_{ik} , sowie die Layout-abhängigen Distanzen d_j , \tilde{d}_j , \hat{d}_j und d_{jl} . Die Distanzen \hat{d}_{jl} werden analog zu Abschnitt 4.2.2 automatisch berechnet. Der Wert von \tilde{p} für den Ansatz THR (siehe Abschnitt 3.2.4) wird wie gewohnt gleich 0,6%, der fixierte Buchstaben und der Tastatur-Teilbereich für SYMM gleich e beziehungsweise gleich der rechten Hälfte H_R gewählt. Unter diesen Voraussetzungen erhält man folgende Tastenbelegungen.

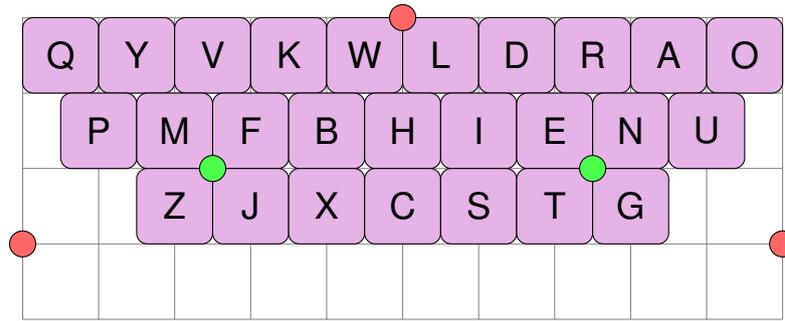


Abbildung 40: Distanz-optimierte Tastenbelegung für das QWERTY-like-Layout bei realistischer Schreibweise im Hybrid-Fall.

Obige Tastenbelegung für das QWERTY-like-Layout erhält man in 53,2s (Zielfunktionswert: 135,022 LE, zurückzulegende Distanz: 218,02 LE). Für das 3x9-Layout erhält man nach 74,9s folgende Belegungen mit Zielfunktionswert 133,09 LE und zurückzulegender Distanz 206,753 LE.

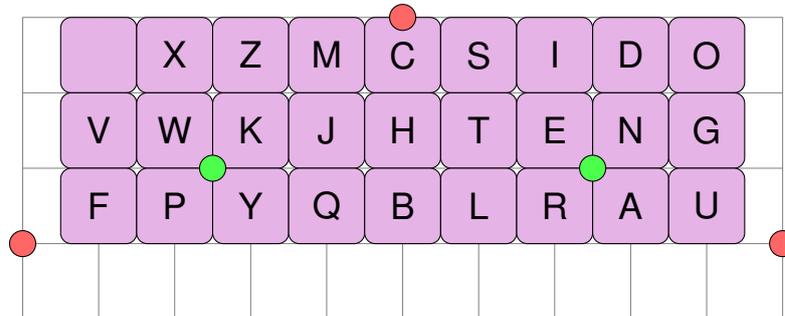


Abbildung 41: Distanz-optimierte Tastenbelegung für das 3x9-Layout bei realistischer Schreibweise im Hybrid-Fall.

Das Ergebnis der Optimierung für das 3x10-Layout (siehe Abb. 42) erhält man in 121,7s (Zielfunktionswert: 133,068 LE, Zurückzulegende Distanz: 211,635 LE).

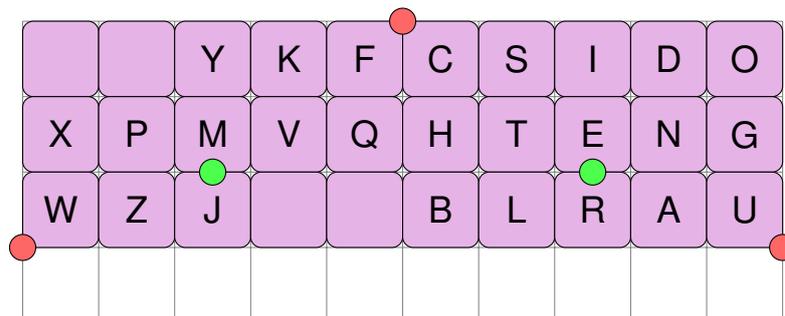


Abbildung 42: Distanz-optimierte Tastenbelegung für das 3x10-Layout bei realistischer Schreibweise im Hybrid-Fall.

Die Dauer der Berechnung der distanzoptimierten 4x8-Tastenbelegung aus Abbildung 43 mit

Zielfunktionswert 132,191 LE und zurückzulegender Distanz 197,567 LE beträgt 435,0 s.



Abbildung 43: Distanz-optimierte Tastenbelegung für das 4x8-Layout bei realistischer Schreibweise im Hybrid-Fall.

Eine Analyse der resultierenden Tastenbelegungen im Hybrid-Fall zeigt, dass die Gruppe der Buchstaben mit den häufigsten Bigrammen (D, E, I, N, R, S, T) diesmal im Gegensatz zum horizontalen Fall (siehe Abschnitt 4.2.2) auf einer Seite positioniert wurde. Dies erscheint insofern sinnvoll, als dass die Tastatur vertikal und horizontal verwendet wird, weshalb eine Platzierung auf beiden Seiten lange zurückzulegende Strecken bei vertikaler Nutzung erfordern würde. Dies ist auch der Grund, warum sämtliche Vokale auf der rechten Seite liegen und seltene Buchstaben auf der linken Seite. In der Praxis könnte diese Positionierung dazu führen, dass die beiden am Schreibvorgang beteiligten Finger nicht gleichmäßig belastet werden, was problematisch sein könnte (siehe auch Kapitel 10). Leere Tasten und seltene Buchstaben sind analog zu den Ergebnissen im vertikalen und horizontalen Fall weit entfernt von den Startpunkten platziert.

5.3 Zusammenfassung der Ergebnisse im Hybrid-Fall

Die nachfolgenden Ergebnisse basieren auf der Verwendung der realistischen Schreibweisen und resultieren aus der Wahl $\tilde{p} = 0,6\%$ für den Ansatz THR und $\tilde{i} = e, T = H_R$ für den Ansatz SYMM.

Tabelle 11 fasst die Eigenschaften und Resultate der Optimierung von Problem (P20) zusammen. Dabei ist sowohl der Zielfunktionswert (Spalte 5) als auch die zurückzulegende Gesamtdistanz (Spalte 6) in LE angegeben.

Tabelle 11: Ergebnisse der Distanzminimierung für Hybrid-Anwendungen.

Layout	NB	Var.	Laufzeit	Zfkts.wert	Distanz
QWERTY-like	25040	17514	53,2 s	135,022	218,02
3x9-Layout	27163	19020	74,9 s	133,09	206,753
3x10-Layout	32971	22950	121,7 s	133,068	211,635
4x8-Layout	37598	26112	435,0 s	132,191	197,567

Vergleicht man die Werte aus Tabelle 11 mit jenen aus den Tabellen 8 und 9, so stellt man fest, dass die Anzahl der Variablen und Nebenbedingungen für alle Gerätepositionen übereinstimmt. Dies macht insofern Sinn, als dass sich die zugehörigen Probleme nur in den Koeffizienten der Zielfunktionen unterscheiden.

Die Werte der zurückzulegenden Gesamtdistanz liegen in der Regel über denen im vertikalen und horizontalen Fall, was daher rührt, dass die bereits mehrfach erwähnten konträren Tastenbelegungs-Anforderungen der beiden regulären Positionen im Hybrid-Fall vereint werden müssen.

Unabhängig von den konkreten Werten stellt sich die Frage, ob durch die Modellierung in Abschnitt 5.1 die Eignung der resultierenden Tastenbelegungen für vertikale und horizontale Bedienung sichergestellt werden konnte. Tabelle 12 gibt Aufschluss über die Eigenschaften der hybrid-optimierten Tastenbelegungen in verschiedenen Gerätepositionen, wobei die Werte der Spalten 2,3 und 5 in LE angegeben sind.

Tabelle 12: Zurückzulegende Gesamtdistanz der hybrid-optimierten Tastenbelegungen in verschiedenen Gerätepositionen, sowie zusätzlicher Aufwand gegenüber den Optima der ursprünglichen Positionen.

Layout	Distanz (vert.)	Distanz (hor.)	zusätzl. Aufwand ggü. vert./hor. Optimum		Distanz (hybrid)
QWERTY-like (hybr.opt, Abb.40)	240,173	195,867	+9,0 %	+6,6 %	218,02
3x9-Layout (hybr.opt, Abb.41)	228,568	184,939	+15,0 %	+6,8 %	206,753
3x10-Layout (hybr.opt, Abb.42)	238,191	185,079	+19,6 %	+10,0 %	211,653
4x8-Layout (hybr.opt, Abb.43)	212,803	182,331	+14,5 %	+11,5 %	197,567

Der Vergleich der Informationen aus Tabelle 12 mit denen aus Tabelle 10 offenbart, dass die zurückzulegende Distanz auf hybrid-optimierten Belegungen deutlich geringer ist, als die mittlere zurückzulegende Distanz von Tastenbelegungen, die für eine Position optimiert wurde und gleichermaßen in der anderen Position genutzt werden. Damit geht eine Reduktion des durchschnittlichen zusätzlichen Aufwandes einher. Außerdem sind die Abweichungen der Distanzen vom Optimum der jeweiligen Position deutlich gleichmäßiger, also "fairer" verteilt, weshalb die Hybrid-Tastenbelegungen den ursprünglichen Wunsch Vorbehalte gegen eine Position zu verhindern erfüllen und in der Lage sind die Textübertragung auf Smartphones effizienter zu gestalten.

6 Minimierung der benötigten Zeit

Alternativ zur Minimierung des Schreibaufwandes in Form der zurückzulegenden Gesamtstrecke, sind Smartphone-Nutzer oftmals an einer Minimierung der Zeit interessiert, die für die Übertragung eines Textes benötigt wird. Um diesen Wunsch in einer Tastenbelegung umsetzen zu können, ist ein Modell erforderlich, mit dem sich die Bewegungsdauer zwischen zwei Punkten bestimmen lässt. Für diesen Zweck wird in Übereinstimmung mit verwandter Literatur das Gesetz von Fitt verwendet (siehe Abschnitt 2.4.5). Analog zu Kapitel 5 werden vereinfachte sowie semi-realistische Schreibweisen nicht betrachtet.

6.1 Zeitminimierung für vertikale, horizontale und Hybrid-Anwendungen

6.1.1 Modellierung

Analog zur Suche nach einer distanzminimalen Tastenbelegung (siehe Abschnitt 3.2.1) lässt sich die Suche nach einer zeitminimalen Belegung für realistische Schreibweisen als quadratisches Zuordnungsproblem modellieren. Der Unterschied zwischen beiden Fällen besteht lediglich in der Verwendung der Distanz beziehungsweise der Zeit als Koeffizienten der Variablen in der Zielfunktion. Wie in Abschnitt 3.2.2 lässt sich daraus ein ganzzahliges, lineares Optimierungsproblem ableiten, welches folgende Form hat.

$$\begin{aligned} \min (B - 1) \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m p_{ik} \cdot t_{jl} \cdot x_{ijkl} \\ \text{u.d.N. } (y, x) \in \mathcal{F}_{P_4} \end{aligned} \quad (\text{P21})$$

In (P21) bezeichnet $t_{jl} \geq 0$ die Zeit, die benötigt wird, um die Tasten $j \in [m]$ und $l \in [m]$ aufeinanderfolgend zu tippen. Für eine Tastengröße von 1 LE lässt sich diese dank (2.6) durch folgenden Ausdruck bestimmen.

$$t_{jl} = a + b \cdot \log_2(d_{jl} + c) \quad (6.1)$$

Das Problem (P21) ist aufgrund der langen Laufzeit für realistische Instanzen abermals ungeeignet (vgl. (P4) in Abschnitt 3.2.2). Möchte man die Verbesserungsansätze aus Abschnitt

3.2.3 darauf anwenden, so stellt sich die Frage, welche Auswirkungen die Veränderung der Koeffizienten auf die potentiellen Verbesserungen hat.

Die Eigenschaften und der Erfolg der Ansätze CUTS, SYMM, AJLIN, AJCON sind unabhängig von den Koeffizienten, weshalb sie unverändert verwendet werden können. Die Einführung eines neuen Kriteriums für die Zielfunktion mittels OBJFCT kann analog mithilfe der Bewegungsdauer t_j vom Startpunkt zur Taste j statt der Distanz d_j erfolgen. Es gilt analog zu (6.1):

$$t_j = a + b \cdot \log_2(d_j + c) \quad (6.2)$$

Die Wahl der Parameter in *Fitts' Law* (siehe Abschnitt 2.4.5) hat zur Folge, dass die Zeiten t_{jl} für alle Tastenpaare $(j, l) \in [m] \times [m]$ echt größer Null sind, da man selbst für das doppelte Antippen derselben Taste Zeit benötigt ($a = 0.083s$). Dies führt dazu, dass es im Gegensatz zu den Optimierungsproblemen aus den Kapiteln 3, 4 und 5 nur noch von den Wahrscheinlichkeiten $0 \leq p_{ik} \leq 1$ abhängt, ob die Koeffizienten gleich Null sind. Erinnern wir uns an die Aussage aus Lemma 3.30, welches sich übertragen lässt, dann erkennt man, dass die Zulässigkeitsbereiche der Optimierungsprobleme, welche aus (P21) und den Relaxationen UPBD und NOBIN entstehen, unter geeigneten Umständen ($p_{ik} \neq 0 \forall i, k$) gleich \mathcal{F}_{P21} sind. Die Konsequenz davon wäre die nicht explizit geforderte Ganzzahligkeit der Lösungen aufgrund von $y_{ij} \in \{0, 1\}$ und dem impliziten Zusammenhang $x_{ijkl} = y_{ij} \cdot y_{kl}$.

Allein die Verbesserungen aus Abschnitt 3.2.3 konnten im Fall der Distanzminimierung die Laufzeit nicht ausreichend senken, weshalb der Ansatz IMPLZERO und Methoden zur Reduktion der Einträge der Wahrscheinlichkeitsmatrix entwickelt wurden (bspw. THRESHOLD, siehe Abschnitt 3.2.4). Letztere können unverändert auch für zeitminimierende Optimierungsprobleme genutzt werden. Der Ansatz IMPLZERO muss insofern auf die Verwendung von t_{jl} statt d_{jl} angepasst werden, als dass die Variablen x_{ijkl} , für die $j = l$ gilt, nun nicht mehr allein aufgrund dieses Zusammenhangs nullgesetzt werden dürfen. Der Grund dafür ist oben erläuterte Eigenschaft, dass $t_{jl} > 0$. Damit ergibt sich eine neue Indexmenge I^1 , deren zugehörige x -Variablen nullgesetzt werden (vergleiche I^0 in Abschnitt 3.2.4).

$$I^1 := \{(i, j, k, l) \in I \mid (i = k \wedge j \neq l) \vee (i \neq k \wedge j = l) \vee p_{ik} = 0\} \quad (6.3)$$

Die Verwendung der Reduktionsmethode $\text{THR}(\tilde{p})$ gestattet folgenden Definition.

$$\tilde{I}^1 := \{(i, j, k, l) \in I \mid (i = k \wedge j \neq l) \vee (i \neq k \wedge j = l) \vee p_{ik} < \tilde{p}\} \quad (6.4)$$

Damit erhält man ein lineares Optimierungsproblem, das mit einer geeigneten Definition von \bar{t}_{jl} und \bar{t}_j (siehe 6.5 und 6.6) für die Berechnung von zeitminimierenden Tastenbelegungen für die realistischen Schreibweisen in verschiedenen Gerätepositionen geeignet ist.

$$\begin{aligned}
 \min \quad & \frac{B-1}{\sum_{i,k=1}^n \tilde{p}_{ik}} \cdot \sum_{i=1}^n \sum_{j=1}^m \sum_{k=1}^n \sum_{l=1}^m \tilde{p}_{ik} \cdot \bar{t}_{jl} \cdot x_{ijkl} + 2 \cdot \sum_{i=1}^n \sum_{j=1}^m \bar{t}_j \cdot p_i \cdot y_{ij} \\
 \text{u.d.N.} \quad & \begin{cases} \sum_{i=1}^n y_{ij} = 1 & \text{falls } m = n \\ \sum_{i=1}^n y_{ij} \leq 1 & \text{falls } m > n \end{cases} \quad \forall j \in [m] \\
 & \sum_{j=1}^m y_{ij} = 1 \quad \forall i \in [n], i \neq \tilde{i} \\
 & \sum_{j \in T} y_{ij} = 1 \\
 & x_{ijkl} = 0 \quad \forall i, j, k, l \in \tilde{I}^1 \\
 & y_{ij} + y_{kl} - 1 \leq x_{ijkl} \quad \forall i, j, k, l \in I \setminus \tilde{I}^1 \\
 & \sum_{i=1}^n x_{ijkl} \leq y_{kl} \quad \forall j, k, l \in I_{jkl} \\
 & \sum_{j=1}^m x_{ijkl} = y_{kl} \quad \forall i, k, l \in I_{ikl} \setminus \tilde{I}_{ikl}^1 \\
 & \sum_{j=1}^m x_{ijkl} \leq y_{kl} \quad \forall i, k, l \in \tilde{I}_{ikl}^1 \\
 & x_{ijkl} = x_{klij} \quad \forall i, j, k, l \in I \setminus \tilde{I}^1 \\
 & y_{ij} \in \{0, 1\} \quad \forall i \in [n], j \in [m] \\
 & x_{ijkl} \geq 0 \quad \forall i, k \in [n], j, l \in [m]
 \end{aligned} \tag{P22}$$

Analog zur Distanzminimierung unterscheiden sich die Positionen lediglich in den Inputdaten und \bar{t}_{jl} ist definiert als:

$$\bar{t}_{jl} := \begin{cases} t_{jl} & \text{im vertikalen Fall} \\ \hat{t}_{jl} := \begin{cases} t_{jl}, & \text{wenn gilt: } j, l \in T_L \vee j, l \in T_R \\ \hat{t}_j + \hat{t}_l, & \text{sonst} \end{cases} & \text{im horizontalen Fall} \\ \alpha t_{jl} + \beta \hat{t}_{jl} & \text{im Hybrid-Fall} \end{cases} \tag{6.5}$$

und \bar{t}_j als

$$\bar{t}_j := \begin{cases} t_j, & \text{im vertikalen Fall} \\ \tilde{t}_j, & \text{im horizontalen Fall} \\ \alpha t_j + \beta \tilde{t}_j, & \text{im Hybrid-Fall} \end{cases} \tag{6.6}$$

Dabei bezeichnen \tilde{t}_j und \hat{t}_j die Dauer einer Bewegung von der Taste j zum zugehörigen ho-

horizontalen Startpunkt beziehungsweise zum zugehörigen Wartepunkt. Sie lassen sich mithilfe von *Fitts' Law* (siehe (6.2)) aus den Distanzen \tilde{d}_j und \hat{d}_j bestimmen.

6.1.2 Ergebnisse der Optimierung

Die nachfolgenden Tastenbelegungen sind die Lösungen von (P22) für eine realistische Schreibweise in vertikaler Geräteposition. Sie basieren auf der Wahl $\tilde{i} = e$, $T = H_R$ und $\tilde{p} = 0,6\%$. Die Lösungen der Zeitminimierung für horizontale und Hybrid-Anwendungen werden erst später präsentiert, da sie zuvor in Abschnitt 6.2 mithilfe einer neuen Schreibweise verbessert werden können. Die Werte für \bar{t}_{jl} und \bar{t}_j werden von *VirtualKeyboardOptimization.mos* automatisch aus den eingelesenen Layout-spezifischen Distanzen (siehe Anhang A.2) und in Abhängigkeit der Geräteposition errechnet.

Das *QWERTY-like*-Layout wird in 158,3 s auf folgende Art und Weise mit Buchstaben belegt.

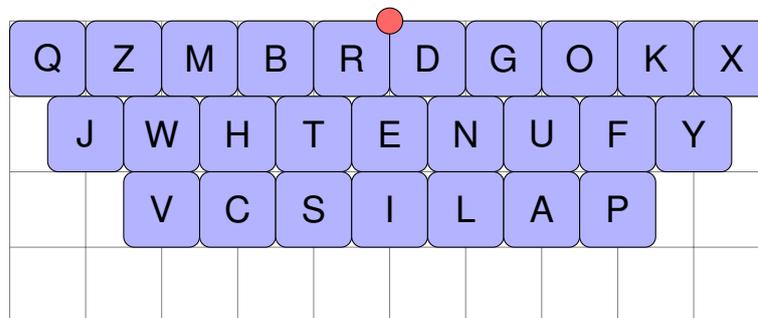


Abbildung 44: Zeit-optimierte Tastenbelegung für das *QWERTY-like*-Layout bei realistischer Schreibweise im vertikalen Fall.

Die benötigte Zeit für die Übertragung eines Textes der Länge B auf obiger Tastenbelegung beträgt 27,8433 s (Zielfunktionswert: 23,465 s).

Die optimierte Belegung des 3×9 -Layouts (siehe Abb. 45) erhält man in 139,9 s und führt zu einer benötigten Schreibzeit von 27,1755 s (Zielfunktionswert: 23,4319 s).

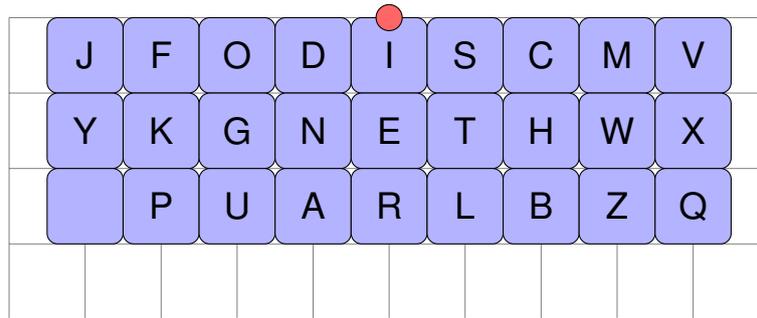


Abbildung 45: Zeit-optimierte Tastenbelegung für das 3x9-Layout bei realistischer Schreibweise im vertikalen Fall.

Das 3x10-Layout erhält dank der Optimierung in 153,0s die Tastenbelegung in Abbildung 46 mit Zielfunktionswert 23,4345 s. Die Schreibdauer darauf beträgt 27,0569 s.

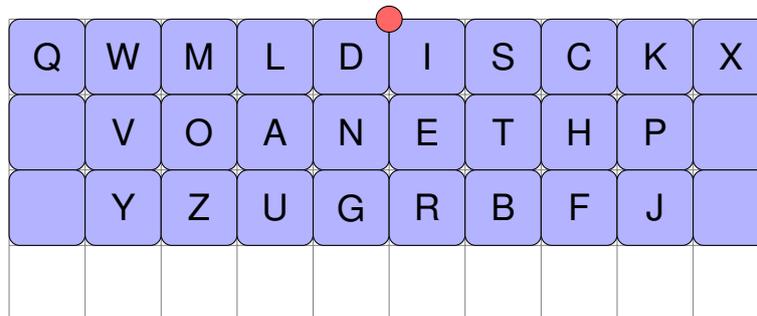


Abbildung 46: Zeit-optimierte Tastenbelegung für das 3x10-Layout bei realistischer Schreibweise im vertikalen Fall.

Abbildung 47 zeigt die Tastenbelegung des 4x8-Layouts (Laufzeit: 259,6s, Zielfunktionswert: 23,3447 s), auf welcher man 26,5841 s für die Textübertragung benötigt.

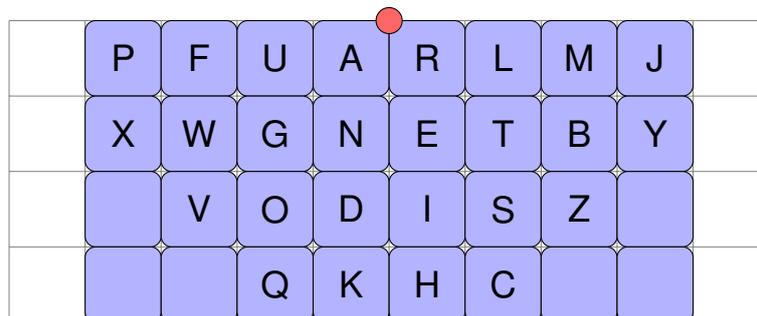


Abbildung 47: Zeit-optimierte Tastenbelegung für das 4x8-Layout bei realistischer Schreibweise im vertikalen Fall.

Vergleicht man die eben präsentierten zeit-optimierten, vertikalen Tastenbelegungen mit den

distanz-optimierten Ergebnissen aus Kapitel 3, so stellt man fest, dass sich die Belegungen kaum unterscheiden. Der Grund dafür ist, dass trotz des nichtlinearen Zusammenhangs aus *Fitts' Law* (siehe Abschnitt 2.4.5) eine kurze zurückzulegende Strecke auch eine geringe benötigte Zeit mit sich bringt. Damit gelten die in Abschnitt 3.3.3 beschriebenen Charakteristika bezüglich der Buchstabenverteilung auch für die zeit-optimierten Tastenbelegungen.

Der Vollständigkeit halber und um später die eingangs erwähnten Verbesserungen für die horizontale und die Hybrid-Position belegen zu können, zeigt Tabelle 13 die Eigenschaften und Ergebnisse der Zeitminimierung für realistische Schreibweisen in allen Gerätepositionen mithilfe von (P22) und $\tilde{i} = e, T = H_R, \tilde{p} = 0,6\%$. Die Werte der Spalten 6 und 7 sind dabei in Sekunden angegeben. Abermals stimmt die Anzahl der Nebenbedingungen und Variablen für alle Positionen überein und ist darüber hinaus identisch zu den Werten der Distanzminimierung.

Tabelle 13: Ergebnisse der Zeitminimierung für unterschiedliche Gerätepositionen bei Verwendung der realistischen Schreibweisen.

Layout	Position	NB	Var.	Laufzeit	Zfkts.wert	Schreibdauer
QWERTY-like	vertikal	25040	17514	158,3 s	23,465	27,8433
QWERTY-like	horizontal	25040	17514	254,1 s	23,7947	30,4198
QWERTY-like	hybrid	25040	17514	52,0 s	23,7821	29,6797
3x9-Layout	vertikal	27163	19020	139,9 s	23,4319	27,1755
3x9-Layout	horizontal	27163	19020	166,3 s	23,5551	29,3373
3x9-Layout	hybrid	27163	19020	118,0 s	23,5342	28,8278
3x10-Layout	vertikal	32971	22950	153,0 s	23,4345	27,0569
3x10-Layout	horizontal	32971	22950	123,7 s	23,5158	29,7443
3x10-Layout	hybrid	32971	22950	128,3 s	23,5328	29,2674
4x8-Layout	vertikal	37598	26112	259,6 s	23,3447	26,5841
4x8-Layout	horizontal	37598	26112	260,3 s	23,4664	28,9742
4x8-Layout	hybrid	37598	26112	332,4 s	23,4764	28,6867

Die Auswertung der Resultate zeigt, dass die Schreibdauer auf Belegungen, welche für den horizontalen Fall optimiert wurden, länger ist als die der vertikalen Belegungen. Auf den ersten Blick entspricht dieses Resultat nicht den Erwartungen, da man durch die Benutzung von zwei Fingern eine schnellere Eingabe prognostiziert hätte. Der Grund warum die Erwartungen in diesem Fall nicht erfüllt sind, liegt in der verwendeten Schreibweise, die darauf basiert, dass Bewegungen hintereinander ausgeführt werden. Im folgenden Abschnitt wird deshalb eine Schreibweise zugrundegelegt, welche parallele Bewegungen zulässt.

6.2 Zeitreduktion mithilfe der Hover-over-Schreibweise

Wie in Abschnitt 2.4.4 erläutert, unterscheiden sich die Schreibweisen im horizontalen Fall von denen im vertikalen Fall durch die Eingabe mit zwei Fingern statt mit einem. Dies führt unter

anderem dazu, dass sich der Finger, welcher im Moment keine Aufgabe zu erledigen hat, auf eine Warteposition zurückzieht (siehe Abschnitt 2.4.4 und 4.2). Es ist aber wahrscheinlich, dass dieser Finger nicht auf dem Wartepunkt positioniert bleibt, bis er an der Reihe ist, sondern sich ab einem gewissen Zeitpunkt bereits auf den Weg zur nächsten anzutippenden Taste macht. Für die zurückzulegende Distanz macht diese spezielle Schreibweise keinen Unterschied, aber sie eignet sich dafür die benötigte Zeit zu reduzieren. Der Name "Hover-over" resultiert aus der Tatsache, dass ein Finger in diesem Fall unter Umständen schon über der Taste schwebt, die er als nächstes antippen muss.

In der Annahme, dass ein Benutzer sich auf die aktuelle Texteingabe konzentrieren muss und deshalb nicht in der Lage ist vorausschauend zu schreiben, kann man davon ausgehen, dass sich ein wartender Finger frühestens dann in Bewegung setzt, wenn der letzte Buchstabe vor seinem Einsatz getippt wurde. Das bedeutet, dass ab dem Moment, ab dem sich der eine Finger auf den Rückweg zur Mitte macht, der andere Finger sich in Richtung des nächsten Buchstabens bewegt (siehe Abb. 48).

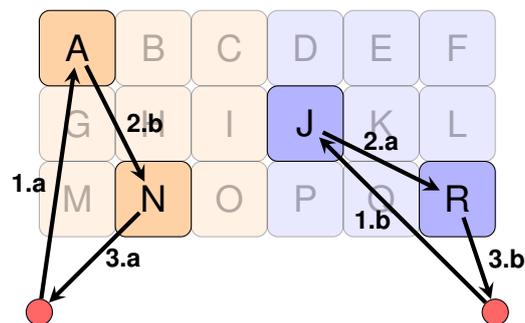


Abbildung 48: Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der Hover-over-Schreibweise in horizontaler Geräteposition. Dabei erfolgen die Bewegung 1.a und 1.b zur gleichen Zeit, jedoch muss der rechte Daumen mit der Eingabe von "J" warten, bis der linke Daumen "A" getippt hat. Im Anschluss daran folgen (gleichzeitig) die Bewegungen 2.a und 2.b.

Um die Formulierung (P22) zur Berechnung einer optimalen Tastenbelegung für den Hover-over-Ansatz verwenden zu können, genügt eine neue Definition von \bar{t}_{jl} . Da die Hover-over-Schreibweise nur dann möglich ist, wenn sich das Smartphone in horizontaler Lage befindet und zwei aufeinanderfolgende Buchstaben auf verschiedenen Tastaturseiten liegen, verändert sich die ursprüngliche Definition von \bar{t}_{jl} (siehe Definition (6.5)) vor allem für diesen speziellen Fall der Bedienung. Damit ändert sich auch die Definition im Hybrid-Fall, da dieser zum Teil auf der horizontalen Schreibweise basiert. Insgesamt gilt nun:

$$\bar{t}_{jl} := \begin{cases} t_{jl} & \text{im vertikalen Fall} \\ \hat{t}_{jl} := \begin{cases} t_{jl}, & \text{wenn gilt: } j, l \in T_L \vee j, l \in T_R \\ \max\{\hat{t}_j, \hat{t}_l\}, & \text{sonst} \end{cases} & \text{im horizontalen Fall} \\ \alpha t_{jl} + \beta \hat{t}_{jl} & \text{im Hybrid-Fall} \end{cases} \quad (6.7)$$

Betrachtet man die Unterschiede zwischen den Definitionen (6.5) und (6.7) im horizontalen Fall, so stellt man fest, dass für den Tippvorgang zweier Tasten auf unterschiedlichen Seiten der Term $\hat{t}_j + \hat{t}_l$ durch $\max\{\hat{t}_j, \hat{t}_l\}$ ersetzt wurde. Während Ersterer die Dauer der Bewegung eines Fingers zum zugehörigen Wartepunkt und des anderen Fingers vom Wartepunkt zum jeweiligen Ziel modelliert, kann $\max\{\hat{t}_j, \hat{t}_l\}$ anschaulich als die gleichzeitige Bewegung beider Finger samt Warten desjenigen, welcher früher sein Ziel erreicht, interpretiert werden.

6.2.1 Ergebnisse im horizontalen Fall

Analog zu den Lösungen der Zeitminimierung im vertikalen Fall (siehe Abschnitt 6.1.2) basieren folgende horizontale Ergebnisse auf Problem (P22) und der Wahl $\tilde{i} = e, T = H_R$ und $\tilde{p} = 0,6\%$, wobei für \bar{t}_{jl} Definition (6.7) verwendet wird. Damit sind nachfolgend präsentierte Tastenbelegung für die Hover-over-Schreibweise optimiert.

Die Berechnung der Koeffizienten \bar{t}_{jl} erfolgt wie gewohnt automatisch mithilfe von *VirtualKeyboardOptimization.mos*, weshalb sich die Inputdaten auf die Wahrscheinlichkeiten aus Tabelle 24 und 25, sowie auf die Distanzen aus Anhang A.2 beschränken.

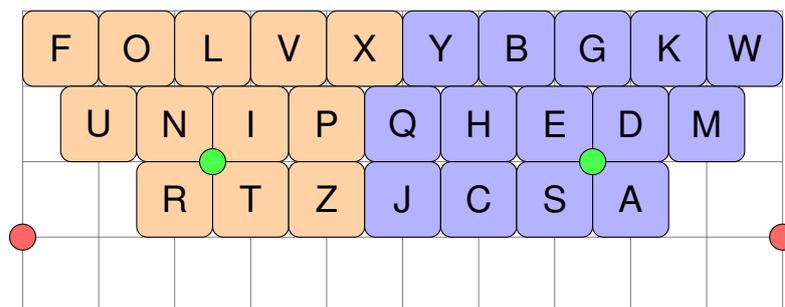


Abbildung 49: Zeit-optimierte Tastenbelegung für das QWERTY-like-Layout und die Hover-over-Schreibweise im horizontalen Fall.

Das QWERTY-like-Layout erhält durch die Optimierung (Laufzeit: 160,3 s) die Tastenbelegung aus Abbildung 49 mit Zielfunktionswert 20,511 s und Schreibdauer 23,4275 s.

Folgende Tastenbelegung ist die Lösung für das 3x9-Layout mit einer benötigten Textübertragungszeit von 22,8984 s (Laufzeit: 208,4 s, Zielfunktionswert: 20,5061 s).

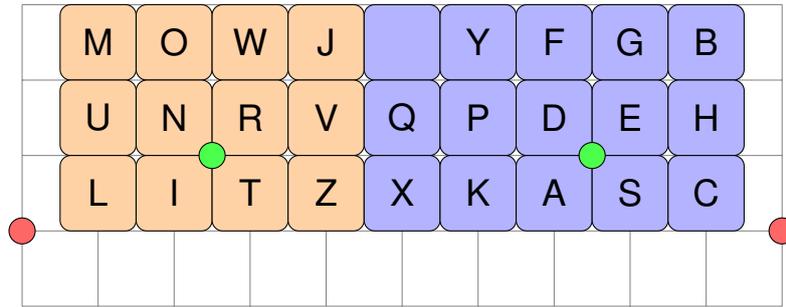


Abbildung 50: Zeit-optimierte Tastenbelegung für das 3x9-Layout und die Hover-over-Schreibweise im horizontalen Fall.

In 61,72 s lässt sich die Tastenbelegung für das 3x10-Layout errechnen (siehe Abb. 51). Sie besitzt einen Zielfunktionswert von 20,5698 s und führt zu einer Schreibdauer von 23,0816 s.

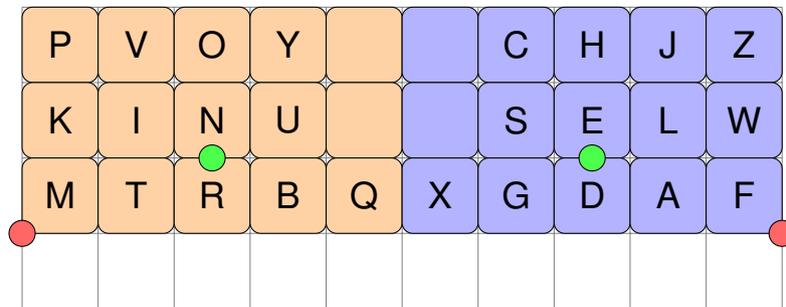


Abbildung 51: Zeit-optimierte Tastenbelegung für das 3x10-Layout und die Hover-over-Schreibweise im horizontalen Fall.

Die Optimierung liefert in 118,3 s eine Belegung für das 4x8-Layout, die einen Zielfunktionswert von 20,5678 s hat und eine Schreibzeit von 22,7836 s nach sich zieht (siehe Abb. 52).

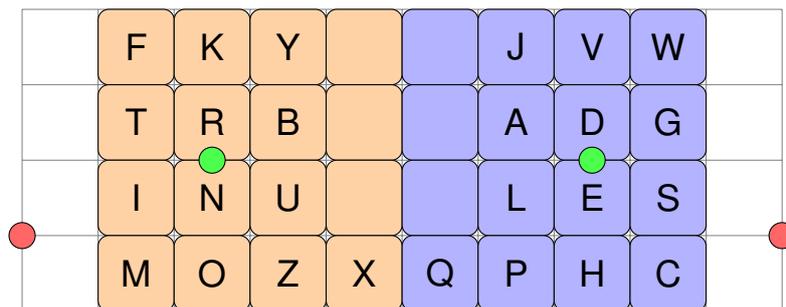


Abbildung 52: Zeit-optimierte Tastenbelegung für das 4x8-Layout und die Hover-over-Schreibweise im horizontalen Fall.

Während die Buchstaben bei realistischer Schreibweise im horizontalen Fall (siehe Abschnitt 4.2.2) so verteilt werden, dass die Buchstaben der häufigsten Bigramme (EN, ER) direkt nebeneinanderliegen (siehe bspw. Abb. 34), werden diese bei Verwendung der Hover-over-Schreib-

weise auf unterschiedliche Seiten aufgeteilt (siehe bspw. Abb. 49). Die Positionierung erfolgt wie gewohnt nahe des Wartepunktes. Der Grund für die Aufteilung häufiger Bigramme ist, dass sich so die dafür benötigte Eingabezeit reduzieren lässt. Während die Bewegung zwischen benachbarten Tasten mindestens die Zeit benötigt, die es dauert 1 LE zurückzulegen, erfordern zweiseitige Bigramme bei Hover-over-Bedienung nur einmal die Zeit von der weiter entfernten zugehörigen Taste zum Wartepunkt. Diese ist im besten Fall nur so groß wie die Bewegungsdauer der minimalen Entfernung einer Taste zum Startpunkt (0,5 LE).

Die übrigen Buchstaben, deren Bigrammwahrscheinlichkeiten nicht in die Optimierung einfließen, werden gemäß ihrer Wahrscheinlichkeiten mit Bezug zum Startpunkt verteilt.

Tabelle 14 fasst die Ergebnisse der Hover-over-Schreibweise im horizontalen Fall zusammen.

Tabelle 14: Ergebnisse der Zeitminimierung für die Hover-over-Schreibweise im horizontalen Fall.

Layout	NB	Var.	Laufzeit	Zfks.wert (s)	Schreibdauer (s)
<i>QWERTY-like</i>	25040	17514	160,3 s	20,511	23,4275
3x9-Layout	27163	19020	208,4 s	20,5061	22,8984
3x10-Layout	32971	22950	61,7 s	20,5698	23,0816
4x8-Layout	37598	26112	118,3 s	20,5678	22,7836

Wie man anhand der Schreibdauer für verschiedene Layouts in Tabelle 14 sehen kann, hat sich die für die Textübertragung benötigte Zeit in horizontaler Geräteposition beim Wechsel von der realistischen Schreibweise (siehe Definition (6.5) und Tabelle 13) auf die Hover-over-Schreibweise (siehe Definition (6.7)) deutlich reduziert (um durchschnittlich 22%). Dies entspricht den Erwartungen, die der Modellierung der Hover-over-Bedienung zugrundeliegen.

6.2.2 Ergebnisse in Hybridanwendungen

Da die Hybrid-Position zu einem gewissen Teil auf der horizontalen Geräteposition basiert, deren zugehörige Bedienung durch die Hover-over-Schreibweise verbessert wurde, werden in diesem Abschnitt die Auswirkungen auf die Ergebnisse des Hybrid-Falls betrachtet. Um die Vergleichbarkeit zu bisherigen Ergebnissen zu gewährleisten werden die Parameter von SYMM und THR identisch zu Abschnitt 6.2.1 gewählt. α und β seien analog zu Abschnitt 5.2 gleich 0,5.

Die Suche nach einer optimalen Tastenbelegung für die Hybrid-Position bei Verwendung der Hover-over-Schreibweise liefert in 157,1 s folgende Lösung (Zielfunktionswert: 23,5829 s) für das *QWERTY-like*-Layout. Auf ihr beträgt die Dauer der Textübertragung 27,2262 s.

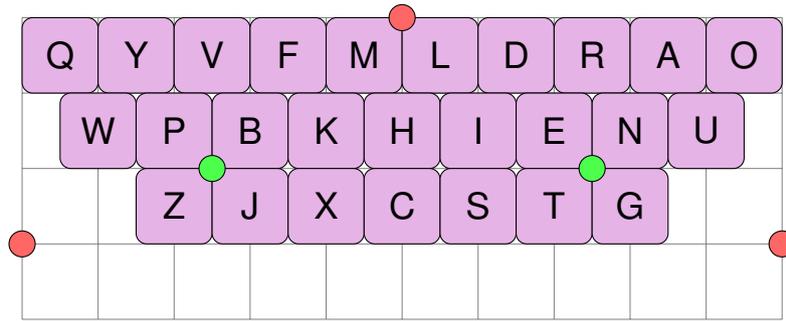


Abbildung 53: Zeit-optimierte Tastenbelegung für das QWERTY-like-Layout und die Hover-over-Schreibweise im Hybrid-Fall.

In 424,4 s erhält man die hybrid-optimierte Tastenbelegung für das 3x9-Layout (siehe Abb. 54). Der Zielfunktionswert ist dabei gleich 23,4922 s und die Schreibdauer gleich 26,7147 s.

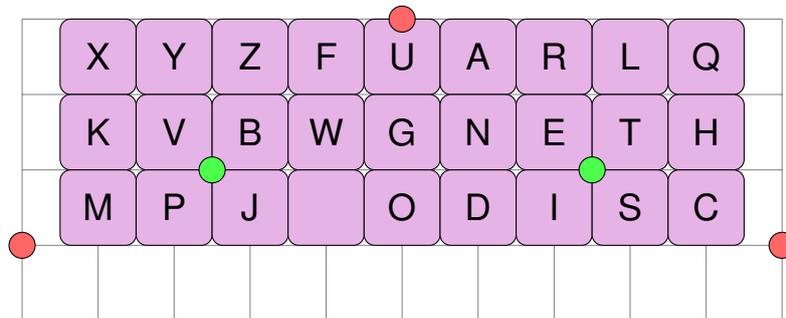


Abbildung 54: Zeit-optimierte Tastenbelegung für das 3x9-Layout und die Hover-over-Schreibweise im Hybrid-Fall.

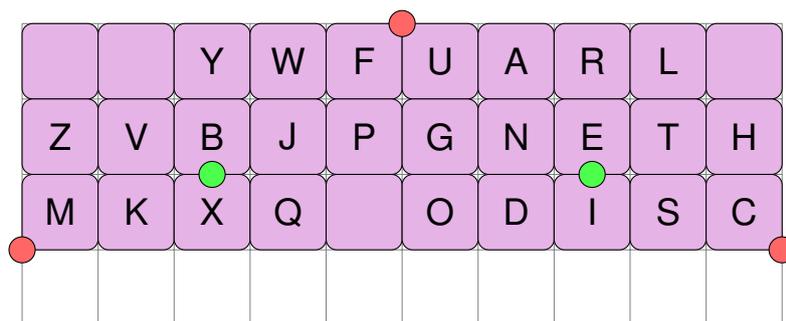


Abbildung 55: Zeit-optimierte Tastenbelegung für das 3x10-Layout und die Hover-over-Schreibweise im Hybrid-Fall.

Die 3x10-Belegung aus Abbildung 55 mit Zielfunktionswert 23,501 s lässt sich in 257,6 s errechnen und provoziert eine Schreibdauer von 27,039 s.

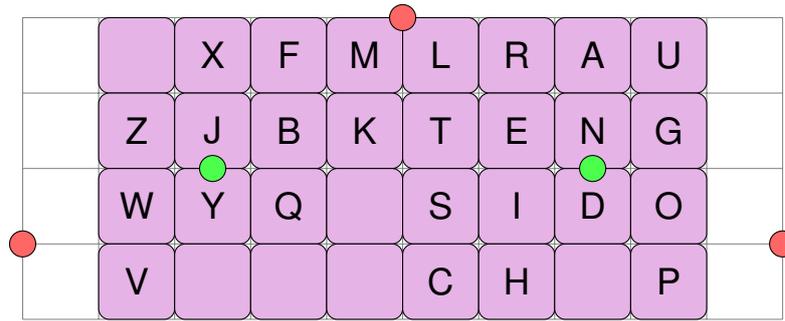


Abbildung 56: Zeit-optimierte Tastenbelegung für das 4x8-Layout und die Hover-over-Schreibweise im Hybrid-Fall.

Im Vergleich dazu ist die Schreibdauer auf der Tastenbelegung des 4x8-Layouts (siehe Abb. 56) mit 26,6507 s etwas geringer, jedoch dauert ihre Berechnung mit 515,5 s auch deutlich länger (Zielfunktionswert: 23,4388 s).

Analog zur horizontalen Geräteposition ließ sich auch im Hybrid-Fall die benötigte Zeit durch die Verwendung der Hover-over-Schreibweise reduzieren. Allerdings beträgt die durchschnittliche Reduktion in diesem Fall nur 7,6 % und damit deutlich weniger als in Abschnitt 6.2.1. Der Grund dafür ist, dass die Hover-over-Schreibweise nur in horizontaler Position verwendet werden kann und diese nur einen Teil des Hybrid-Falls (genauer gesagt $\beta \cdot 100\%$) ausmacht. Damit ist der in Abschnitt 6.2.1 beschriebene Wunsch häufige Bigramme aufzuteilen weniger stark ausgeprägt und die grundsätzliche Verteilung der Buchstaben auf die Seiten der Layouts sowie die relative Lage der Buchstaben untereinander hat sich im Vergleich zu den distanz-optimierten Hybrid-Belegungen aus Abschnitt 5.2 kaum verändert (vergleiche bspw. Abb. 40 und Abb. 53). Lediglich das Bigramm *BE* wird neuerdings auf verschiedenen Seiten positioniert (siehe bspw. Abb. 55), was die Änderungen der absoluten Buchstabenpositionen erklärt. Der Grund dafür ist, dass *B* nur zusammen mit *E* häufige Bigramme bildet und somit nur an einer kurzen Distanz zu *E* nicht aber an einer Positionierung zusammen mit anderen häufigen Buchstaben (wie bspw. *D*) auf einer Seite interessiert ist.

Tabelle 15 dient als Zusammenfassung der Ergebnisse der Hover-over-Schreibweise im Hybrid-Fall. Die Anzahl der Nebenbedingungen und Variablen der zugehörigen Optimierungsprobleme stimmen mit denen zuvor präsentierter Probleme überein.

Tabelle 15: Ergebnisse der Zeitminimierung für die Hover-over-Schreibweise im Hybrid-Fall.

Layout	NB	Var.	Laufzeit	Zfkts.wert (s)	Schreibdauer (s)
QWERTY-like	25040	17514	157,1 s	23,5829	27,2262
3x9-Layout	27163	19020	424,4 s	23,4922	26,7147
3x10-Layout	32971	22950	257,6 s	23,501	27,039
4x8-Layout	37598	26112	515,5 s	23,4388	26,6507

Teil III

Implementierung

7 Implementierung

Die Implementierung der vorgestellten Optimierungsprobleme wurde in *Xpress IVE* vorgenommen. Das daraus resultierende Tool *VirtualKeyboardOptimization.mos* liefert die in der Arbeit präsentierten Ergebnisse. Um diese reproduzieren zu können und zum tieferen Verständnis werden im Folgenden der Funktionsumfang von *VirtualKeyboardOptimization.mos* und die darin wählbaren Optionen näher erläutert.

7.1 Aufbau der Inputdaten

Um eine fehlerfreie Funktionsweise von *VirtualKeyboardOptimization.mos* sicherzustellen und die gewünschten Ergebnisse zu erhalten, sind die Inputdaten einer zu optimierenden Instanz in englischsprachiger Standardformatierung als **.dat*-Datei an das Tool zu übergeben. Diese enthält folgende Daten:

'**Keyboard**' (*string*): Der Name des Tastaturlayouts

'**Shape**' (*string*): Die Tastenanordnung des zugrundeliegenden Layouts, also ob die Tastatur eine rechteckige oder unregelmäßige Form hat (*rectangular* oder *free*)

'**Rows**' (*integer*): Die Anzahl der Zeilen des zugrundeliegenden Tastaturlayouts

'**Columns_free**' (*array(1..Rows) of integer*): Die Anzahl der Tasten jeder Zeile im Fall einer unregelmäßigen Layoutform

'**Columns_rect**' (*integer*): Die Anzahl der Spalten im Fall einer rechteckigen Layoutform

'**numberofkeys**' (*integer*): Die Anzahl der verfügbaren Tasten

'**numberofletters**' (*integer*): Die Anzahl der zu verteilenden Zeichen

'**Keyname**' (*array(1..numberofkeys) of string*): Die Namen der Tasten, welche zur Verfügung stehen

'**Lettername**' (*array(1..numberofletters) of string*): Die Namen der Zeichen, welche verteilt werden sollen

'**dist_keys_orig**' (*array(1..numberofkeys, 1..numberofkeys) of string*): Die Distanzen zwischen den einzelnen Tasten

'**dist_start_vert_orig**' (*array(1..numberofkeys) of real*): Die Distanzen d_j der Tasten zum vertikalen Startpunkt

'**dist_start_hor_orig**' (*array(1..numberofkeys) of real*): Die Distanzen \tilde{d}_j der Tas-

ten zum jeweils zugehörigen horizontalen Startpunkt

'dist_mid_hor_orig' (array(1..numberofkeys) of real): Die Distanzen \hat{d}_j der Tasten zum jeweils zugehörigen Wartepunkt

'letter_prob_orig' (array(1..numberofletters) of real): Die Buchstabenwahrscheinlichkeiten p_i

'bigram_prob_orig' (array(1..numberofletters,1..numberofletters) of real): Die Bigrammwahrscheinlichkeiten p_{ik}

'assignment_letters_keys' (array(1..numberofletters) of integer): Eine optionale Möglichkeit eigene Tastenbelegungen auswerten zu lassen, indem man jedem Buchstaben eine der verfügbaren Tasten zuordnet

7.2 Schematischer Aufbau des Tools *VirtualKeyboardOptimization.mos*

Algorithmus 1 *VirtualKeyboardOptimization.mos*

Eingabe: Inputdaten der in Abschnitt 7.1 beschriebenen Form

1. Wähle Parameter und lege Optionen fest
 2. Deklariere die Variablen und lies die vorhandenen Daten ein
 3. Initialisiere Funktionen und Prozeduren
 4. Initialisiere (Hilfs-)Variablen
 5. Prüfe die Inputdaten und die gewählten Optionen auf Richtigkeit
- if** Aufgabe des Tools ist die Optimierung einer Tastenbelegung **then**
- if** Kein Fehler in den Daten/Optionen **then**
- 6.1.1 Erzeuge die für die Optimierung benötigten Daten aus den eingelesenen Daten (in Abhängigkeit der Schreibweise und Position des Smartphones)
 - 6.1.2 Optionale Ausgabe der Einstellungen
 - 6.1.3 Optionale Ausgabe der Eigenschaften der Daten
 - 6.1.4 Erzeuge die Nebenbedingungen
 - 6.1.5 Optionale Anzeige der Modellgröße
 - 6.1.6 Erzeuge die Zielfunktion
 - 6.1.7 Optimierte die Zielfunktion
 - 6.1.8 Optionale Verwendung einer Heuristik
 - 6.1.9 Ausgabe der resultierenden Tastenbelegung
 - 6.1.10 Bewertung der resultierenden Tastenbelegung
 - 6.1.11 Optionale Bewertung einer selbst erstellten Tastenbelegung
- end if**
- else if** Aufgabe des Tools ist die Auswertung einer Tastenbelegung **then**
- if** Kein Fehler in den Daten/Optionen **then**
- 6.2.1 Erzeuge die für die Evaluation benötigten Daten (in Abhängigkeit der Schreibweise und Position des Smartphones)
 - 6.2.2 Ausgabe der eingelesenen Tastenbelegung
 - 6.2.3 Bewertung der eingelesenen Tastenbelegung
 - 6.2.4 Optionale Verwendung einer Heuristik
- end if**
- end if**
7. Definiere Funktionen und Prozeduren
-

7.3 Funktionen von *VirtualKeyboardOptimization.mos*

Das parallel zur Erforschung der Theorie der Optimierung von Tastenbelegungen (siehe Kapitel 3 - 6) entstandene Programm *VirtualKeyboardOptimization.mos* verfügt über eine Vielzahl von Funktionen und Einstellungsmöglichkeiten. Diese erlauben es dem Benutzer Tastenbelegungen hinsichtlich seiner individuellen Gegebenheiten und Vorstellungen zu optimieren. Eine

Auswahl davon sei im Folgenden genannt.

- **Optimierung oder Evaluation** Der Benutzer kann entscheiden, ob er die Tastenbelegung eines Layouts optimieren oder lediglich eine selbst erstellte Tastatur auswerten möchte.
- **verschiedene Zielfunktionen** Die Tastenbelegung eines eingelesenen Layouts kann hinsichtlich der erwarteten zurückzulegenden Distanz beziehungsweise der erwarteten benötigten Zeit optimiert werden (siehe Kapitel 3 - 6). Die notwendigen Daten berechnet das Tool automatisch aus den eingelesenen Werten (siehe Abschnitt 7.1).
- **verschiedene Gerätepositionen** Tastenbelegungen werden abhängig von der gewählten Geräteposition optimiert. Dabei steht die vertikale (siehe Kapitel 3), horizontale (siehe Kapitel 4) und hybrid-Position (siehe Kapitel 5) zur Auswahl.
- **verschiedene Schreibweisen** Die Schreibweise des Benutzers (vereinfacht (Abschnitt 3.1 und 4.1), semirealistisch (Abschnitt 4.2.1), realistisch (Abschnitt 4.2.2), Hover-over (Abschnitt 6.2)) kann frei gewählt werden. Die benötigten Daten werden auch in diesem Fall automatisch berechnet und in der Optimierung verwendet.
- **Kompatibilität zu unterschiedlichen Layouts** Das Tool ist nicht beschränkt auf die in der Arbeit verwendeten Tastaturformen, sondern optimiert die Belegungen beliebiger Layouts (einzulesen in der Form aus Abschnitt 7.1). Analog dazu ist das Tool nicht an die Wahl der speziellen Punkte aus Abschnitt 2.4.4 gebunden.
- **Grafische Ausgabe von Tastenbelegungen** Um sich eine Tastenbelegung nicht mühsam aus einem 0-1-Vektor konstruieren zu müssen, ist das Tool in der Lage Tastenbelegungen grafisch darzustellen.
- **Textlänge** Die Länge eines zu übertragenden Textes für den die Tastenbelegung optimiert wird kann frei gewählt werden.
- **Modell der Bewegungsdauer** Die Bewegungsdauer wird mithilfe von *Fitts' Law* bestimmt (siehe Abschnitt 2.4.5). Aufgrund der uneindeutigen Wahl der Parameter in verfügbarer Literatur (siehe Abschnitt 2.2) existiert innerhalb des Tools die Möglichkeit die Parameter frei zu wählen, sodass die Vergleichbarkeit zwischen eigenen Ergebnissen und Resultaten aus der Literatur sichergestellt ist.
- **Verhältnis im Hybrid-Fall** Das Verhältnis zwischen vertikaler und horizontaler Schreibweise im Hybrid-Fall kann beliebig gewählt werden (siehe α, β in Kapitel 5).
- **Reduktionsmethoden** Es stehen verschiedenste Methoden zur Auswahl, welche die Einträge der Bigramm-Wahrscheinlichkeitsmatrix reduzieren (siehe Abschnitt 3.2.4).

- **Ansätze zur Verbesserung der Lösbarkeit** Alle in Abschnitt 3.2.3 und 3.2.4 präsentierten Ansätze zur Verbesserung der Lösbarkeit stehen in vollem Umfang zur Verfügung und können in verschiedenen Kombinationen vom Benutzer getestet werden.
- **Grad der Ausgabe** Je nach Informationsbedürfnis des Benutzers lässt sich der Umfang der ausgegebenen Informationen anpassen.
- **Heuristik** Das Tool enthält das *2-opt*-Verfahren, welches als Heuristik für die weitere Verbesserung der aus der Optimierung resultierenden Tastenbelegungen dient. Dabei kann eine beliebige Anzahl an Iterationen gewählt werden.
- **Tastaturtest in verschiedenen Situationen** Das Tool ermöglicht die Auswertung einer optimierten Tastenbelegung in verschiedenen Situationen, das heißt Tests für alle Schreibweisen und Gerätepositionen, für die sie eigentlich nicht erstellt wurde. Analog dazu kann eine eingelesene Tastatur für verschiedene Szenarien getestet werden, beispielsweise um herauszufinden, für welches Setting sie am besten geeignet ist.
- **Bestimmung der Laufzeit** Das Tool liefert automatisch neben der Gesamtlaufzeit die Dauer der Optimierung, der Heuristik und der Evaluationen.
- **Fehlerprüfung** Das Tool überprüft eigenständig, ob die eingelesenen Daten sinnvoll sind und ob die gewählten Einstellungen zulässig sind. Sollte das nicht der Fall sein, erkennt es an welcher Stelle die Problematik liegt und liefert Korrekturhinweise.
- **Auslesen der Belegung** Auf Wunsch gibt das Tool die Tastenbelegung in Form eines Vektors an, um beispielsweise die Evaluation zu einem späteren Zeitpunkt unabhängig von der Optimierung vornehmen zu können.

Neben den wichtigsten Funktionen, welche soeben vorgestellt wurden, existieren zahlreiche weitere Features, deren detaillierte Erklärung an dieser Stelle zu umfangreich wäre. Der Leser ist deshalb herzlich eingeladen eigene Erfahrung mit dem Tool zu sammeln oder einen Blick in die Kommentare des Tools (siehe Quellcode in Anhang B) zu werfen, wo sich weitere Erklärungen finden.

Teil IV

Auswertung und Diskussion der Ergebnisse

8 Auswertung und Vergleich der Ergebnisse

Neben der zurückzulegenden Distanz und der benötigten Zeit für die Eingabe eines Standardtextes der Länge B gibt es ein weiteres Maß, das einem Vergleich zwischen verschiedenen Tastaturlayouts und -belegungen zugrunde gelegt werden kann.

8.1 Theoretische Bestimmung der Schreibgeschwindigkeit

Oftmals wird in der Literatur die auf einer Tastatur oder mithilfe einer Tastenbelegung erzielbare Schreibgeschwindigkeit präsentiert. Sie wird in Wörtern pro Minute (wpm) angegeben und lässt sich allgemein wie folgt berechnen:

$$wpm = \frac{60}{5t} \quad (8.1)$$

Dabei gibt t die durchschnittlich benötigte Zeit in Sekunden an, die für die Eingabe eines Buchstabens benötigt wird. Die Formel zur Berechnung des wpm -Wertes einer Tastenbelegung basiert dabei auf der allgemeinen Konvention, dass ein Wort durchschnittlich aus 5 Buchstaben besteht (siehe bspw. [MacKenzie u. Soukoreff, 2002b]). Außerdem fließt die benötigte Zeit für den Hin- und Rückweg der Finger zum jeweiligen Startpunkt nicht in die Berechnung der Schreibgeschwindigkeit ein. Damit lässt sich die durchschnittliche Eingabedauer t eines Buchstabens bestimmen durch:

$$t = \sum_{i=1}^n \sum_{k=1}^n p_{ik} \cdot \bar{t}_{T(i)T(k)} \quad (8.2)$$

Dabei bezeichnen $T(i)$ und $T(k)$ die Tasten, auf denen die Buchstaben i und k liegen und $\bar{t}_{T(i)T(k)}$ die Dauer einer Bewegung zwischen diesen Tasten. Letztere ist abhängig von der Position des Gerätes und der Schreibweise des Benutzers.

Da man an der Leistungsfähigkeit der Tastenbelegungen für möglichst realitätsnahe Schreibweisen interessiert ist, ist $\bar{t}_{T(i)T(k)}$ definiert wie in (6.7). Die Geschwindigkeiten, welche auf den in der Arbeit erzeugten Tastenbelegungen erzielt werden können, finden sich in den Tabellen 16, 18 und 20 in Abschnitt 8.3.

8.2 Alternative Tastaturen und Tastenbelegungen

Aufgrund der Vielzahl von Tastenbelegungen auf unterschiedlichsten Layouts beschränken wir uns auf eine Auswahl der Bekanntesten und Vielversprechendsten, um diese mit den Ergebnissen aus den vorangegangenen Kapiteln zu vergleichen. Dabei wird auf die Auswertung von unregelmäßigen Tastaturen (siehe bspw. *Metropolis* oder *Hooke's Keyboard* aus [Zhai et al., 2000]) oder solchen, die aufgrund ihrer Proportionen für den Einsatz auf Smartphones ungeeignet sind (siehe bspw. Tastaturform III aus [Li et al., 2006]), verzichtet.

8.2.1 Belegungen des *QWERTY-like*-Layouts

Die Abbildungen 57 - 58 zeigen verschiedene Tastenbelegungen für das *QWERTY-like*-Layout. Sie reichen von einer alphabetischen Anordnung der Buchstaben (siehe Abb. 58) über die Standard-*QWERTY*-Belegung (siehe Abb. 57), welche im 19. Jahrhundert von Christopher L. Sholes entwickelt wurde (siehe Abschnitt 1.2), bis hin zu weniger bekannten Tastenbelegungen.

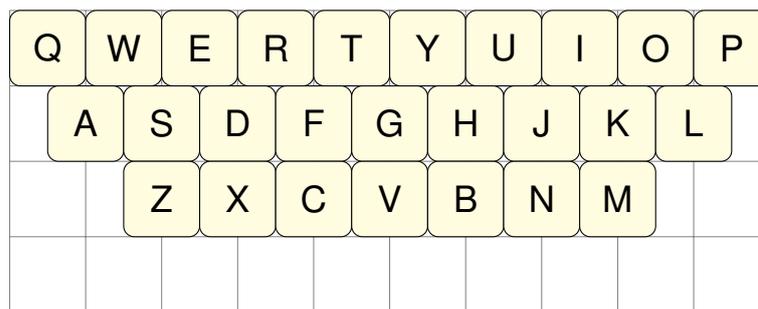


Abbildung 57: Standard-*QWERTY*-Tastatur (siehe bspw. [Lewis et al., 1999b])



Abbildung 58: Alphabetische Belegung des *QWERTY-like*-Layouts

Abbildung 59 zeigt die aktuellste der ausgewählten Belegungen für das *QWERTY-like*-Layout. Sie wurde von Li et al. [2006] mithilfe einer *simulated-annealing*-Heuristik für die Eingabe von englischem Text mit einem Finger erzeugt und ermöglicht laut den Autoren eine 30% schnellere Texteingabe als *QWERTY*.



Abbildung 59: [Li et al., 2006]-Tastenbelegung für das QWERTY-like-Layout

Die Tastenbelegung aus Abbildung 60, welche in [Lesher u. Moulton, 2000] präsentiert wird, entstand durch die Verwendung einer *2-opt*-Heuristik und reduziert den einhändigen Schreibaufwand um etwa 36 % gegenüber der *QWERTY*-Belegung.



Abbildung 60: [Lesher u. Moulton, 2000]-Tastenbelegung für das QWERTY-like-Layout

Im Mittelpunkt der Arbeit von Light u. Anderson [1993] steht die Textübertragung mit 10-Fingern, für welche die Autoren mithilfe eines *simulated-annealing*-Ansatzes eine 8 % schnellere Belegung als *QWERTY* entwickeln (siehe Abb. 61).



Abbildung 61: [Light u. Anderson, 1993]-Tastenbelegung für das QWERTY-like-Layout

8.2.2 DVORAK-Tastatur

Abbildung 62 zeigt die *DVORAK*-Tastatur. Sie besitzt zeilenweise 7,10,9 Tasten und verfügt über die Belegung, welche 1943 von August Dvorak als Konkurrent zur Standard-*QWERTY*-Tastatur vorgestellt wurde (siehe Abschnitt 2.2). Laut Buzing [2003] beträgt der Unterschied in der Schreibgeschwindigkeit bei 10-Finger-Bedienung etwa 5% zugunsten der *DVORAK*-Tastatur.



Abbildung 62: *DVORAK*-Tastatur (siehe bspw. [Lewis et al., 1999b])

8.2.3 KALQ-Tastatur

Die *KALQ*-Tastatur (siehe Abb. 63) ist einer der neuesten Versuche die *QWERTY*-Belegung zu reformieren. Sie wurde von Oulasvirta et al. [2013] mithilfe einer Kombination aus *gradient-descent*- und *annealing*-Ansatz für die horizontale Textübertragung auf Smartphones erstellt. Das Ergebnis einer empirischen Studie ist eine Verbesserung der Schreibgeschwindigkeit der Probanden um etwa 32% in horizontaler Geräteposition.



Abbildung 63: *KALQ*-Tastatur aus [Oulasvirta et al., 2013]

8.2.4 Weitere Belegungen für rechteckige Layouts

Die *OPTI I & II*-Belegungen aus Abbildung 64 und 66 verfügen laut MacKenzie u. Soukoreff [2002b] im vertikalen Fall über eine ähnliche Schreibgeschwindigkeit (Abweichung: 0,5%), de-

ren relative Verbesserung gegenüber der *QWERTY*-Tastatur in [Zhai et al., 2002b] mithilfe einer exakteren Parameterwahl (siehe auch Abschnitt 2.4.5) auf 25 % korrigiert wird. Der Vorteil der *FITALY*-Tastatur (siehe Abb. 65) wird mit 20 % angegeben.

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

Abbildung 64: *OPTI I*-Tastatur (siehe bspw. [MacKenzie u. Soukoreff, 2002b])

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

Abbildung 65: *FITALY*-Tastatur (siehe [Ichbiah, 1996])

		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		

Abbildung 66: *OPTI II*-Tastatur (siehe bspw. [MacKenzie u. Soukoreff, 2002b])

8.3 Vergleich der optimierten Tastenbelegungen mit alternativen Tastaturen

8.3.1 Vergleich der Eigenschaften im vertikalen Fall

Die Tabellen 16 und 17 geben Auskunft über die Leistungsfähigkeit der in der Arbeit präsentierten Tastaturen und Tastenbelegungen bei realistischer Bedienung in vertikaler Geräteposition.

Tabelle 16: Leistungsfähigkeit der in der Arbeit errechneten *distanzopt.* (*) oder *zeitopt.* (**) Tastenbelegungen in vertikaler Geräteposition.

Layout	Belegung	Distanz (LE)	Zeit (s)	Geschwindigkeit (wpm)
QWERTY-like	Abb. 23 *	220,372	27,8664	43,5171
3x9-Layout	Abb. 24 *	198,884	27,1582	44,6604
3x10-Layout	Abb. 25 *	199,199	27,126	44,7166
4x8-Layout	Abb. 26 *	185,922	26,5801	45,6607
QWERTY-like	Abb. 44 **	219,865	27,8433	43,5539
3x9-Layout	Abb. 45 **	199,316	27,1755	44,6314
3x10-Layout	Abb. 46 **	197,783	27,0569	44,8332
4x8-Layout	Abb. 47 **	186,087	26,5841	45,6537

Tabelle 17: Leistungsfähigkeit der Tastaturen und Tastenbelegungen aus verwandter Literatur bei vertikaler Geräteposition und realistischer Schreibweise.

Layout	Belegung	Distanz (LE)	Zeit (s)	Geschwindigk. (wpm)
QWERTY-like	QWERTY (Abb. 57)	331,389	33,1674	36,5021
QWERTY-like	ABC (Abb. 58)	334,753	33,2452	36,3977
QWERTY-like	Abb. 59	260,405	29,8661	40,5943
QWERTY-like	Abb. 60	236,001	28,7688	42,1355
QWERTY-like	Abb. 61	312,195	32,4602	37,295
Dvorak-Layout	DVORAK (Abb. 62)	394,192	35,6411	33,9583
4x8-Layout	KALQ (Abb. 63)	318,216	32,737	37,0066
5x7-Layout	OPTI I (Abb. 64)	197,564	27,3247	44,514
5x6-Layout	FITALY (Abb. 65)	208,212	27,9832	43,4351
5x6-Layout	OPTI II (Abb. 66)	191,873	27,056	44,9444

Bereits auf den ersten Blick offenbaren sich die Unterschiede in der Performance verschiedener Tastaturen und Belegungen und damit die Sinnhaftigkeit, welche in der Erforschung und Erzeugung von verbesserten, vertikalen Tastenbelegungen steckt.

Die geringe Abweichung der benötigten Schreibzeit auf distanz- und zeit-optimierten Belegungen konnte bereits in Abschnitt 6.1.2 beobachtet werden. Darüber hinaus stellt man fest, dass sich auch die jeweilige zurückzulegende Strecke sowie die Leistungsfähigkeit in Form der Schreibgeschwindigkeit nur minimal unterscheiden (um maximal 0,71 % bzw. 0,26 %).

Eine Analyse der Tastaturen aus der Literatur zeigt, dass nur *OPTI I & II* und *FITALY* mit den Tastenbelegungen aus Teil II der vorliegenden Arbeit konkurrieren können. Ihre guten Ergebnisse begründen sich vor allem darin, dass diese Tastaturen explizit für die Bedienung mit einem Finger erstellt wurden. Die Tastenbelegungen des *QWERTY-like*-Layouts aus Abbildung 59 und 60 basieren ebenfalls auf dem Versuch die vertikale Textübertragung effizienter zu gestalten. Der Grund, dass die zugehörigen Ergebnisse hinter denen von *OPTI I & II* und *FITALY* zurückbleiben, sind die unterschiedlichen Layoutformen (*QWERTY-like* vs. *5x6-* bzw. *5x7-*Layout). Die *DVORAK*-Tastatur, welche genau wie die Belegung aus Abbildung 61 für die 10-Finger-Schreibweise erstellt wurde, schneidet aus diesem Grund im vertikalen Fall sehr schlecht ab. Ihre Leistungsfähigkeit ist sogar geringer als die der Standardtastatur, was jedoch nicht den Ergebnissen aus [Buzing, 2003] (siehe Abschnitt 2.2) widerspricht, da in der genannten Arbeit die Bedienung mit 10-Fingern zugrundegelegt wurde. Die kürzlich vorgestellte *KALQ*-Tastatur (siehe Oulasvirta et al. [2013]) bringt für die einhändige Textübertragung kaum Vorteile gegenüber der Standardtastatur mit sich, was sich darin begründet, dass sie für die horizontale Schreibweise konzipiert ist.

Im Vergleich mit der *QWERTY*-Tastatur und den Tastaturen aus der Literatur (siehe Tabelle 17) liefern die in der vorliegenden Arbeit optimierten Belegungen (siehe Tabelle 16) meist deutlich bessere Ergebnisse hinsichtlich der zurückzulegenden Distanz (Reduktion um durchschnittlich 27,8 % und im besten Fall bis zu 52,8 %) und der benötigten Zeit (Reduktion um durchschnittlich 14,8 % und im besten Fall bis zu 25,4 %).

Da diese Werte aufgrund der unterschiedlichen zugrundeliegenden Layouts und Schreibweisen sowie dem Einfluss der alphabetischen (siehe Abb. 58) und Standard-*QWERTY*-Belegung (siehe Abb. 57) nur eine begrenzte Aussagekraft haben, wird darüber hinaus ein Vergleich zwischen den optimierten Tastenbelegungen aus Tabelle 16 und den für die 1-Finger-Schreibweise entwickelten Belegungen aus Tabelle 17 (siehe Abb. 59, 60, Abb. 64, 65 und 66) angestellt. Dieser liefert eine durchschnittliche Reduktion der zurückzulegenden Gesamtdistanz um 8,2 % und der benötigten Zeit um 3,6 %. Beschränkt man sich zusätzlich auf die Belegungen derselben Layouts, dann ergibt sich eine durchschnittliche Reduktion der Distanz und der Schreibdauer um 11,3 % beziehungsweise 5,0 % für das *QWERTY-like*-Layout.

Ein Grund für das Verbesserungspotential der Tastenbelegungen aus der Literatur hinsichtlich der zurückzulegenden Distanz und benötigten Zeit ist, dass dort statt der Distanz- und Zeitminimierung eine möglichst hohe Schreibgeschwindigkeit im Vordergrund steht. Trotzdem liegen auch in diesem Bereich die optimierten Belegungen vorne (durchschnittlich 3,5 % Verbesserungspotential bzw. 5,2 % auf dem *QWERTY-like*-Layout), was teilweise durch die Wahl des Textkorpus zu erklären ist. So wurden die Tastaturen aus Tabelle 17 oftmals nicht für die deutsche Sprache, sondern für die Übertragung eines englischen Textes entwickelt.

Gegenüber der Standard-*QWERTY*-Tastatur erzielen die optimierten Tastenbelegungen aus Abbildung 23 und 44 potentielle Verbesserungen von durchschnittlich 33,6 % im Bereich der

zurückzulegenden Distanz, 16,0% hinsichtlich der benötigten Zeit und 19,2% bezüglich der erzielbaren Geschwindigkeit.

Insgesamt kann man festhalten, dass die Tastenbelegungen, welche im Verlauf der Arbeit als Lösung verschiedener Optimierungsprobleme auftraten, im vertikalen Fall durchweg sehr gute Resultate erzielen und nur wenige der alternativen Tastaturen ähnliche Eigenschaften besitzen.

8.3.2 Vergleich der Eigenschaften im horizontalen Fall

Die Eigenschaften verschiedener Tastaturen und Tastenbelegungen hinsichtlich der zurückzulegenden Distanz, der benötigten Zeit und der Schreibgeschwindigkeit in horizontaler Geräteposition kann man den Tabellen 18 und 19 entnehmen. Den Werten der benötigten Zeit und der Geschwindigkeit liegt dabei die Annahme zugrunde, dass der Benutzer die Hover-over-Schreibweise verwendet.

Tabelle 18: Leistungsfähigkeit der in der Arbeit errechneten *distanzopt.* (*) oder *zeitopt.* (**) Tastenbelegungen in horizontaler Geräteposition.

Layout	Belegung	Distanz (LE)	Zeit (s)	Geschwindigkeit (wpm)
QWERTY-like	Abb. 34 *	183,791	24,3013	50,2618
3x9-Layout	Abb. 35 *	173,236	23,7396	51,4432
3x10-Layout	Abb. 36 *	167,751	23,2795	52,4558
4x8-Layout	Abb. 37 *	163,457	23,0041	53,1037
QWERTY-like	Abb. 49 **	184,167	23,4275	52,1897
3x9-Layout	Abb. 50 **	171,4	22,8984	53,3513
3x10-Layout	Abb. 51 **	171,351	23,0816	52,9168
4x8-Layout	Abb. 52 **	165,713	22,7836	53,624

Tabelle 19: Leistungsfähigkeit der Tastaturen und Tastenbelegungen aus verwandter Literatur in horizontaler Geräteposition.

Layout	Belegung	Distanz (LE)	Zeit (s)	Geschwindigk. (wpm)
QWERTY-like	QWERTY (Abb. 57)	255,454	26,9233	45,3133
QWERTY-like	ABC (Abb. 58)	269,736	28,0001	43,5273
QWERTY-like	Abb. 59	259,016	27,331	44,6808
QWERTY-like	Abb. 60	240,122	26,7825	45,6354
QWERTY-like	Abb. 61	251,26	27,035	45,1463
Dvorak-Layout	DVORAK (Abb. 62)	241,218	26,7153	45,6108
4x8-Layout	KALQ (Abb. 63)	242,43	26,0118	46,9321
5x7-Layout	OPTI I (Abb. 64)	277,603	28,1012	43,5012
5x6-Layout	FITALY (Abb. 65)	293,521	28,1789	43,365
5x6-Layout	OPTI II (Abb. 66)	250,131	26,6657	45,8965

Beim Blick auf die erzielbare Schreibgeschwindigkeit stellt man zuallererst fest, dass die Werte

deutlich über denen im vertikalen Fall liegen. Dies überrascht insofern nicht, als dass man im horizontalen Fall zwei Finger für die Textübertragung zur Verfügung hat, die sich die Schreibezeit aufteilen. Die zurückzulegende Distanz nimmt beim Wechsel der Geräteposition jedoch nicht im selben Maß ab, wie die für die Textübertragung benötigte Zeit abnimmt oder die erzielbare Schreibgeschwindigkeit zunimmt. Der Grund dafür liegt darin, dass sowohl die Zeit als auch die Geschwindigkeit im Gegensatz zur Distanz von der Verwendung der Hover-over-Schreibweise aus Abschnitt 6.2 profitieren.

Im Gegensatz zu Abschnitt 8.3.1 lässt sich diesmal eine geringfügig größere Geschwindigkeit bei den hinsichtlich der benötigten Zeit optimierten Tastenbelegungen feststellen. Erwartungsgemäß liegen die Vorteile der distanzoptimierten Belegungen im Bereich der zurückzulegenden Gesamtstrecke.

Vergleicht man die Ergebnisse der Tastenbelegungen aus Tabelle 18 mit denen aus Tabelle 19 so erkennt man, dass sämtliche Tastaturen aus der Literatur im horizontalen Fall einen deutlich größeren Schreibaufwand in Form der zurückzulegenden Distanz erfordern als die optimierten Belegungen. Analog verhält es sich für die erzielbaren Geschwindigkeiten und die benötigte Zeit. Dies liegt vor allem daran, dass der Großteil der Tastaturen aus Tabelle 19 für die Bedienung mit einem Finger konzipiert wurde.

Die *KALQ*-Tastatur, deren Erzeugung auf der 2-Finger-Schreibweise basiert, liefert diesbezüglich erwartungsgemäß die besten Ergebnisse. Der Grund warum sie trotzdem weit hinter den Werten aus Tabelle 18 zurückbleibt, ist die Verwendung einer alternativen Hover-over-Strategie im Entstehungsprozess (siehe auch Kapitel 9). Die Verwendung der *KALQ*-Tastatur würde unter den der Arbeit zugrundegelegten Voraussetzungen im Vergleich zu den optimierten 4x8-Tastenbelegungen (siehe Abb. 37 und 52) die Gesamtstrecke im Durchschnitt um 32,1 % verlängern und einen zeitlichen Mehraufwand 12,0 % hervorrufen. Die Verschlechterung hinsichtlich der erzielbaren Geschwindigkeit würde 10,8 % betragen.

Der Vergleich der Leistungsfähigkeit der errechneten Tastenbelegungen für das *QWERTY-like*-Layout aus Tabelle 18 mit der Standard-*QWERTY*-Tastatur offenbart die besseren Eigenschaften der Erstgenannten. Ihre Nutzung würde zu einer Reduktion der zurückzulegenden Strecke (-28,0 %) und der benötigten Zeit (-11,4 %), sowie einer um 13,0 % größeren Schreibgeschwindigkeit führen.

Erinnert man sich zurück an die Tastaturen, die im vertikalen Fall gute Resultate erzielen konnten (*OPTI I*, *OPTI II*, *FITALY*, siehe Abschnitt 8.3.1), und analysiert ihre Daten im horizontalen Fall, so zeigt sich dass ihre Leistungsfähigkeit für die 2-Daumen-Schreibweise deutlich unter den Leistungen der optimierten Tastenbelegungen liegt (bspw. eine bis zu 23,7 % langsamere Textübertragung). Der Grund dafür ist, dass ihrer Erzeugung eine vertikale Geräteposition zugrundelag. Unabhängig von der Performance müsste man jedoch allein aufgrund ihrer Proportionen über einen Einsatz auf Smartphones in horizontaler Position diskutieren (siehe Kapitel 9).

Aufgrund der größeren Ähnlichkeit des horizontalen Falls zur 10-Finger-Schreibweise können die *DVORAK*-Tastatur und die Belegung aus Abbildung 61 im Gegensatz zum vertikalen Fall hinsichtlich des Schreibaufwandes und der Leistung zur Standard-*QWERTY*-Tastatur aufschließen.

Zusammenfassend lässt sich feststellen, dass keine der in Abschnitt 8.2 präsentierten Tastaturen wirklich gute Ergebnisse für die horizontale Textübertragung liefert. Vielmehr sollte bei Verwendung der 2-Daumen-Schreibweise über einen Wechsel hin zu den im Verlauf der Arbeit optimierten Tastenbelegungen nachgedacht werden.

8.3.3 Vergleich der Eigenschaften im Hybrid-Fall

Für Hybrid-Anwendungen liefern die im Verlauf der Arbeit präsentierten Tastaturen und Tastenbelegungen die Resultate, welche in Tabelle 20 und 21 zusammengefasst sind. Analog zu den Ergebnissen in horizontaler Geräteposition (siehe Abschnitt 8.3.2) wird davon ausgegangen, dass die Hover-over-Schreibweise verwendet wird.

Tabelle 20: Leistungsfähigkeit der in der Arbeit errechneten *distanzopt.* (*) oder *zeitopt.* (**) Tastenbelegungen für Hybrid-Anwendungen.

Layout	Belegung	Distanz (LE)	Zeit (s)	Geschwindigkeit (wpm)
<i>QWERTY-like</i>	Abb. 40 *	218,02	27,2141	44,7506
3x9-Layout	Abb. 41 *	206,753	26,8374	45,3833
3x10-Layout	Abb. 42 *	211,635	27,0307	45,0489
4x8-Layout	Abb. 43 *	197,567	26,5032	45,984
<i>QWERTY-like</i>	Abb. 53 **	218,755	27,2262	44,7288
3x9-Layout	Abb. 54 **	205,641	26,7147	45,5946
3x10-Layout	Abb. 55 **	214,665	27,039	45,0316
4x8-Layout	Abb. 56 **	202,973	26,6507	45,723

Bei der Auswertung der Daten bestätigt sich die Vermutung, dass die Werte der zurückzulegenden Distanz, der benötigten Zeit und der erzielbaren Geschwindigkeit für die Tastaturen aus Tabelle 21 zwischen denen der einfachen Gerätepositionen liegen. Auch die optimierten Tastenbelegungen aus Tabelle 20 erreichen nicht mehr die guten Werte, welche für die horizontale Geräteposition in Abschnitt 8.3.2 zustandekamen. Der Grund dafür sind die bereits mehrfach erwähnten konträren Anforderungen der verschiedenen Gerätepositionen an eine Tastenbelegung, welche im Hybrid-Fall vereint werden müssen. Diese haben auch zur Folge, dass der Unterschied zwischen den Tastaturen aus der Literatur und den optimierten Tastenbelegungen im Hybrid-Fall weniger groß ausfällt, als dies zuvor für vertikale und horizontale Anwendungen der Fall war. Nichtsdestotrotz sind die optimierten Tastenbelegungen aus Tabelle 20 den Tastaturen aus der Literatur (siehe Tabelle 21) in der Regel überlegen.

Legt man dem Vergleich nur die Tastaturen aus Tabelle 21 zugrunde, deren Motivation eine

Tabelle 21: Leistungsfähigkeit der Tastaturen und Tastenbelegungen aus verwandter Literatur für Hybrid-Anwendungen.

Layout	Belegung	Distanz (LE)	Zeit (s)	Geschwindigk. (wpm)
<i>QWERTY-like</i>	QWERTY (Abb. 57)	293,421	30,0454	40,4332
<i>QWERTY-like</i>	ABC (Abb. 58)	302,245	30,6226	39,6445
<i>QWERTY-like</i>	Abb. 59	259,71	28,5985	42,5396
<i>QWERTY-like</i>	Abb. 60	238,061	27,7757	43,8157
<i>QWERTY-like</i>	Abb. 61	281,727	29,7476	40,8468
<i>Dvorak-Layout</i>	DVORAK (Abb. 62)	317,705	31,1782	38,9314
<i>4x8-Layout</i>	KALQ (Abb. 63)	280,323	29,3744	41,3825
<i>5x7-Layout</i>	OPTI I (Abb. 64)	237,583	27,713	44,0018
<i>5x6-Layout</i>	FITALY (Abb. 65)	250,866	28,0811	43,4
<i>5x6-Layout</i>	OPTI II (Abb. 66)	221,002	26,8609	45,4154

Verbesserung der *QWERTY*-Tastatur war und welche für die Bedienung eines Smartphone-ähnlichen Gerätes konzipiert sind, und lässt deshalb die Belegungen der Abbildungen 57, 61, 58 und 62 außen vor, so erhält man folgende Unterschiede hinsichtlich der durchschnittlichen Eigenschaften. Die hybrid-optimierten Tastenbelegungen aus Tabelle 20 sind in der Lage die durchschnittlich zurückzulegende Strecke um 15,5 % und die durchschnittliche Schreibdauer um 4,2 % zu reduzieren. Die erzielbare Geschwindigkeit liegt im Durchschnitt um 4,6 % höher als die der einbezogenen Tastaturen aus Tabelle 21.

Der Vergleich mit der Standard-*QWERTY*-Tastatur liefert die Überlegenheit der in Kapitel 5 und Abschnitt 6.2.2 optimierten Tastenbelegungen, welche sich in einer reduzierten Gesamtstrecke (-28,6 %), einer reduzierten Schreibdauer (-10,5 %) und einer höheren Schreibgeschwindigkeit (+12,0 %) ausdrückt. Beschränkt man sich beim Vergleich auf hybrid-optimierte Belegungen für das *QWERTY-like*-Layout (siehe Abb. 40 und 53) erhält man geringfügig schlechtere Werte für das Verbesserungspotential (Distanz: -25,6 %, Zeit: -9,4 %, Geschwindigkeit: +10,6 %), da der positive Einfluss der leistungsfähigen rechteckigen Layouts verloren geht.

8.3.4 Zusammenfassung der Ergebnisse der Vergleiche

Die Auswertung der Ergebnisse und der Vergleich der Tastenbelegungen und Tastaturen in den Abschnitten 8.3.1, 8.3.2 und 8.3.3 haben gezeigt, dass sich die Entwicklung und Einführung einer neuen Tastenbelegung für Smartphones aus Sicht der zurückzulegenden Strecke, der benötigten Zeit und der Schreibgeschwindigkeit prinzipiell lohnen würde.

Eine Nutzung der als Konkurrent zur *QWERTY*-Tastatur bekannten *DVORAK*-Tastatur (siehe Abb. 62) auf Smartphones ist nicht zu empfehlen, da sie insbesondere bei einhändiger Bedienung allen anderen Tastaturen und Belegungen deutlich unterlegen ist. Ein erstaunliches Ergebnis ist, dass die alphabetische Belegung (siehe Abb. 58) nur im horizontalen Fall

schlechtere Eigenschaften als die *QWERTY*-Tastatur mit sich bringt. Auch die alternativen Tastaturen, welche in Abschnitt 8.2 vorgestellt wurden, besitzen teilweise noch großes Verbesserungspotential hinsichtlich der Übertragung eines deutschen Textes auf Smartphones, weshalb von einer Empfehlung abgesehen werden muss. Insbesondere im horizontalen Fall weisen sie deutliche Schwächen auf.

Unabhängig von der Einsatztauglichkeit in der Praxis stellt sich in diesem Zusammenhang die Frage, ob die relativen Ergebnisse aus der Literatur für die Übertragung eines deutschen Textes reproduziert werden können. Mit Ausnahme der Tastenbelegung in Abbildung 59 (siehe [Li et al., 2006]) waren die in der vorliegenden Arbeit bestimmten relativen Geschwindigkeitszuwächse mit denen der Quellen vergleichbar. Ein möglicher Grund für die Abweichung der Werte im Vergleich zur erstgenannten Arbeit ist neben der Korpuswahl beispielsweise die Platzierung der im Deutschenn sehr häufigen Buchstaben *E*, *N* und *R* auf Tasten, die keinen minimalen Abstand zueinander haben (siehe Abb. 59). Der zugrundegelegte Text dürfte auch ein Grund für die (um 7,2%) geringere Reduktion der Distanz bei der Belegung aus Abbildung 60 sein. Die große Diskrepanz hinsichtlich der relativen Verbesserung der Schreibgeschwindigkeit der *KALQ*-Tastatur im Vergleich zur *QWERTY*-Tastatur (empirische Studie: +32 %, Werte aus Tabelle 19: +4 %) ist nur durch die abweichende Schreibweise zu erklären, die der Erzeugung der *KALQ*-Tastatur zugrundelag (siehe Kapitel 9).

Darüber hinaus haben die Resultate aus den Abschnitten 8.3.1 - 8.3.3 gezeigt, dass die vorliegende Arbeit und insbesondere Kapitel 3 die theoretische Basis für eine leistungsfähige Verteilung der Buchstaben liefert. Die aus der Optimierung resultierenden Belegungen sind den fortgeschritteneren Belegungen aus Abschnitt 8.2 (siehe Abb. 59, 60 und 63 - 66) je nach Geräteposition und Evaluationskriterium (Distanz, Zeit, Geschwindigkeit) um bis zu 41,6 % (vertikal), 44,3 % (horizontal) oder 29,5 % (hybrid) überlegen. Diese Überlegenheit setzt sich beim Vergleich mit der *QWERTY*-Tastatur fort, sodass am Ende der Arbeit für alle Positionen Tastenbelegungen zur Verfügung stehen, welche leistungsfähiger sind als die derzeitige Standard-Smartphone-Tastatur (siehe Tabelle 22).

Tabelle 22: *Betragsmäßig größte, relative Verbesserung der Leistungsfähigkeit der Standard-QWERTY-Tastatur hinsichtlich verschiedener Kriterien durch eine in der vorliegenden Arbeit errechnete (jeweils in Klammern genannte) Tastenbelegung.*

Position	Distanz	Zeit	Geschwindigkeit
vertikal	-43,9 % (Abb. 26)	-19,9 % (Abb. 26)	+25,1 % (Abb. 26)
horizontal	-36,0 % (Abb. 37)	-15,4 % (Abb. 52)	+18,3 % (Abb. 52)
hybrid	-32,7 % (Abb. 43)	-11,8 % (Abb. 43)	+13,7 % (Abb. 43)

Betrachtet man die Tastenbelegungen, welche die größten Verbesserungen gegenüber der *QWERTY*-Tastatur hervorrufen und welche in Tabelle 22 aufgelistet sind, so bemerkt man, dass all diese Belegungen für das 4x8-Layout entwickelt wurden. Untersucht man die maxima-

Die Verbesserung der *QWERTY*-Tastatur durch Tastenbelegungen für das *QWERTY-like*-Layout, dann ergeben sich die Werte aus Tabelle 23. Der Unterschied in den Werten der beiden Tabellen zugunsten der *4x8*-Belegungen ist ein Indiz dafür, dass neben der Optimierung der Tastenbelegung auch an einem besseren Tastaturlayout gearbeitet werden sollte (siehe Kapitel 10).

Tabelle 23: *Betragsmäßig größte, relative Verbesserung der Leistungsfähigkeit der Standard-QWERTY-Tastatur hinsichtlich verschiedener Kriterien durch eine in der vorliegenden Arbeit errechnete QWERTY-like-Tastenbelegung.*

Position	Distanz	Zeit	Geschwindigkeit
vertikal	-33,7 % (Abb. 44)	-16,1 % (Abb. 44)	+19,3 % (Abb. 44)
horizontal	-28,0 % (Abb. 34)	-13,0 % (Abb. 49)	+15,2 % (Abb. 49)
hybrid	-25,7 % (Abb. 40)	-9,4 % (Abb. 40)	+10,7 % (Abb. 40)

Neben der Überlegenheit der optimierten Tastaturen konnte in der Auswertung beobachtet werden, dass die Unterschiede zwischen den distanz-minimierten (siehe Kapitel 3, 4 und 5) Tastenbelegungen und den zeit-optimierten Belegungen (aus Kapitel 6) in allen Positionen nicht sehr groß sind. Aus diesem Grund ist es naheliegend sich beispielsweise für die Optimierung von Tastenbelegungen alternativer Layouts zukünftig für eines der beiden Ziele zu entscheiden.

9 Diskussion

In Kapitel 8 wurde die theoretische Überlegenheit der optimierten Tastenbelegungen gegenüber den Tastaturen aus der Literatur für das in Abschnitt 2 gewählte Setting festgestellt. Trotz der möglichst realistischen Wahl der Voraussetzungen lohnt es sich einen Blick zurückzuwerfen, um die Ergebnisse der Arbeit im Gesamtkontext der Tastaturforschung einordnen zu können.

Es wurde zu Beginn der Arbeit festgelegt, Tastenbelegungen für vorgegebene Layouts zu optimieren. Zu diesem Zweck wurden in Abschnitt 2.4.3 vier Layouts gewählt, ohne zuvor untersucht zu haben, ob diese Tastenanordnungen für eine leistungsfähige Tastatur geeignet sind. Wir haben uns bei der Wahl der Layouts an den derzeit verwendeten Tastaturen orientiert und darauf geachtet, dass sie bezüglich ihrer Größe aus praktischer Sicht kompatibel sind zu gewöhnlichen Smartphones. Sowohl ihre Höhe als auch die Breite spielen in diesem Zusammenhang eine wichtige Rolle, da die Ansteuerbarkeit der einzelnen Tasten und die Sichtbarkeit beispielsweise eines Textfeldes gewährleistet sein muss. Insofern könnte man durch die Wahl alternativer Layouts potentiell die Ergebnisse weiter verbessern, jedoch ist fraglich, ob die resultierenden Tastaturen für den Einsatz in der Praxis geeignet wären.

Darüber hinaus wurden die Verteilung der einzelnen Tasten auf die Hände festgelegt. Eine mittige Aufteilung erscheint nicht zuletzt aufgrund der Händigkeit der Nutzer sinnvoll, jedoch müssten für eine einwandfreie Legitimation zunächst Modelle erarbeitet oder Studien durchgeführt werden, die diese Thematik behandeln. Solange die Disjunktheit der Tastenmengen der beiden Seiten erhalten bleibt, beeinflusst dies zwar die resultierenden Tastenbelegungen, jedoch gelten die theoretischen Aussagen der Arbeit unverändert. Analog verhält es sich, wenn die Teilung nicht nur im Hintergrund stattfindet, sondern das Layout zusätzlich optisch gesplittet wird, was zu einer Veränderung der Distanzen zwischen den Tasten verschiedener Seiten führt. Die Existenz von Tasten, welche von beiden Daumen angesteuert werden können, würde die Überarbeitung der präsentierten mathematischen Formulierungen erfordern.

Um das Ziel der Minimierung der zurückzulegenden Gesamtstrecke beziehungsweise der benötigten Zeit zu erreichen, war es notwendig, bestimmte Start- und Wartepunkte für die Finger zu definieren. Dazu ist zu sagen, dass die Wahl der Punkte vor allem die Optimierungsprobleme beeinflusst, denen eine Schreibweise mit häufiger Ansteuerung jener Punkte zugrundeliegt. Betrachtet man beispielsweise die realistische Schreibweise im vertikalen Fall (Abschnitt 3.2), so spielt der Startpunkt pro übertragenem Text nur zweimal eine Rolle, nämlich zu Be-

ginn und am Ende. Der Einfluss hängt damit von der gewählten Textlänge B ab, ist für eine hinreichend große Wahl aber gering. Es stellt sich die Frage, ob im horizontalen Fall nicht auch eine Schreibweise existiert, welche ohne die Rückkehr zum Wartepunkt auskommt. Tatsächlich gibt es Literatur, die auf der Verwendung einer solchen Schreibweise basiert (siehe [Oulasvirta et al., 2013]) und davon ausgeht, dass ein Finger über seiner zuletzt getippten Taste verharnt, bis er eine neue Aufgabe bekommt. In der Zwischenzeit erledigt der andere Finger seine Schreibarbeit. Meiner Ansicht nach ist diese Schreibweise nicht realitätsnah, weil sie davon ausgeht, dass der Nutzer seine Finger in unnatürlicher Weise auf den zuvor gedrückten Tasten positioniert lässt. Die Inkaufnahme dieses muskulären Aufwands kann ich weder aus eigener Erfahrung noch aus Befragungen und Beobachtungen in der Praxis bestätigen. Trotzdem sei angemerkt, dass die Resultate des in Abschnitt 6.2 erarbeiteten Ansatzes eine obere Schranke an eben genannte Schreibweise ist. Eine untere Schranke könnte man erhalten, indem man in Gleichung (6.7) die Dauer einer Bewegung zwischen zwei Tasten auf verschiedenen Seiten gleich Null setzt. Im Gegensatz zur konkreten Wahl in Abschnitt 2.4.4 hängt die Position der Wartepunkte in der Realität sicher von Faktoren wie der Form und Größe des Geräts ab.

Zur bereits erwähnten Wahl der Textlänge B ist zu bemerken, dass sie für die Gewichtung zwischen der Positionierung der Buchstaben zueinander und ihrer Positionierung auf dem Tastaturlayout verantwortlich ist. Eine Reduktion der durchschnittlichen Buchstabenanzahl eines Textes führt dazu, dass sich die resultierenden Tastenbelegungen den Ergebnissen aus Abschnitt 3.1 und 4.1 angleichen, da die zugehörigen Optimierungsprobleme lediglich die Positionierung auf dem Layout betrachten.

Für die Erzeugung von Tastenbelegungen im Hybrid-Fall (siehe Kapitel 5) war es nötig ein Verhältnis zwischen vertikaler und horizontaler Nutzung zu wählen. In der vorliegenden Arbeit wurde mangels Studien zu diesem Thema davon ausgegangen, dass die Textübertragung in beiden Positionen zu gleichen Teilen erfolgt. Eine Anpassung auf zukünftig potentiell verfügbare Parameterwerte würde die Lösungen in Richtung der Ergebnisse für die einfachen Gerätepositionen (siehe Kapitel 3 und 4) verändern.

In Kapitel 2 wurde die Minimierung der zurückzulegenden Gesamtdistanz als ein mögliches Ziel der Optimierung identifiziert. Für eine Tastenbelegung, welche neben dem Schreibaufwand auch Eigenschaften wie die Fehleranzahl miteinbezieht, gilt es herauszufinden, ob eine geringe Distanz zwischen den Tasten überhaupt wünschenswert ist, oder aber beispielsweise die Fehleranzahl nach oben treibt. Im Gegensatz zu verwandter Literatur konzentriert sich die vorliegende Arbeit auf die Minimierung der zurückzulegenden Gesamtdistanz oder der benötigten Schreibdauer anstatt auf die Maximierung der Schreibgeschwindigkeit. Letzteres ist über die Zusammenhänge (8.1) und (8.2) äquivalent zur Minimierung der durchschnittlichen Eingabedauer eines Buchstabens. Somit unterscheidet sich die Zielfunktion, auf welcher sämtliche präsentierten Ergebnisse basieren, von der in der Literatur verwendeten Zielfunktion nur durch den Zusatzterm, welcher durch den Ansatz OBJFCT hinzukam (siehe 3.2.3). Verzicht-

tet man auf diesen Zusatzterm, der anschaulich als ein Gefälle in der Zielfunktion interpretiert werden kann, so ließe sich der Tastaturbereich T auf dem der Buchstabe \tilde{i} fixiert wird weiter einschränken, was positive Effekte auf die Lösbarkeit der Optimierungsprobleme (aus zeitlicher Sicht) mit sich bringt. Darüber hinaus zieht die Festlegung auf ein Ziel der Optimierung keine großen Veränderungen nach sich. Es sollte an dieser Stelle nicht unerwähnt bleiben, dass das Tool *VirtualKeyboardOptimization.mos* bei geeigneten Einstellungen in der Lage ist sowohl die Distanz und die Schreibdauer als auch die Schreibgeschwindigkeit zu optimieren.

Die Minimierung der benötigten Schreibdauer basiert wie auch die Schreibgeschwindigkeit auf *Fitts' Law* (siehe Abschnitt 2.4.5), welches jedoch für sehr fortgeschrittene Schreiber konzipiert ist. Ist man daran interessiert, dass auf einer Tastenbelegung auch Anfänger gute Resultate erzielen, ist es nötig die Modelle daraufhin anzupassen.

Die Optimierung von Tastenbelegungen im horizontalen Fall in Kapitel 4 basiert auf der Annahme, dass sich beide Daumen gleich schnell bewegen. Praktischen Studien (siehe bspw. [Oulasvirta et al., 2013]) zufolge existiert jedoch ein Unterschied zwischen den Geschwindigkeiten der Ansteuerung einer Taste. Um die Praxistauglichkeit der Tastenbelegungen zu erhöhen, müsste ausführlicher in diese Richtung geforscht werden (siehe auch Kapitel 10).

Die Auswahl des Textkorpus ist ein weiterer Punkt über den diskutiert werden kann. Die präsentierten Tastenbelegungen sind aufgrund der Gewinnung der Bigrammwahrscheinlichkeiten (siehe Abschnitt 2.4.1) für die Übertragung eines deutschen Textes optimiert. Je nachdem wie empfänglich die großen Smartphone-Hersteller für die Einführung einer eigenen Tastenbelegung in jedem Land sind, müsste man gegebenenfalls an der Mehrsprachlichkeit der Tastenbelegungen arbeiten, anstatt sich auf eine Sprache festzulegen.

Neben der Verwendung von Wahrscheinlichkeiten, welche aus einem deutschen Text hervorgegangen sind, wurden im Verlauf der Arbeit nur die 26 Standardbuchstaben auf die Tasten der Layouts verteilt. Es wurde nicht zwischen Groß- und Kleinschreibung unterschieden, Umlaute, Sonderzeichen, Leertaste(n), Symbole und Funktionstasten wurden nicht betrachtet. Natürlich wäre es möglich all diese Tasten in einem zweiten Schritt manuell um die optimierte Tastenbelegung herumzulegen, es würde sich jedoch anbieten die Input-Daten diesbezüglich zu erweitern und so die besonderen Zeichen in die Optimierung aufzunehmen. Die Folge dieser Vergrößerung der Menge an Zeichen wäre ein Anwachsen der Optimierungsprobleme, was sich negativ auf deren Lösungsdauer auswirkt.

In Abschnitt 3.2.4 wurden sogenannte Reduktionsmethoden eingeführt, welche maßgeblich an der Reduktion der Laufzeit beteiligt waren. Eine Konsequenz dieser Methoden war es jedoch, dass die Lage einiger Buchstaben zueinander nicht optimiert wurde. Zugunsten der zeitlichen Lösbarkeit wurde also auf die globale Optimalität der Tastenbelegungen verzichtet. Die Resultate aus Kapitel 8 legitimieren rückblickend dieses Vorgehen und es sei erwähnt, dass ein Teil der eben genannten, unerwünschten Auswirkungen mithilfe nachgeschalteter Heuristiken abfangen werden kann (siehe Kapitel 10).

10 Ausblick

Aufgrund der vielen individuellen Faktoren wie beispielsweise der Präferenz einer Geräteposition, welche in die Erzeugung von besseren Tastenbelegungen einfließen, ist es nahezu ausgeschlossen, dass eine Tastenbelegung alle Nutzer gleichermaßen zufriedenstellt. Nichtsdestotrotz ist die Suche nach einer besseren Tastenbelegung generell sinnvoll, da die Verwendung von *QWERTY* oder eine unbedachte Wahl der Belegung einen deutlich höheren Aufwand provoziert als nötig. Aus diesem Grund wird auf absehbare Zeit weiter an Tastaturen und deren Belegungen geforscht werden.

Eine Möglichkeit die Resultate der vorliegenden Arbeit weiter zu verbessern ist die Verwendung einer Heuristik im Anschluss an die Optimierung. Erste Tests mit dem Tool *VirtualKeyboardOptimization.mos*, in welchem das *2-opt*-Verfahren bereits implementiert wurde, haben gezeigt, dass sich die präsentierten Tastenbelegungen durch 1000 Iterationen um durchschnittlich 5,2% verbessern lassen. Die Dauer der implementierten *2-opt*-Heuristik beträgt dabei etwa 0,26 Sekunden pro Iteration. Die Verwendung alternativer heuristischer Methode könnte im Mittelpunkt zukünftiger Forschungsprojekte stehen.

Außerdem ist es nicht unwahrscheinlich, dass die Optimierung von Tastenbelegungen in Zukunft auch die Verteilung von Umlauten und Sonderzeichen umfasst, was die Qualität der Resultate weiter verbessern würde. Die Frage nach der Form, Größe und Positionierung einer oder mehrerer Leertasten zählt zu den spannendsten in diesem Bereich und wird bei zukünftigen Forschungsansätzen eine große Rolle spielen.

Ein weiterer Punkt, welcher für einhändig bediente Geräte bereits im Mittelpunkt der Forschung stand ist die Erlernbarkeit einer schnellen Textübertragung durch Anfänger. Im Bereich der Smartphones könnte diese Thematik weiter untersucht werden.

Erinnert man sich zurück an die Tastenbelegungen aus Abschnitt 5, die häufige Buchstaben auf einer Layoutseite platzierten, ist es naheliegend über eine aus Sicht der Häufigkeiten ausgeglichene Verteilung der Buchstaben anzustreben, um eine Überlastung eines Fingers zu vermeiden.

Bevor man sich jedoch einer umfassenden Optimierung der Smartphone-Tastatur für alle Gerätepositionen widmet, müssten noch einige Punkte hinsichtlich der genauen Handhabung insbesondere im horizontalen Fall geklärt werden. Im Vergleich zum vertikalen Fall besteht in diesem Bereich noch Aufholbedarf hinsichtlich der verwendeten Schreibweisen in der Praxis.

Anhang

A Input-Daten

A.1 Konkrete Buchstaben- und Bigrammwahrscheinlichkeiten

Tabelle 24: Buchstabenwahrscheinlichkeiten p_i

a	6.27	b	2.12	c	2.66	d	5.04	e	17.55	f	1.68	g	2.96
h	4.05	i	7.78	j	0.28	k	1.43	l	3.71	m	2.6	n	9.51
o	2.96	p	1.01	q	0.03	r	7.64	s	6.2	t	6.2	u	4.48
v	0.96	w	1.45	x	0.07	y	0.17	z	1.21				

Tabelle 25: Bigrammwahrscheinlichkeiten p_{ik}

aa	0.05	ab	0.24	ac	0.27	ad	0.14	ae	0.57	af	0.13	ag	0.17
ah	0.26	ai	0.07	aj	0.01	ak	0.07	al	0.64	am	0.31	an	0.98
ao	0.01	ap	0.06	aq	0	ar	0.55	as	0.43	at	0.42	au	0.77
av	0.03	aw	0.02	ax	0.01	ay	0.02	az	0.02	ba	0.23	bb	0.01
bc	0	bd	0.03	be	1.04	bf	0.01	bg	0.03	bh	0.01	bi	0.19
bj	0.01	bk	0	bl	0.08	bm	0	bn	0.01	bo	0.06	bp	0
bq	0	br	0.13	bs	0.06	bt	0.04	bu	0.13	bv	0	bw	0.02
bx	0	by	0	bz	0.01	ca	0.04	cb	0	cc	0	cd	0.01
ce	0.03	cf	0	cg	0	ch	2.3	ci	0.02	cj	0	ck	0.16
cl	0.01	cm	0	cn	0	co	0.05	cp	0	cq	0	cr	0.01
cs	0.01	ct	0.01	cu	0.01	cv	0	cw	0	cx	0	cy	0
cz	0	da	0.5	db	0.05	dc	0.01	dd	0.15	de	2.35	df	0.03
dg	0.04	dh	0.04	di	0.93	dj	0.01	dk	0.03	dl	0.06	dm	0.05
dn	0.04	do	0.12	dp	0.02	dq	0	dr	0.09	ds	0.12	dt	0.07
du	0.2	dv	0.04	dw	0.06	dx	0	dy	0.01	dz	0.03	ea	0.28
eb	0.53	ec	0.24	ed	0.58	ee	0.28	ef	0.26	eg	0.48	eh	0.5
ei	1.97	ej	0.03	ek	0.19	el	0.78	em	0.55	en	3.73	eo	0.07
ep	0.12	eq	0.01	er	4	es	1.33	et	0.63	eu	0.39	ev	0.16
ew	0.22	ex	0.04	ey	0.01	ez	0.15	fa	0.2	fb	0.02	fc	0.01
fd	0.07	fe	0.26	ff	0.11	fg	0.04	fh	0.01	fi	0.12	fj	0

A Input-Daten

fk	0.01	fl	0.09	fm	0.01	fn	0.02	fo	0.13	fp	0.01	fq	0
fr	0.12	fs	0.04	ft	0.17	fu	0.23	fv	0.01	fw	0.01	fx	0
fy	0	fz	0.01	ga	0.19	gb	0.03	gc	0	gd	0.12	ge	1.38
gf	0.02	gg	0.03	gh	0.03	gi	0.17	gj	0	gk	0.04	gl	0.12
gm	0.03	gn	0.04	go	0.05	gp	0.01	gq	0	gr	0.21	gs	0.16
gt	0.12	gu	0.12	gv	0.04	gw	0.03	gx	0	gy	0.01	gz	0.03
ha	0.54	hb	0.04	hc	0.01	hd	0.16	he	0.98	hf	0.02	hg	0.04
hh	0.02	hi	0.28	hj	0.01	hk	0.03	hl	0.2	hm	0.11	hn	0.2
ho	0.19	hp	0.01	hq	0	hr	0.42	hs	0.13	ht	0.37	hu	0.14
hv	0.03	hw	0.08	hx	0	hy	0.01	hz	0.03	ia	0.09	ib	0.06
ic	0.72	id	0.13	ie	1.64	if	0.08	ig	0.37	ih	0.1	ii	0.03
ij	0.01	ik	0.11	il	0.28	im	0.29	in	1.7	io	0.2	ip	0.03
iq	0	ir	0.18	is	0.79	it	0.79	iu	0.03	iv	0.08	iw	0.02
ix	0.01	iy	0	iz	0.04	ja	0.14	jb	0	jc	0	jd	0
je	0.07	jf	0	jg	0	jh	0	ji	0	jj	0	jk	0
jl	0	jm	0	jn	0	jo	0.02	jp	0	jq	0	jr	0
js	0	jt	0	ju	0.04	jv	0	jw	0	jx	0	jy	0
jz	0	ka	0.23	kb	0.01	kc	0	kd	0.02	ke	0.26	kf	0.01
kg	0.01	kh	0.01	ki	0.07	kj	0	kk	0.01	kl	0.08	km	0.02
kn	0.01	ko	0.21	kp	0	kq	0	kr	0.1	ks	0.05	kt	0.16
ku	0.11	kv	0.01	kw	0.01	kx	0	ky	0.01	kz	0.01	la	0.45
lb	0.08	lc	0.02	ld	0.15	le	0.67	lf	0.04	lg	0.08	lh	0.02
li	0.63	lj	0	lk	0.04	ll	0.39	lm	0.05	ln	0.05	lo	0.15
lp	0.02	lq	0	lr	0.02	ls	0.26	lt	0.31	lu	0.17	lv	0.03
lw	0.03	lx	0	ly	0.02	lz	0.03	ma	0.4	mb	0.1	mc	0.01
md	0.08	me	0.51	mf	0.04	mg	0.05	mh	0.03	mi	0.39	mj	0.05
mk	0.03	ml	0.03	mm	0.19	mn	0.04	mo	0.13	mp	0.08	mq	0
mr	0.03	ms	0.11	mt	0.08	mu	0.13	mv	0.03	mw	0.04	mx	0
my	0.01	mz	0.03	na	0.68	nb	0.27	nc	0.06	nd	1.92	ne	1.19
nf	0.19	ng	0.93	nh	0.15	ni	0.52	nj	0.05	nk	0.25	nl	0.12
nm	0.16	nn	0.38	no	0.19	np	0.09	nq	0	nr	0.09	ns	0.66
nt	0.6	nu	0.34	nv	0.16	nw	0.24	nx	0	ny	0.01	nz	0.25
oa	0.02	ob	0.08	oc	0.12	od	0.12	oe	0.29	of	0.07	og	0.08
oh	0.08	oi	0.01	oj	0.01	ok	0.04	ol	0.26	om	0.16	on	0.64
oo	0.02	op	0.07	oq	0	or	0.45	os	0.14	ot	0.1	ou	0.04
ov	0.03	ow	0.05	ox	0.01	oy	0	oz	0.02	pa	0.14	pb	0
pc	0	pd	0.01	pe	0.14	pf	0.06	pg	0	ph	0.04	pi	0.1
pj	0	pk	0	pl	0.06	pm	0	pn	0	po	0.09	pp	0.05
pq	0	pr	0.18	ps	0.02	pt	0.05	pu	0.04	pv	0	pw	0
px	0	py	0	pz	0	qa	0	qb	0	qc	0	qd	0
qe	0	qf	0	qg	0	qh	0	qi	0	qj	0	qk	0
ql	0	qm	0	qn	0	qo	0	qp	0	qq	0	qr	0
qs	0	qt	0	qu	0.03	qv	0	qw	0	qx	0	qy	0
qz	0	ra	0.73	rb	0.27	rc	0.15	rd	0.72	re	1.14	rf	0.19
rg	0.26	rh	0.18	ri	0.57	rj	0.04	rk	0.24	rl	0.17	rm	0.21

A.1 Konkrete Buchstaben- und Bigrammwahrscheinlichkeiten

rn	0.28	ro	0.33	rp	0.08	rq	0	rr	0.14	rs	0.56	rt	0.52
ru	0.43	rv	0.11	rw	0.18	rx	0	ry	0.01	rz	0.16	sa	0.32
sb	0.12	sc	0.84	sd	0.18	se	0.86	sf	0.07	sg	0.14	sh	0.09
si	0.56	sj	0.03	sk	0.09	sl	0.07	sm	0.09	sn	0.05	so	0.24
sp	0.24	sq	0	sr	0.05	ss	0.48	st	1.21	su	0.15	sv	0.08
sw	0.1	sx	0	sy	0.03	sz	0.09	ta	0.57	tb	0.09	tc	0.01
td	0.34	te	1.94	tf	0.06	tg	0.09	th	0.15	ti	0.58	tj	0.02
tk	0.05	tl	0.14	tm	0.09	tn	0.07	to	0.18	tp	0.03	tq	0
tr	0.32	ts	0.39	tt	0.23	tu	0.32	tv	0.08	tw	0.21	tx	0
ty	0.02	tz	0.26	ua	0.04	ub	0.07	uc	0.17	ud	0.08	ue	0.71
uf	0.28	ug	0.1	uh	0.02	ui	0.02	uj	0	uk	0.04	ul	0.09
um	0.24	un	1.25	uo	0	up	0.06	uq	0	ur	0.55	us	0.44
ut	0.22	uu	0.01	uv	0.02	uw	0.01	ux	0	uy	0	uz	0.02
va	0.05	vb	0	vc	0	vd	0	ve	0.39	vf	0	vg	0
vh	0	vi	0.09	vj	0	vk	0	vl	0	vm	0	vn	0
vo	0.4	vp	0	vq	0	vr	0	vs	0	vt	0	vu	0
vv	0	vw	0	vx	0	vy	0	vz	0	wa	0.31	wb	0
wc	0	wd	0	we	0.48	wf	0	wg	0	wh	0	wi	0.32
wj	0	wk	0	wl	0	wm	0	wn	0.01	wo	0.09	wp	0
wq	0	wr	0	ws	0.01	wt	0	wu	0.2	wv	0	ww	0
wx	0	wy	0	wz	0	xa	0.01	xb	0	xc	0	xd	0
xe	0.01	xf	0	xg	0	xh	0	xi	0.02	xj	0	xk	0
xl	0	xm	0	xn	0	xo	0	xp	0.01	xq	0	xr	0
xs	0	xt	0.01	xu	0	xv	0	xw	0	xx	0	xy	0
xz	0	ya	0.01	yb	0.01	yc	0.01	yd	0.01	ye	0.01	yf	0
yg	0	yh	0	yi	0	yj	0	yk	0	yl	0.01	ym	0.02
yn	0.01	yo	0.01	yp	0.01	yq	0	yr	0.01	ys	0.03	yt	0.01
yu	0	yv	0	yw	0	yx	0	yy	0	yz	0	za	0.05
zb	0.02	zc	0	zd	0.02	ze	0.28	zf	0.01	zg	0.01	zh	0.01
zi	0.11	zj	0	zk	0.01	zl	0.01	zm	0.01	zn	0.01	zo	0.03
zp	0	zq	0	zr	0	zs	0.01	zt	0.07	zu	0.42	zv	0.01
zw	0.1	zx	0	zy	0	zz	0.01						

A.2 Numerische Beschreibung der Layouts

A.2.1 Layout 1 - QWERTY-like

- $\text{numberofkeys} = 26$, Rows = 3, Columns = (10,9,7)

Tabelle 26: Tastenbelegungen für das QWERTY-like-Layout

Belegung	Input-Daten
QWERTY	(11 24 22 13 3 14 15 16 8 17 18 19 26 25 9 10 1 4 12 5 7 23 2 21 20 6)
[Li et al., 2006]	(14 11 6 21 22 18 5 23 16 19 8 4 12 15 17 3 9 13 25 24 7 2 26 1 20 10)
[Leshner u. Moulton, 2000]	(15 18 3 21 14 25 12 22 4 19 11 6 17 13 16 8 10 5 24 23 7 2 20 9 26 1)
[Light u. Anderson, 1993]	(17 24 14 20 16 12 19 5 8 21 7 11 13 15 26 2 1 3 4 9 25 23 22 10 18 6)
Alphabet	(1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26)

Tabelle 27: Distanzen d_j zum vertikalen Startpunkt (5, 4) für das QWERTY-like-Layout

T1	4.53	T2	3.54	T3	2.55	T4	1.58	T5	0.71	T6	0.71	T7	1.58
T8	2.55	T9	3.54	T10	4.53	T11	4.27	T12	3.35	T13	2.5	T14	1.8
T15	1.5	T16	1.8	T17	2.5	T18	3.35	T19	4.27	T20	3.91	T21	3.2
T22	2.69	T23	2.5	T24	2.69	T25	3.2	T26	3.91				

Tabelle 28: Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das QWERTY-like-Layout

T1	2.55	T2	2.92	T3	3.54	T4	4.3	T5	5.15	T6	5.15	T7	4.3
T8	3.54	T9	2.92	T10	2.55	T11	1.8	T12	2.5	T13	3.35	T14	4.27
T15	5.22	T16	4.27	T17	3.35	T18	2.5	T19	1.8	T20	2.06	T21	3.04
T22	4.03	T23	5.02	T24	4.03	T25	3.04	T26	2.06				

Tabelle 29: Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das QWERTY-like-Layout

T1	2.5	T2	1.8	T3	1.5	T4	1.8	T5	2.5	T6	2.5	T7	1.8
T8	1.5	T9	1.8	T10	2.5	T11	1.58	T12	0.71	T13	0.71	T14	1.58
T15	2.55	T16	1.58	T17	0.71	T18	0.71	T19	1.58	T20	0.71	T21	0.71
T22	1.58	T23	2.55	T24	1.58	T25	0.71	T26	0.71				

Tabelle 30: Distanzen d_{jl} zwischen den Tasten für das vertikale QWERTY-like-Layout

T1-T1	0	T1-T2	1	T1-T3	2	T1-T4	3	T1-T5	4
T1-T6	5	T1-T7	6	T1-T8	7	T1-T9	8	T1-T10	9
T1-T11	1.12	T1-T12	1.8	T1-T13	2.69	T1-T14	3.64	T1-T15	4.61
T1-T16	5.59	T1-T17	6.58	T1-T18	7.57	T1-T19	8.56	T1-T20	2.5
T1-T21	3.2	T1-T22	4.03	T1-T23	4.92	T1-T24	5.85	T1-T25	6.8
T1-T26	7.76	T2-T1	1	T2-T2	0	T2-T3	1	T2-T4	2
T2-T5	3	T2-T6	4	T2-T7	5	T2-T8	6	T2-T9	7
T2-T10	8	T2-T11	1.12	T2-T12	1.12	T2-T13	1.8	T2-T14	2.69
T2-T15	3.64	T2-T16	4.61	T2-T17	5.59	T2-T18	6.58	T2-T19	7.57
T2-T20	2.06	T2-T21	2.5	T2-T22	3.2	T2-T23	4.03	T2-T24	4.92
T2-T25	5.85	T2-T26	6.8	T3-T1	2	T3-T2	1	T3-T3	0
T3-T4	1	T3-T5	2	T3-T6	3	T3-T7	4	T3-T8	5
T3-T9	6	T3-T10	7	T3-T11	1.8	T3-T12	1.12	T3-T13	1.12
T3-T14	1.8	T3-T15	2.69	T3-T16	3.64	T3-T17	4.61	T3-T18	5.59
T3-T19	6.58	T3-T20	2.06	T3-T21	2.06	T3-T22	2.5	T3-T23	3.2
T3-T24	4.03	T3-T25	4.92	T3-T26	5.85	T4-T1	3	T4-T2	2
T4-T3	1	T4-T4	0	T4-T5	1	T4-T6	2	T4-T7	3
T4-T8	4	T4-T9	5	T4-T10	6	T4-T11	2.69	T4-T12	1.8
T4-T13	1.12	T4-T14	1.12	T4-T15	1.8	T4-T16	2.69	T4-T17	3.64
T4-T18	4.61	T4-T19	5.59	T4-T20	2.5	T4-T21	2.06	T4-T22	2.06
T4-T23	2.5	T4-T24	3.2	T4-T25	4.03	T4-T26	4.92	T5-T1	4
T5-T2	3	T5-T3	2	T5-T4	1	T5-T5	0	T5-T6	1
T5-T7	2	T5-T8	3	T5-T9	4	T5-T10	5	T5-T11	3.64
T5-T12	2.69	T5-T13	1.8	T5-T14	1.12	T5-T15	1.12	T5-T16	1.8
T5-T17	2.69	T5-T18	3.64	T5-T19	4.61	T5-T20	3.2	T5-T21	2.5
T5-T22	2.06	T5-T23	2.06	T5-T24	2.5	T5-T25	3.2	T5-T26	4.03
T6-T1	5	T6-T2	4	T6-T3	3	T6-T4	2	T6-T5	1
T6-T6	0	T6-T7	1	T6-T8	2	T6-T9	3	T6-T10	4
T6-T11	4.61	T6-T12	3.64	T6-T13	2.69	T6-T14	1.8	T6-T15	1.12
T6-T16	1.12	T6-T17	1.8	T6-T18	2.69	T6-T19	3.64	T6-T20	4.03
T6-T21	3.2	T6-T22	2.5	T6-T23	2.06	T6-T24	2.06	T6-T25	2.5
T6-T26	3.2	T7-T1	6	T7-T2	5	T7-T3	4	T7-T4	3
T7-T5	2	T7-T6	1	T7-T7	0	T7-T8	1	T7-T9	2
T7-T10	3	T7-T11	5.59	T7-T12	4.61	T7-T13	3.64	T7-T14	2.69
T7-T15	1.8	T7-T16	1.12	T7-T17	1.12	T7-T18	1.8	T7-T19	2.69
T7-T20	4.92	T7-T21	4.03	T7-T22	3.2	T7-T23	2.5	T7-T24	2.06
T7-T25	2.06	T7-T26	2.5	T8-T1	7	T8-T2	6	T8-T3	5
T8-T4	4	T8-T5	3	T8-T6	2	T8-T7	1	T8-T8	0
T8-T9	1	T8-T10	2	T8-T11	6.58	T8-T12	5.59	T8-T13	4.61
T8-T14	3.64	T8-T15	2.69	T8-T16	1.8	T8-T17	1.12	T8-T18	1.12
T8-T19	1.8	T8-T20	5.85	T8-T21	4.92	T8-T22	4.03	T8-T23	3.2
T8-T24	2.5	T8-T25	2.06	T8-T26	2.06	T9-T1	8	T9-T2	7
T9-T3	6	T9-T4	5	T9-T5	4	T9-T6	3	T9-T7	2

A Input-Daten

T9-T8	1	T9-T9	0	T9-T10	1	T9-T11	7.57	T9-T12	6.58
T9-T13	5.59	T9-T14	4.61	T9-T15	3.64	T9-T16	2.69	T9-T17	1.8
T9-T18	1.12	T9-T19	1.12	T9-T20	6.8	T9-T21	5.85	T9-T22	4.92
T9-T23	4.03	T9-T24	3.2	T9-T25	2.5	T9-T26	2.06	T10-T1	9
T10-T2	8	T10-T3	7	T10-T4	6	T10-T5	5	T10-T6	4
T10-T7	3	T10-T8	2	T10-T9	1	T10-T10	0	T10-T11	8.56
T10-T12	7.57	T10-T13	6.58	T10-T14	5.59	T10-T15	4.61	T10-T16	3.64
T10-T17	2.69	T10-T18	1.8	T10-T19	1.12	T10-T20	7.76	T10-T21	6.8
T10-T22	5.85	T10-T23	4.92	T10-T24	4.03	T10-T25	3.2	T10-T26	2.5
T11-T1	1.12	T11-T2	1.12	T11-T3	1.8	T11-T4	2.69	T11-T5	3.64
T11-T6	4.61	T11-T7	5.59	T11-T8	6.58	T11-T9	7.57	T11-T10	8.56
T11-T11	0	T11-T12	1	T11-T13	2	T11-T14	3	T11-T15	4
T11-T16	5	T11-T17	6	T11-T18	7	T11-T19	8	T11-T20	1.41
T11-T21	7.76	T11-T22	6.8	T11-T23	5.85	T11-T24	4.92	T11-T25	4.03
T11-T26	3.2	T12-T1	1.8	T12-T2	1.12	T12-T3	1.12	T12-T4	1.8
T12-T5	2.69	T12-T6	3.64	T12-T7	4.61	T12-T8	5.59	T12-T9	6.58
T12-T10	7.57	T12-T11	1	T12-T12	0	T12-T13	1	T12-T14	2
T12-T15	3	T12-T16	4	T12-T17	5	T12-T18	6	T12-T19	7
T12-T20	1	T12-T21	1.41	T12-T22	7.76	T12-T23	6.8	T12-T24	5.85
T12-T25	4.92	T12-T26	4.03	T13-T1	2.69	T13-T2	1.8	T13-T3	1.12
T13-T4	1.12	T13-T5	1.8	T13-T6	2.69	T13-T7	3.64	T13-T8	4.61
T13-T9	5.59	T13-T10	6.58	T13-T11	2	T13-T12	1	T13-T13	0
T13-T14	1	T13-T15	2	T13-T16	3	T13-T17	4	T13-T18	5
T13-T19	6	T13-T20	1.41	T13-T21	1	T13-T22	1.41	T13-T23	7.76
T13-T24	6.8	T13-T25	5.85	T13-T26	4.92	T14-T1	3.64	T14-T2	2.69
T14-T3	1.8	T14-T4	1.12	T14-T5	1.12	T14-T6	1.8	T14-T7	2.69
T14-T8	3.64	T14-T9	4.61	T14-T10	5.59	T14-T11	3	T14-T12	2
T14-T13	1	T14-T14	0	T14-T15	1	T14-T16	2	T14-T17	3
T14-T18	4	T14-T19	5	T14-T20	2.24	T14-T21	1.41	T14-T22	1
T14-T23	1.41	T14-T24	7.76	T14-T25	6.8	T14-T26	5.85	T15-T1	4.61
T15-T2	3.64	T15-T3	2.69	T15-T4	1.8	T15-T5	1.12	T15-T6	1.12
T15-T7	1.8	T15-T8	2.69	T15-T9	3.64	T15-T10	4.61	T15-T11	4
T15-T12	3	T15-T13	2	T15-T14	1	T15-T15	0	T15-T16	1
T15-T17	2	T15-T18	3	T15-T19	4	T15-T20	3.16	T15-T21	2.24
T15-T22	1.41	T15-T23	1	T15-T24	1.41	T15-T25	7.76	T15-T26	6.8
T16-T1	5.59	T16-T2	4.61	T16-T3	3.64	T16-T4	2.69	T16-T5	1.8
T16-T6	1.12	T16-T7	1.12	T16-T8	1.8	T16-T9	2.69	T16-T10	3.64
T16-T11	5	T16-T12	4	T16-T13	3	T16-T14	2	T16-T15	1
T16-T16	0	T16-T17	1	T16-T18	2	T16-T19	3	T16-T20	4.12
T16-T21	3.16	T16-T22	2.24	T16-T23	1.41	T16-T24	1	T16-T25	1.41
T16-T26	7.76	T17-T1	6.58	T17-T2	5.59	T17-T3	4.61	T17-T4	3.64
T17-T5	2.69	T17-T6	1.8	T17-T7	1.12	T17-T8	1.12	T17-T9	1.8
T17-T10	2.69	T17-T11	6	T17-T12	5	T17-T13	4	T17-T14	3
T17-T15	2	T17-T16	1	T17-T17	0	T17-T18	1	T17-T19	2
T17-T20	5.1	T17-T21	4.12	T17-T22	3.16	T17-T23	2.24	T17-T24	1.41
T17-T25	1	T17-T26	1.41	T18-T1	7.57	T18-T2	6.58	T18-T3	5.59

T18-T4	4.61	T18-T5	3.64	T18-T6	2.69	T18-T7	1.8	T18-T8	1.12
T18-T9	1.12	T18-T10	1.8	T18-T11	7	T18-T12	6	T18-T13	5
T18-T14	4	T18-T15	3	T18-T16	2	T18-T17	1	T18-T18	0
T18-T19	1	T18-T20	6.08	T18-T21	5.1	T18-T22	4.12	T18-T23	3.16
T18-T24	2.24	T18-T25	1.41	T18-T26	1	T19-T1	8.56	T19-T2	7.57
T19-T3	6.58	T19-T4	5.59	T19-T5	4.61	T19-T6	3.64	T19-T7	2.69
T19-T8	1.8	T19-T9	1.12	T19-T10	1.12	T19-T11	8	T19-T12	7
T19-T13	6	T19-T14	5	T19-T15	4	T19-T16	3	T19-T17	2
T19-T18	1	T19-T19	0	T19-T20	7.07	T19-T21	6.08	T19-T22	5.1
T19-T23	4.12	T19-T24	3.16	T19-T25	2.24	T19-T26	1.41	T20-T1	2.5
T20-T2	2.06	T20-T3	2.06	T20-T4	2.5	T20-T5	3.2	T20-T6	4.03
T20-T7	4.92	T20-T8	5.85	T20-T9	6.8	T20-T10	7.76	T20-T11	1.41
T20-T12	1	T20-T13	1.41	T20-T14	2.24	T20-T15	3.16	T20-T16	4.12
T20-T17	5.1	T20-T18	6.08	T20-T19	7.07	T20-T20	0	T20-T21	1
T20-T22	2	T20-T23	3	T20-T24	4	T20-T25	5	T20-T26	6
T21-T1	3.2	T21-T2	2.5	T21-T3	2.06	T21-T4	2.06	T21-T5	2.5
T21-T6	3.2	T21-T7	4.03	T21-T8	4.92	T21-T9	5.85	T21-T10	6.8
T21-T11	7.76	T21-T12	1.41	T21-T13	1	T21-T14	1.41	T21-T15	2.24
T21-T16	3.16	T21-T17	4.12	T21-T18	5.1	T21-T19	6.08	T21-T20	1
T21-T21	0	T21-T22	1	T21-T23	2	T21-T24	3	T21-T25	4
T21-T26	5	T22-T1	4.03	T22-T2	3.2	T22-T3	2.5	T22-T4	2.06
T22-T5	2.06	T22-T6	2.5	T22-T7	3.2	T22-T8	4.03	T22-T9	4.92
T22-T10	5.85	T22-T11	6.8	T22-T12	7.76	T22-T13	1.41	T22-T14	1
T22-T15	1.41	T22-T16	2.24	T22-T17	3.16	T22-T18	4.12	T22-T19	5.1
T22-T20	2	T22-T21	1	T22-T22	0	T22-T23	1	T22-T24	2
T22-T25	3	T22-T26	4	T23-T1	4.92	T23-T2	4.03	T23-T3	3.2
T23-T4	2.5	T23-T5	2.06	T23-T6	2.06	T23-T7	2.5	T23-T8	3.2
T23-T9	4.03	T23-T10	4.92	T23-T11	5.85	T23-T12	6.8	T23-T13	7.76
T23-T14	1.41	T23-T15	1	T23-T16	1.41	T23-T17	2.24	T23-T18	3.16
T23-T19	4.12	T23-T20	3	T23-T21	2	T23-T22	1	T23-T23	0
T23-T24	1	T23-T25	2	T23-T26	3	T24-T1	5.85	T24-T2	4.92
T24-T3	4.03	T24-T4	3.2	T24-T5	2.5	T24-T6	2.06	T24-T7	2.06
T24-T8	2.5	T24-T9	3.2	T24-T10	4.03	T24-T11	4.92	T24-T12	5.85
T24-T13	6.8	T24-T14	7.76	T24-T15	1.41	T24-T16	1	T24-T17	1.41
T24-T18	2.24	T24-T19	3.16	T24-T20	4	T24-T21	3	T24-T22	2
T24-T23	1	T24-T24	0	T24-T25	1	T24-T26	2	T25-T1	6.8
T25-T2	5.85	T25-T3	4.92	T25-T4	4.03	T25-T5	3.2	T25-T6	2.5
T25-T7	2.06	T25-T8	2.06	T25-T9	2.5	T25-T10	3.2	T25-T11	4.03
T25-T12	4.92	T25-T13	5.85	T25-T14	6.8	T25-T15	7.76	T25-T16	1.41
T25-T17	1	T25-T18	1.41	T25-T19	2.24	T25-T20	5	T25-T21	4
T25-T22	3	T25-T23	2	T25-T24	1	T25-T25	0	T25-T26	1
T26-T1	7.76	T26-T2	6.8	T26-T3	5.85	T26-T4	4.92	T26-T5	4.03
T26-T6	3.2	T26-T7	2.5	T26-T8	2.06	T26-T9	2.06	T26-T10	2.5
T26-T11	3.2	T26-T12	4.03	T26-T13	4.92	T26-T14	5.85	T26-T15	6.8
T26-T16	7.76	T26-T17	1.41	T26-T18	1	T26-T19	1.41	T26-T20	6
T26-T21	5	T26-T22	4	T26-T23	3	T26-T24	2	T26-T25	1
T26-T26	0								

A.2.2 Layout 2 - 3x9-Layout

- numberofkeys = 27, Rows = 3, Columns = 9

Tabelle 31: Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 3x9-Layout

T1	4.03	T2	3.04	T3	2.06	T4	1.12	T5	0.5	T6	1.12	T7	2.06
T8	3.04	T9	4.03	T10	4.27	T11	3.35	T12	2.5	T13	1.8	T14	1.5
T15	1.8	T16	2.5	T17	3.35	T18	4.27	T19	4.72	T20	3.91	T21	3.2
T22	2.69	T23	2.5	T24	2.69	T25	3.2	T26	3.91	T27	4.72		

Tabelle 32: Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 3x9-Layout

T1	2.69	T2	3.2	T3	3.91	T4	4.72	T5	5.59	T6	4.72	T7	3.91
T8	3.2	T9	2.69	T10	1.8	T11	2.5	T12	3.35	T13	4.27	T14	5.22
T15	4.27	T16	3.35	T17	2.5	T18	1.8	T19	1.12	T20	2.06	T21	3.04
T22	4.03	T23	5.02	T24	4.03	T25	3.04	T26	2.06	T27	1.12		

Tabelle 33: Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 3x9-Layout

T1	2.12	T2	1.58	T3	1.58	T4	2.12	T5	2.92	T6	2.12	T7	1.58
T8	1.58	T9	2.12	T10	1.58	T11	0.71	T12	0.71	T13	1.58	T14	2.55
T15	1.58	T16	0.71	T17	0.71	T18	1.58	T19	1.58	T20	0.71	T21	0.71
T22	1.58	T23	2.55	T24	1.58	T25	0.71	T26	0.71	T27	1.58		

Tabelle 34: Distanzen d_{ji} zwischen den Tasten für das vertikale 3x9-Layout

T1-T1	0	T1-T2	1	T1-T3	2	T1-T4	3	T1-T5	4
T1-T6	5	T1-T7	6	T1-T8	7	T1-T9	8	T1-T10	1
T1-T11	1.41	T1-T12	2.24	T1-T13	3.16	T1-T14	4.12	T1-T15	5.1
T1-T16	6.08	T1-T17	7.07	T1-T18	8.06	T1-T19	2	T1-T20	2.24
T1-T21	2.83	T1-T22	3.61	T1-T23	4.47	T1-T24	5.39	T1-T25	6.32
T1-T26	7.28	T1-T27	8.25	T2-T1	1	T2-T2	0	T2-T3	1
T2-T4	2	T2-T5	3	T2-T6	4	T2-T7	5	T2-T8	6
T2-T9	7	T2-T10	1.41	T2-T11	1	T2-T12	1.41	T2-T13	2.24
T2-T14	3.16	T2-T15	4.12	T2-T16	5.1	T2-T17	6.08	T2-T18	7.07
T2-T19	2.24	T2-T20	2	T2-T21	2.24	T2-T22	2.83	T2-T23	3.61
T2-T24	4.47	T2-T25	5.39	T2-T26	6.32	T2-T27	7.28	T3-T1	2

T3-T2	1	T3-T3	0	T3-T4	1	T3-T5	2	T3-T6	3
T3-T7	4	T3-T8	5	T3-T9	6	T3-T10	2.24	T3-T11	1.41
T3-T12	1	T3-T13	1.41	T3-T14	2.24	T3-T15	3.16	T3-T16	4.12
T3-T17	5.1	T3-T18	6.08	T3-T19	2.83	T3-T20	2.24	T3-T21	2
T3-T22	2.24	T3-T23	2.83	T3-T24	3.61	T3-T25	4.47	T3-T26	5.39
T3-T27	6.32	T4-T1	3	T4-T2	2	T4-T3	1	T4-T4	0
T4-T5	1	T4-T6	2	T4-T7	3	T4-T8	4	T4-T9	5
T4-T10	3.16	T4-T11	2.24	T4-T12	1.41	T4-T13	1	T4-T14	1.41
T4-T15	2.24	T4-T16	3.16	T4-T17	4.12	T4-T18	5.1	T4-T19	3.61
T4-T20	2.83	T4-T21	2.24	T4-T22	2	T4-T23	2.24	T4-T24	2.83
T4-T25	3.61	T4-T26	4.47	T4-T27	5.39	T5-T1	4	T5-T2	3
T5-T3	2	T5-T4	1	T5-T5	0	T5-T6	1	T5-T7	2
T5-T8	3	T5-T9	4	T5-T10	4.12	T5-T11	3.16	T5-T12	2.24
T5-T13	1.41	T5-T14	1	T5-T15	1.41	T5-T16	2.24	T5-T17	3.16
T5-T18	4.12	T5-T19	4.47	T5-T20	3.61	T5-T21	2.83	T5-T22	2.24
T5-T23	2	T5-T24	2.24	T5-T25	2.83	T5-T26	3.61	T5-T27	4.47
T6-T1	5	T6-T2	4	T6-T3	3	T6-T4	2	T6-T5	1
T6-T6	0	T6-T7	1	T6-T8	2	T6-T9	3	T6-T10	5.1
T6-T11	4.12	T6-T12	3.16	T6-T13	2.24	T6-T14	1.41	T6-T15	1
T6-T16	1.41	T6-T17	2.24	T6-T18	3.16	T6-T19	5.39	T6-T20	4.47
T6-T21	3.61	T6-T22	2.83	T6-T23	2.24	T6-T24	2	T6-T25	2.24
T6-T26	2.83	T6-T27	3.61	T7-T1	6	T7-T2	5	T7-T3	4
T7-T4	3	T7-T5	2	T7-T6	1	T7-T7	0	T7-T8	1
T7-T9	2	T7-T10	6.08	T7-T11	5.1	T7-T12	4.12	T7-T13	3.16
T7-T14	2.24	T7-T15	1.41	T7-T16	1	T7-T17	1.41	T7-T18	2.24
T7-T19	6.32	T7-T20	5.39	T7-T21	4.47	T7-T22	3.61	T7-T23	2.83
T7-T24	2.24	T7-T25	2	T7-T26	2.24	T7-T27	2.83	T8-T1	7
T8-T2	6	T8-T3	5	T8-T4	4	T8-T5	3	T8-T6	2
T8-T7	1	T8-T8	0	T8-T9	1	T8-T10	7.07	T8-T11	6.08
T8-T12	5.1	T8-T13	4.12	T8-T14	3.16	T8-T15	2.24	T8-T16	1.41
T8-T17	1	T8-T18	1.41	T8-T19	7.28	T8-T20	6.32	T8-T21	5.39
T8-T22	4.47	T8-T23	3.61	T8-T24	2.83	T8-T25	2.24	T8-T26	2
T8-T27	2.24	T9-T1	8	T9-T2	7	T9-T3	6	T9-T4	5
T9-T5	4	T9-T6	3	T9-T7	2	T9-T8	1	T9-T9	0
T9-T10	8.06	T9-T11	7.07	T9-T12	6.08	T9-T13	5.1	T9-T14	4.12
T9-T15	3.16	T9-T16	2.24	T9-T17	1.41	T9-T18	1	T9-T19	8.25
T9-T20	7.28	T9-T21	6.32	T9-T22	5.39	T9-T23	4.47	T9-T24	3.61
T9-T25	2.83	T9-T26	2.24	T9-T27	2	T10-T1	1	T10-T2	1.41
T10-T3	2.24	T10-T4	3.16	T10-T5	4.12	T10-T6	5.1	T10-T7	6.08
T10-T8	7.07	T10-T9	8.06	T10-T10	0	T10-T11	1	T10-T12	2
T10-T13	3	T10-T14	4	T10-T15	5	T10-T16	6	T10-T17	7
T10-T18	8	T10-T19	1	T10-T20	1.41	T10-T21	2.24	T10-T22	3.16
T10-T23	4.12	T10-T24	5.1	T10-T25	6.08	T10-T26	7.07	T10-T27	8.06
T11-T1	1.41	T11-T2	1	T11-T3	1.41	T11-T4	2.24	T11-T5	3.16
T11-T6	4.12	T11-T7	5.1	T11-T8	6.08	T11-T9	7.07	T11-T10	1

A Input-Daten

T11-T11	0	T11-T12	1	T11-T13	2	T11-T14	3	T11-T15	4
T11-T16	5	T11-T17	6	T11-T18	7	T11-T19	1.41	T11-T20	1
T11-T21	1.41	T11-T22	2.24	T11-T23	3.16	T11-T24	4.12	T11-T25	5.1
T11-T26	6.08	T11-T27	7.07	T12-T1	2.24	T12-T2	1.41	T12-T3	1
T12-T4	1.41	T12-T5	2.24	T12-T6	3.16	T12-T7	4.12	T12-T8	5.1
T12-T9	6.08	T12-T10	2	T12-T11	1	T12-T12	0	T12-T13	1
T12-T14	2	T12-T15	3	T12-T16	4	T12-T17	5	T12-T18	6
T12-T19	2.24	T12-T20	1.41	T12-T21	1	T12-T22	1.41	T12-T23	2.24
T12-T24	3.16	T12-T25	4.12	T12-T26	5.1	T12-T27	6.08	T13-T1	3.16
T13-T2	2.24	T13-T3	1.41	T13-T4	1	T13-T5	1.41	T13-T6	2.24
T13-T7	3.16	T13-T8	4.12	T13-T9	5.1	T13-T10	3	T13-T11	2
T13-T12	1	T13-T13	0	T13-T14	1	T13-T15	2	T13-T16	3
T13-T17	4	T13-T18	5	T13-T19	3.16	T13-T20	2.24	T13-T21	1.41
T13-T22	1	T13-T23	1.41	T13-T24	2.24	T13-T25	3.16	T13-T26	4.12
T13-T27	5.1	T14-T1	4.12	T14-T2	3.16	T14-T3	2.24	T14-T4	1.41
T14-T5	1	T14-T6	1.41	T14-T7	2.24	T14-T8	3.16	T14-T9	4.12
T14-T10	4	T14-T11	3	T14-T12	2	T14-T13	1	T14-T14	0
T14-T15	1	T14-T16	2	T14-T17	3	T14-T18	4	T14-T19	4.12
T14-T20	3.16	T14-T21	2.24	T14-T22	1.41	T14-T23	1	T14-T24	1.41
T14-T25	2.24	T14-T26	3.16	T14-T27	4.12	T15-T1	5.1	T15-T2	4.12
T15-T3	3.16	T15-T4	2.24	T15-T5	1.41	T15-T6	1	T15-T7	1.41
T15-T8	2.24	T15-T9	3.16	T15-T10	5	T15-T11	4	T15-T12	3
T15-T13	2	T15-T14	1	T15-T15	0	T15-T16	1	T15-T17	2
T15-T18	3	T15-T19	5.1	T15-T20	4.12	T15-T21	3.16	T15-T22	2.24
T15-T23	1.41	T15-T24	1	T15-T25	1.41	T15-T26	2.24	T15-T27	3.16
T16-T1	6.08	T16-T2	5.1	T16-T3	4.12	T16-T4	3.16	T16-T5	2.24
T16-T6	1.41	T16-T7	1	T16-T8	1.41	T16-T9	2.24	T16-T10	6
T16-T11	5	T16-T12	4	T16-T13	3	T16-T14	2	T16-T15	1
T16-T16	0	T16-T17	1	T16-T18	2	T16-T19	6.08	T16-T20	5.1
T16-T21	4.12	T16-T22	3.16	T16-T23	2.24	T16-T24	1.41	T16-T25	1
T16-T26	1.41	T16-T27	2.24	T17-T1	7.07	T17-T2	6.08	T17-T3	5.1
T17-T4	4.12	T17-T5	3.16	T17-T6	2.24	T17-T7	1.41	T17-T8	1
T17-T9	1.41	T17-T10	7	T17-T11	6	T17-T12	5	T17-T13	4
T17-T14	3	T17-T15	2	T17-T16	1	T17-T17	0	T17-T18	1
T17-T19	7.07	T17-T20	6.08	T17-T21	5.1	T17-T22	4.12	T17-T23	3.16
T17-T24	2.24	T17-T25	1.41	T17-T26	1	T17-T27	1.41	T18-T1	8.06
T18-T2	7.07	T18-T3	6.08	T18-T4	5.1	T18-T5	4.12	T18-T6	3.16
T18-T7	2.24	T18-T8	1.41	T18-T9	1	T18-T10	8	T18-T11	7
T18-T12	6	T18-T13	5	T18-T14	4	T18-T15	3	T18-T16	2
T18-T17	1	T18-T18	0	T18-T19	8.06	T18-T20	7.07	T18-T21	6.08
T18-T22	5.1	T18-T23	4.12	T18-T24	3.16	T18-T25	2.24	T18-T26	1.41
T18-T27	1	T19-T1	2	T19-T2	2.24	T19-T3	2.83	T19-T4	3.61
T19-T5	4.47	T19-T6	5.39	T19-T7	6.32	T19-T8	7.28	T19-T9	8.25
T19-T10	1	T19-T11	1.41	T19-T12	2.24	T19-T13	3.16	T19-T14	4.12
T19-T15	5.1	T19-T16	6.08	T19-T17	7.07	T19-T18	8.06	T19-T19	0

T19-T20	1	T19-T21	2	T19-T22	3	T19-T23	4	T19-T24	5
T19-T25	6	T19-T26	7	T19-T27	8	T20-T1	2.24	T20-T2	2
T20-T3	2.24	T20-T4	2.83	T20-T5	3.61	T20-T6	4.47	T20-T7	5.39
T20-T8	6.32	T20-T9	7.28	T20-T10	1.41	T20-T11	1	T20-T12	1.41
T20-T13	2.24	T20-T14	3.16	T20-T15	4.12	T20-T16	5.1	T20-T17	6.08
T20-T18	7.07	T20-T19	1	T20-T20	0	T20-T21	1	T20-T22	2
T20-T23	3	T20-T24	4	T20-T25	5	T20-T26	6	T20-T27	7
T21-T1	2.83	T21-T2	2.24	T21-T3	2	T21-T4	2.24	T21-T5	2.83
T21-T6	3.61	T21-T7	4.47	T21-T8	5.39	T21-T9	6.32	T21-T10	2.24
T21-T11	1.41	T21-T12	1	T21-T13	1.41	T21-T14	2.24	T21-T15	3.16
T21-T16	4.12	T21-T17	5.1	T21-T18	6.08	T21-T19	2	T21-T20	1
T21-T21	0	T21-T22	1	T21-T23	2	T21-T24	3	T21-T25	4
T21-T26	5	T21-T27	6	T22-T1	3.61	T22-T2	2.83	T22-T3	2.24
T22-T4	2	T22-T5	2.24	T22-T6	2.83	T22-T7	3.61	T22-T8	4.47
T22-T9	5.39	T22-T10	3.16	T22-T11	2.24	T22-T12	1.41	T22-T13	1
T22-T14	1.41	T22-T15	2.24	T22-T16	3.16	T22-T17	4.12	T22-T18	5.1
T22-T19	3	T22-T20	2	T22-T21	1	T22-T22	0	T22-T23	1
T22-T24	2	T22-T25	3	T22-T26	4	T22-T27	5	T23-T1	4.47
T23-T2	3.61	T23-T3	2.83	T23-T4	2.24	T23-T5	2	T23-T6	2.24
T23-T7	2.83	T23-T8	3.61	T23-T9	4.47	T23-T10	4.12	T23-T11	3.16
T23-T12	2.24	T23-T13	1.41	T23-T14	1	T23-T15	1.41	T23-T16	2.24
T23-T17	3.16	T23-T18	4.12	T23-T19	4	T23-T20	3	T23-T21	2
T23-T22	1	T23-T23	0	T23-T24	1	T23-T25	2	T23-T26	3
T23-T27	4	T24-T1	5.39	T24-T2	4.47	T24-T3	3.61	T24-T4	2.83
T24-T5	2.24	T24-T6	2	T24-T7	2.24	T24-T8	2.83	T24-T9	3.61
T24-T10	5.1	T24-T11	4.12	T24-T12	3.16	T24-T13	2.24	T24-T14	1.41
T24-T15	1	T24-T16	1.41	T24-T17	2.24	T24-T18	3.16	T24-T19	5
T24-T20	4	T24-T21	3	T24-T22	2	T24-T23	1	T24-T24	0
T24-T25	1	T24-T26	2	T24-T27	3	T25-T1	6.32	T25-T2	5.39
T25-T3	4.47	T25-T4	3.61	T25-T5	2.83	T25-T6	2.24	T25-T7	2
T25-T8	2.24	T25-T9	2.83	T25-T10	6.08	T25-T11	5.1	T25-T12	4.12
T25-T13	3.16	T25-T14	2.24	T25-T15	1.41	T25-T16	1	T25-T17	1.41
T25-T18	2.24	T25-T19	6	T25-T20	5	T25-T21	4	T25-T22	3
T25-T23	2	T25-T24	1	T25-T25	0	T25-T26	1	T25-T27	2
T26-T1	7.28	T26-T2	6.32	T26-T3	5.39	T26-T4	4.47	T26-T5	3.61
T26-T6	2.83	T26-T7	2.24	T26-T8	2	T26-T9	2.24	T26-T10	7.07
T26-T11	6.08	T26-T12	5.1	T26-T13	4.12	T26-T14	3.16	T26-T15	2.24
T26-T16	1.41	T26-T17	1	T26-T18	1.41	T26-T19	7	T26-T20	6
T26-T21	5	T26-T22	4	T26-T23	3	T26-T24	2	T26-T25	1
T26-T26	0	T26-T27	1	T27-T1	8.25	T27-T2	7.28	T27-T3	6.32
T27-T4	5.39	T27-T5	4.47	T27-T6	3.61	T27-T7	2.83	T27-T8	2.24
T27-T9	2	T27-T10	8.06	T27-T11	7.07	T27-T12	6.08	T27-T13	5.1
T27-T14	4.12	T27-T15	3.16	T27-T16	2.24	T27-T17	1.41	T27-T18	1
T27-T19	8	T27-T20	7	T27-T21	6	T27-T22	5	T27-T23	4
T27-T24	3	T27-T25	2	T27-T26	1	T27-T27	0		

A.2.3 Layout 3 - 3x10-Layout

- numberofkeys = 30, Rows = 3, Columns = 10

Tabelle 35: Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 3x10-Layout

T1	4.53	T2	3.54	T3	2.55	T4	1.58	T5	0.71	T6	0.71	T7	1.58
T8	2.55	T9	3.54	T10	4.53	T11	4.74	T12	3.81	T13	2.92	T14	2.12
T15	1.58	T16	1.58	T17	2.12	T18	2.92	T19	3.81	T20	4.74	T21	5.15
T22	4.3	T23	3.54	T24	2.92	T25	2.55	T26	2.55	T27	2.92	T28	3.54
T29	4.3	T30	5.15										

Tabelle 36: Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 3x10-Layout

T1	2.55	T2	2.92	T3	3.54	T4	4.3	T5	5.15	T6	5.15	T7	4.3
T8	3.54	T9	2.92	T10	2.55	T11	1.58	T12	2.12	T13	2.92	T14	3.81
T15	4.74	T16	4.74	T17	3.81	T18	2.92	T19	2.12	T20	1.58	T21	0.71
T22	1.58	T23	2.55	T24	3.54	T25	4.53	T26	4.53	T27	3.54	T28	2.55
T29	1.58	T30	0.71										

Tabelle 37: Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 3x10-Layout

T1	2.5	T2	1.8	T3	1.5	T4	1.8	T5	2.5	T6	2.5	T7	1.8
T8	1.5	T9	1.8	T10	2.5	T11	2.06	T12	1.12	T13	0.5	T14	1.12
T15	2.06	T16	2.06	T17	1.12	T18	0.5	T19	1.12	T20	2.06	T21	2.06
T22	1.12	T23	0.5	T24	1.12	T25	2.06	T26	2.06	T27	1.12	T28	0.5
T29	1.12	T30	2.06										

Tabelle 38: Distanzen d_{jl} zwischen den Tasten für das vertikale 3x10-Layout

T1-T1	0	T1-T2	1	T1-T3	2	T1-T4	3	T1-T5	4
T1-T6	5	T1-T7	6	T1-T8	7	T1-T9	8	T1-T10	9
T1-T11	1	T1-T12	1.41	T1-T13	2.24	T1-T14	3.16	T1-T15	4.12
T1-T16	5.1	T1-T17	6.08	T1-T18	7.07	T1-T19	8.06	T1-T20	9.06
T1-T21	2	T1-T22	2.24	T1-T23	2.83	T1-T24	3.61	T1-T25	4.47
T1-T26	5.39	T1-T27	6.32	T1-T28	7.28	T1-T29	8.25	T1-T30	9.22
T2-T1	1	T2-T2	0	T2-T3	1	T2-T4	2	T2-T5	3
T2-T6	4	T2-T7	5	T2-T8	6	T2-T9	7	T2-T10	8
T2-T11	1.41	T2-T12	1	T2-T13	1.41	T2-T14	2.24	T2-T15	3.16

T2-T16	4.12	T2-T17	5.1	T2-T18	6.08	T2-T19	7.07	T2-T20	8.06
T2-T21	2.24	T2-T22	2	T2-T23	2.24	T2-T24	2.83	T2-T25	3.61
T2-T26	4.47	T2-T27	5.39	T2-T28	6.32	T2-T29	7.28	T2-T30	8.25
T3-T1	2	T3-T2	1	T3-T3	0	T3-T4	1	T3-T5	2
T3-T6	3	T3-T7	4	T3-T8	5	T3-T9	6	T3-T10	7
T3-T11	2.24	T3-T12	1.41	T3-T13	1	T3-T14	1.41	T3-T15	2.24
T3-T16	3.16	T3-T17	4.12	T3-T18	5.1	T3-T19	6.08	T3-T20	7.07
T3-T21	2.83	T3-T22	2.24	T3-T23	2	T3-T24	2.24	T3-T25	2.83
T3-T26	3.61	T3-T27	4.47	T3-T28	5.39	T3-T29	6.32	T3-T30	7.28
T4-T1	3	T4-T2	2	T4-T3	1	T4-T4	0	T4-T5	1
T4-T6	2	T4-T7	3	T4-T8	4	T4-T9	5	T4-T10	6
T4-T11	3.16	T4-T12	2.24	T4-T13	1.41	T4-T14	1	T4-T15	1.41
T4-T16	2.24	T4-T17	3.16	T4-T18	4.12	T4-T19	5.1	T4-T20	6.08
T4-T21	3.61	T4-T22	2.83	T4-T23	2.24	T4-T24	2	T4-T25	2.24
T4-T26	2.83	T4-T27	3.61	T4-T28	4.47	T4-T29	5.39	T4-T30	6.32
T5-T1	4	T5-T2	3	T5-T3	2	T5-T4	1	T5-T5	0
T5-T6	1	T5-T7	2	T5-T8	3	T5-T9	4	T5-T10	5
T5-T11	4.12	T5-T12	3.16	T5-T13	2.24	T5-T14	1.41	T5-T15	1
T5-T16	1.41	T5-T17	2.24	T5-T18	3.16	T5-T19	4.12	T5-T20	5.1
T5-T21	4.47	T5-T22	3.61	T5-T23	2.83	T5-T24	2.24	T5-T25	2
T5-T26	2.24	T5-T27	2.83	T5-T28	3.61	T5-T29	4.47	T5-T30	5.39
T6-T1	5	T6-T2	4	T6-T3	3	T6-T4	2	T6-T5	1
T6-T6	0	T6-T7	1	T6-T8	2	T6-T9	3	T6-T10	4
T6-T11	5.1	T6-T12	4.12	T6-T13	3.16	T6-T14	2.24	T6-T15	1.41
T6-T16	1	T6-T17	1.41	T6-T18	2.24	T6-T19	3.16	T6-T20	4.12
T6-T21	5.39	T6-T22	4.47	T6-T23	3.61	T6-T24	2.83	T6-T25	2.24
T6-T26	2	T6-T27	2.24	T6-T28	2.83	T6-T29	3.61	T6-T30	4.47
T7-T1	6	T7-T2	5	T7-T3	4	T7-T4	3	T7-T5	2
T7-T6	1	T7-T7	0	T7-T8	1	T7-T9	2	T7-T10	3
T7-T11	6.08	T7-T12	5.1	T7-T13	4.12	T7-T14	3.16	T7-T15	2.24
T7-T16	1.41	T7-T17	1	T7-T18	1.41	T7-T19	2.24	T7-T20	3.16
T7-T21	6.32	T7-T22	5.39	T7-T23	4.47	T7-T24	3.61	T7-T25	2.83
T7-T26	2.24	T7-T27	2	T7-T28	2.24	T7-T29	2.83	T7-T30	3.61
T8-T1	7	T8-T2	6	T8-T3	5	T8-T4	4	T8-T5	3
T8-T6	2	T8-T7	1	T8-T8	0	T8-T9	1	T8-T10	2
T8-T11	7.07	T8-T12	6.08	T8-T13	5.1	T8-T14	4.12	T8-T15	3.16
T8-T16	2.24	T8-T17	1.41	T8-T18	1	T8-T19	1.41	T8-T20	2.24
T8-T21	7.28	T8-T22	6.32	T8-T23	5.39	T8-T24	4.47	T8-T25	3.61
T8-T26	2.83	T8-T27	2.24	T8-T28	2	T8-T29	2.24	T8-T30	2.83
T9-T1	8	T9-T2	7	T9-T3	6	T9-T4	5	T9-T5	4
T9-T6	3	T9-T7	2	T9-T8	1	T9-T9	0	T9-T10	1
T9-T11	8.06	T9-T12	7.07	T9-T13	6.08	T9-T14	5.1	T9-T15	4.12
T9-T16	3.16	T9-T17	2.24	T9-T18	1.41	T9-T19	1	T9-T20	1.41
T9-T21	8.25	T9-T22	7.28	T9-T23	6.32	T9-T24	5.39	T9-T25	4.47
T9-T26	3.61	T9-T27	2.83	T9-T28	2.24	T9-T29	2	T9-T30	2.24
T10-T1	9	T10-T2	8	T10-T3	7	T10-T4	6	T10-T5	5
T10-T6	4	T10-T7	3	T10-T8	2	T10-T9	1	T10-T10	0

A Input-Daten

T10-T11	9.06	T10-T12	8.06	T10-T13	7.07	T10-T14	6.08	T10-T15	5.1
T10-T16	4.12	T10-T17	3.16	T10-T18	2.24	T10-T19	1.41	T10-T20	1
T10-T21	9.22	T10-T22	8.25	T10-T23	7.28	T10-T24	6.32	T10-T25	5.39
T10-T26	4.47	T10-T27	3.61	T10-T28	2.83	T10-T29	2.24	T10-T30	2
T11-T1	1	T11-T2	1.41	T11-T3	2.24	T11-T4	3.16	T11-T5	4.12
T11-T6	5.1	T11-T7	6.08	T11-T8	7.07	T11-T9	8.06	T11-T10	9.06
T11-T11	0	T11-T12	1	T11-T13	2	T11-T14	3	T11-T15	4
T11-T16	5	T11-T17	6	T11-T18	7	T11-T19	8	T11-T20	9
T11-T21	1	T11-T22	1.41	T11-T23	2.24	T11-T24	3.16	T11-T25	4.12
T11-T26	5.1	T11-T27	6.08	T11-T28	7.07	T11-T29	8.06	T11-T30	9.06
T12-T1	1.41	T12-T2	1	T12-T3	1.41	T12-T4	2.24	T12-T5	3.16
T12-T6	4.12	T12-T7	5.1	T12-T8	6.08	T12-T9	7.07	T12-T10	8.06
T12-T11	1	T12-T12	0	T12-T13	1	T12-T14	2	T12-T15	3
T12-T16	4	T12-T17	5	T12-T18	6	T12-T19	7	T12-T20	8
T12-T21	1.41	T12-T22	1	T12-T23	1.41	T12-T24	2.24	T12-T25	3.16
T12-T26	4.12	T12-T27	5.1	T12-T28	6.08	T12-T29	7.07	T12-T30	8.06
T13-T1	2.24	T13-T2	1.41	T13-T3	1	T13-T4	1.41	T13-T5	2.24
T13-T6	3.16	T13-T7	4.12	T13-T8	5.1	T13-T9	6.08	T13-T10	7.07
T13-T11	2	T13-T12	1	T13-T13	0	T13-T14	1	T13-T15	2
T13-T16	3	T13-T17	4	T13-T18	5	T13-T19	6	T13-T20	7
T13-T21	2.24	T13-T22	1.41	T13-T23	1	T13-T24	1.41	T13-T25	2.24
T13-T26	3.16	T13-T27	4.12	T13-T28	5.1	T13-T29	6.08	T13-T30	7.07
T14-T1	3.16	T14-T2	2.24	T14-T3	1.41	T14-T4	1	T14-T5	1.41
T14-T6	2.24	T14-T7	3.16	T14-T8	4.12	T14-T9	5.1	T14-T10	6.08
T14-T11	3	T14-T12	2	T14-T13	1	T14-T14	0	T14-T15	1
T14-T16	2	T14-T17	3	T14-T18	4	T14-T19	5	T14-T20	6
T14-T21	3.16	T14-T22	2.24	T14-T23	1.41	T14-T24	1	T14-T25	1.41
T14-T26	2.24	T14-T27	3.16	T14-T28	4.12	T14-T29	5.1	T14-T30	6.08
T15-T1	4.12	T15-T2	3.16	T15-T3	2.24	T15-T4	1.41	T15-T5	1
T15-T6	1.41	T15-T7	2.24	T15-T8	3.16	T15-T9	4.12	T15-T10	5.1
T15-T11	4	T15-T12	3	T15-T13	2	T15-T14	1	T15-T15	0
T15-T16	1	T15-T17	2	T15-T18	3	T15-T19	4	T15-T20	5
T15-T21	4.12	T15-T22	3.16	T15-T23	2.24	T15-T24	1.41	T15-T25	1
T15-T26	1.41	T15-T27	2.24	T15-T28	3.16	T15-T29	4.12	T15-T30	5.1
T16-T1	5.1	T16-T2	4.12	T16-T3	3.16	T16-T4	2.24	T16-T5	1.41
T16-T6	1	T16-T7	1.41	T16-T8	2.24	T16-T9	3.16	T16-T10	4.12
T16-T11	5	T16-T12	4	T16-T13	3	T16-T14	2	T16-T15	1
T16-T16	0	T16-T17	1	T16-T18	2	T16-T19	3	T16-T20	4
T16-T21	5.1	T16-T22	4.12	T16-T23	3.16	T16-T24	2.24	T16-T25	1.41
T16-T26	1	T16-T27	1.41	T16-T28	2.24	T16-T29	3.16	T16-T30	4.12
T17-T1	6.08	T17-T2	5.1	T17-T3	4.12	T17-T4	3.16	T17-T5	2.24
T17-T6	1.41	T17-T7	1	T17-T8	1.41	T17-T9	2.24	T17-T10	3.16
T17-T11	6	T17-T12	5	T17-T13	4	T17-T14	3	T17-T15	2
T17-T16	1	T17-T17	0	T17-T18	1	T17-T19	2	T17-T20	3

T17-T21	6.08	T17-T22	5.1	T17-T23	4.12	T17-T24	3.16	T17-T25	2.24
T17-T26	1.41	T17-T27	1	T17-T28	1.41	T17-T29	2.24	T17-T30	3.16
T18-T1	7.07	T18-T2	6.08	T18-T3	5.1	T18-T4	4.12	T18-T5	3.16
T18-T6	2.24	T18-T7	1.41	T18-T8	1	T18-T9	1.41	T18-T10	2.24
T18-T11	7	T18-T12	6	T18-T13	5	T18-T14	4	T18-T15	3
T18-T16	2	T18-T17	1	T18-T18	0	T18-T19	1	T18-T20	2
T18-T21	7.07	T18-T22	6.08	T18-T23	5.1	T18-T24	4.12	T18-T25	3.16
T18-T26	2.24	T18-T27	1.41	T18-T28	1	T18-T29	1.41	T18-T30	2.24
T19-T1	8.06	T19-T2	7.07	T19-T3	6.08	T19-T4	5.1	T19-T5	4.12
T19-T6	3.16	T19-T7	2.24	T19-T8	1.41	T19-T9	1	T19-T10	1.41
T19-T11	8	T19-T12	7	T19-T13	6	T19-T14	5	T19-T15	4
T19-T16	3	T19-T17	2	T19-T18	1	T19-T19	0	T19-T20	1
T19-T21	8.06	T19-T22	7.07	T19-T23	6.08	T19-T24	5.1	T19-T25	4.12
T19-T26	3.16	T19-T27	2.24	T19-T28	1.41	T19-T29	1	T19-T30	1.41
T20-T1	9.06	T20-T2	8.06	T20-T3	7.07	T20-T4	6.08	T20-T5	5.1
T20-T6	4.12	T20-T7	3.16	T20-T8	2.24	T20-T9	1.41	T20-T10	1
T20-T11	9	T20-T12	8	T20-T13	7	T20-T14	6	T20-T15	5
T20-T16	4	T20-T17	3	T20-T18	2	T20-T19	1	T20-T20	0
T20-T21	9.06	T20-T22	8.06	T20-T23	7.07	T20-T24	6.08	T20-T25	5.1
T20-T26	4.12	T20-T27	3.16	T20-T28	2.24	T20-T29	1.41	T20-T30	1
T21-T1	2	T21-T2	2.24	T21-T3	2.83	T21-T4	3.61	T21-T5	4.47
T21-T6	5.39	T21-T7	6.32	T21-T8	7.28	T21-T9	8.25	T21-T10	9.22
T21-T11	1	T21-T12	1.41	T21-T13	2.24	T21-T14	3.16	T21-T15	4.12
T21-T16	5.1	T21-T17	6.08	T21-T18	7.07	T21-T19	8.06	T21-T20	9.06
T21-T21	0	T21-T22	1	T21-T23	2	T21-T24	3	T21-T25	4
T21-T26	5	T21-T27	6	T21-T28	7	T21-T29	8	T21-T30	9
T22-T1	2.24	T22-T2	2	T22-T3	2.24	T22-T4	2.83	T22-T5	3.61
T22-T6	4.47	T22-T7	5.39	T22-T8	6.32	T22-T9	7.28	T22-T10	8.25
T22-T11	1.41	T22-T12	1	T22-T13	1.41	T22-T14	2.24	T22-T15	3.16
T22-T16	4.12	T22-T17	5.1	T22-T18	6.08	T22-T19	7.07	T22-T20	8.06
T22-T21	1	T22-T22	0	T22-T23	1	T22-T24	2	T22-T25	3
T22-T26	4	T22-T27	5	T22-T28	6	T22-T29	7	T22-T30	8
T23-T1	2.83	T23-T2	2.24	T23-T3	2	T23-T4	2.24	T23-T5	2.83
T23-T6	3.61	T23-T7	4.47	T23-T8	5.39	T23-T9	6.32	T23-T10	7.28
T23-T11	2.24	T23-T12	1.41	T23-T13	1	T23-T14	1.41	T23-T15	2.24
T23-T16	3.16	T23-T17	4.12	T23-T18	5.1	T23-T19	6.08	T23-T20	7.07
T23-T21	2	T23-T22	1	T23-T23	0	T23-T24	1	T23-T25	2
T23-T26	3	T23-T27	4	T23-T28	5	T23-T29	6	T23-T30	7
T24-T1	3.61	T24-T2	2.83	T24-T3	2.24	T24-T4	2	T24-T5	2.24
T24-T6	2.83	T24-T7	3.61	T24-T8	4.47	T24-T9	5.39	T24-T10	6.32
T24-T11	3.16	T24-T12	2.24	T24-T13	1.41	T24-T14	1	T24-T15	1.41
T24-T16	2.24	T24-T17	3.16	T24-T18	4.12	T24-T19	5.1	T24-T20	6.08
T24-T21	3	T24-T22	2	T24-T23	1	T24-T24	0	T24-T25	1
T24-T26	2	T24-T27	3	T24-T28	4	T24-T29	5	T24-T30	6

A Input-Daten

T25-T1	4.47	T25-T2	3.61	T25-T3	2.83	T25-T4	2.24	T25-T5	2
T25-T6	2.24	T25-T7	2.83	T25-T8	3.61	T25-T9	4.47	T25-T10	5.39
T25-T11	4.12	T25-T12	3.16	T25-T13	2.24	T25-T14	1.41	T25-T15	1
T25-T16	1.41	T25-T17	2.24	T25-T18	3.16	T25-T19	4.12	T25-T20	5.1
T25-T21	4	T25-T22	3	T25-T23	2	T25-T24	1	T25-T25	0
T25-T26	1	T25-T27	2	T25-T28	3	T25-T29	4	T25-T30	5
T26-T1	5.39	T26-T2	4.47	T26-T3	3.61	T26-T4	2.83	T26-T5	2.24
T26-T6	2	T26-T7	2.24	T26-T8	2.83	T26-T9	3.61	T26-T10	4.47
T26-T11	5.1	T26-T12	4.12	T26-T13	3.16	T26-T14	2.24	T26-T15	1.41
T26-T16	1	T26-T17	1.41	T26-T18	2.24	T26-T19	3.16	T26-T20	4.12
T26-T21	5	T26-T22	4	T26-T23	3	T26-T24	2	T26-T25	1
T26-T26	0	T26-T27	1	T26-T28	2	T26-T29	3	T26-T30	4
T27-T1	6.32	T27-T2	5.39	T27-T3	4.47	T27-T4	3.61	T27-T5	2.83
T27-T6	2.24	T27-T7	2	T27-T8	2.24	T27-T9	2.83	T27-T10	3.61
T27-T11	6.08	T27-T12	5.1	T27-T13	4.12	T27-T14	3.16	T27-T15	2.24
T27-T16	1.41	T27-T17	1	T27-T18	1.41	T27-T19	2.24	T27-T20	3.16
T27-T21	6	T27-T22	5	T27-T23	4	T27-T24	3	T27-T25	2
T27-T26	1	T27-T27	0	T27-T28	1	T27-T29	2	T27-T30	3
T28-T1	7.28	T28-T2	6.32	T28-T3	5.39	T28-T4	4.47	T28-T5	3.61
T28-T6	2.83	T28-T7	2.24	T28-T8	2	T28-T9	2.24	T28-T10	2.83
T28-T11	7.07	T28-T12	6.08	T28-T13	5.1	T28-T14	4.12	T28-T15	3.16
T28-T16	2.24	T28-T17	1.41	T28-T18	1	T28-T19	1.41	T28-T20	2.24
T28-T21	7	T28-T22	6	T28-T23	5	T28-T24	4	T28-T25	3
T28-T26	2	T28-T27	1	T28-T28	0	T28-T29	1	T28-T30	2
T29-T1	8.25	T29-T2	7.28	T29-T3	6.32	T29-T4	5.39	T29-T5	4.47
T29-T6	3.61	T29-T7	2.83	T29-T8	2.24	T29-T9	2	T29-T10	2.24
T29-T11	8.06	T29-T12	7.07	T29-T13	6.08	T29-T14	5.1	T29-T15	4.12
T29-T16	3.16	T29-T17	2.24	T29-T18	1.41	T29-T19	1	T29-T20	1.41
T29-T21	8	T29-T22	7	T29-T23	6	T29-T24	5	T29-T25	4
T29-T26	3	T29-T27	2	T29-T28	1	T29-T29	0	T29-T30	1
T30-T1	9.22	T30-T2	8.25	T30-T3	7.28	T30-T4	6.32	T30-T5	5.39
T30-T6	4.47	T30-T7	3.61	T30-T8	2.83	T30-T9	2.24	T30-T10	2
T30-T11	9.06	T30-T12	8.06	T30-T13	7.07	T30-T14	6.08	T30-T15	5.1
T30-T16	4.12	T30-T17	3.16	T30-T18	2.24	T30-T19	1.41	T30-T20	1
T30-T21	9	T30-T22	8	T30-T23	7	T30-T24	6	T30-T25	5
T30-T26	4	T30-T27	3	T30-T28	2	T30-T29	1	T30-T30	0

A.2.4 Layout 4 - 4x8-Layout

- numberofkeys = 32, Rows = 4, Columns = 8

Tabelle 39: Tastenbelegungen für das 4x8-Layout

Belegung	Input-Daten
KALQ	(22 2 12 25 14 27 5 4 13 8 21 23 1 26 7 9 24 17 19 6 16 28 3 11 18 20)

Tabelle 40: Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 4x8-Layout

T1	3.54	T2	2.55	T3	1.58	T4	0.71	T5	0.71	T6	1.58	T7	2.55
T8	3.54	T9	3.81	T10	2.92	T11	2.12	T12	1.58	T13	1.58	T14	2.12
T15	2.92	T16	3.81	T17	4.3	T18	3.54	T19	2.92	T20	2.55	T21	2.55
T22	2.92	T23	3.54	T24	4.3	T25	4.95	T26	4.3	T27	3.81	T28	3.54
T29	3.54	T30	3.81	T31	4.3	T32	4.95						

Tabelle 41: Distanzen \hat{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 4x8-Layout

T1	2.92	T2	3.54	T3	4.3	T4	5.15	T5	5.15	T6	4.3	T7	3.54
T8	2.92	T9	2.12	T10	2.92	T11	3.81	T12	4.74	T13	4.74	T14	3.81
T15	2.92	T16	2.12	T17	1.58	T18	2.55	T19	3.54	T20	4.53	T21	4.53
T22	3.54	T23	2.55	T24	1.58	T25	1.58	T26	2.55	T27	3.54	T28	4.53
T29	4.53	T30	3.54	T31	2.55	T32	1.58						

Tabelle 42: Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 4x8-Layout

T1	1.8	T2	1.5	T3	1.8	T4	2.5	T5	2.5	T6	1.8	T7	1.5
T8	1.8	T9	1.12	T10	0.5	T11	1.12	T12	2.06	T13	2.06	T14	1.12
T15	0.5	T16	1.12	T17	1.12	T18	0.5	T19	1.12	T20	2.06	T21	2.06
T22	1.12	T23	0.5	T24	1.12	T25	1.8	T26	1.5	T27	1.8	T28	2.5
T29	2.5	T30	1.8	T31	1.5	T32	1.8						

Tabelle 43: Distanzen d_{jl} zwischen den Tasten für das vertikale 4x8-Layout

T1-T1	0	T1-T2	1	T1-T3	2	T1-T4	3	T1-T5	4
T1-T6	5	T1-T7	6	T1-T8	7	T1-T9	1	T1-T10	1.41
T1-T11	2.24	T1-T12	3.16	T1-T13	4.12	T1-T14	5.1	T1-T15	6.08

A Input-Daten

T1-T16	7.07	T1-T17	2	T1-T18	2.24	T1-T19	2.83	T1-T20	3.61
T1-T21	4.47	T1-T22	5.39	T1-T23	6.32	T1-T24	7.28	T1-T25	3
T1-T26	3.16	T1-T27	3.61	T1-T28	4.24	T1-T29	5	T1-T30	5.83
T1-T31	6.71	T1-T32	7.62	T2-T1	1	T2-T2	0	T2-T3	1
T2-T4	2	T2-T5	3	T2-T6	4	T2-T7	5	T2-T8	6
T2-T9	1.41	T2-T10	1	T2-T11	1.41	T2-T12	2.24	T2-T13	3.16
T2-T14	4.12	T2-T15	5.1	T2-T16	6.08	T2-T17	2.24	T2-T18	2
T2-T19	2.24	T2-T20	2.83	T2-T21	3.61	T2-T22	4.47	T2-T23	5.39
T2-T24	6.32	T2-T25	3.16	T2-T26	3	T2-T27	3.16	T2-T28	3.61
T2-T29	4.24	T2-T30	5	T2-T31	5.83	T2-T32	6.71	T3-T1	2
T3-T2	1	T3-T3	0	T3-T4	1	T3-T5	2	T3-T6	3
T3-T7	4	T3-T8	5	T3-T9	2.24	T3-T10	1.41	T3-T11	1
T3-T12	1.41	T3-T13	2.24	T3-T14	3.16	T3-T15	4.12	T3-T16	5.1
T3-T17	2.83	T3-T18	2.24	T3-T19	2	T3-T20	2.24	T3-T21	2.83
T3-T22	3.61	T3-T23	4.47	T3-T24	5.39	T3-T25	3.61	T3-T26	3.16
T3-T27	3	T3-T28	3.16	T3-T29	3.61	T3-T30	4.24	T3-T31	5
T3-T32	5.83	T4-T1	3	T4-T2	2	T4-T3	1	T4-T4	0
T4-T5	1	T4-T6	2	T4-T7	3	T4-T8	4	T4-T9	3.16
T4-T10	2.24	T4-T11	1.41	T4-T12	1	T4-T13	1.41	T4-T14	2.24
T4-T15	3.16	T4-T16	4.12	T4-T17	3.61	T4-T18	2.83	T4-T19	2.24
T4-T20	2	T4-T21	2.24	T4-T22	2.83	T4-T23	3.61	T4-T24	4.47
T4-T25	4.24	T4-T26	3.61	T4-T27	3.16	T4-T28	3	T4-T29	3.16
T4-T30	3.61	T4-T31	4.24	T4-T32	5	T5-T1	4	T5-T2	3
T5-T3	2	T5-T4	1	T5-T5	0	T5-T6	1	T5-T7	2
T5-T8	3	T5-T9	4.12	T5-T10	3.16	T5-T11	2.24	T5-T12	1.41
T5-T13	1	T5-T14	1.41	T5-T15	2.24	T5-T16	3.16	T5-T17	4.47
T5-T18	3.61	T5-T19	2.83	T5-T20	2.24	T5-T21	2	T5-T22	2.24
T5-T23	2.83	T5-T24	3.61	T5-T25	5	T5-T26	4.24	T5-T27	3.61
T5-T28	3.16	T5-T29	3	T5-T30	3.16	T5-T31	3.61	T5-T32	4.24
T6-T1	5	T6-T2	4	T6-T3	3	T6-T4	2	T6-T5	1
T6-T6	0	T6-T7	1	T6-T8	2	T6-T9	5.1	T6-T10	4.12
T6-T11	3.16	T6-T12	2.24	T6-T13	1.41	T6-T14	1	T6-T15	1.41
T6-T16	2.24	T6-T17	5.39	T6-T18	4.47	T6-T19	3.61	T6-T20	2.83
T6-T21	2.24	T6-T22	2	T6-T23	2.24	T6-T24	2.83	T6-T25	5.83
T6-T26	5	T6-T27	4.24	T6-T28	3.61	T6-T29	3.16	T6-T30	3
T6-T31	3.16	T6-T32	3.61	T7-T1	6	T7-T2	5	T7-T3	4
T7-T4	3	T7-T5	2	T7-T6	1	T7-T7	0	T7-T8	1
T7-T9	6.08	T7-T10	5.1	T7-T11	4.12	T7-T12	3.16	T7-T13	2.24
T7-T14	1.41	T7-T15	1	T7-T16	1.41	T7-T17	6.32	T7-T18	5.39
T7-T19	4.47	T7-T20	3.61	T7-T21	2.83	T7-T22	2.24	T7-T23	2
T7-T24	2.24	T7-T25	6.71	T7-T26	5.83	T7-T27	5	T7-T28	4.24
T7-T29	3.61	T7-T30	3.16	T7-T31	3	T7-T32	3.16	T8-T1	7
T8-T2	6	T8-T3	5	T8-T4	4	T8-T5	3	T8-T6	2
T8-T7	1	T8-T8	0	T8-T9	7.07	T8-T10	6.08	T8-T11	5.1
T8-T12	4.12	T8-T13	3.16	T8-T14	2.24	T8-T15	1.41	T8-T16	1
T8-T17	7.28	T8-T18	6.32	T8-T19	5.39	T8-T20	4.47	T8-T21	3.61

T8-T22	2.83	T8-T23	2.24	T8-T24	2	T8-T25	7.62	T8-T26	6.71
T8-T27	5.83	T8-T28	5	T8-T29	4.24	T8-T30	3.61	T8-T31	3.16
T8-T32	3	T9-T1	1	T9-T2	1.41	T9-T3	2.24	T9-T4	3.16
T9-T5	4.12	T9-T6	5.1	T9-T7	6.08	T9-T8	7.07	T9-T9	0
T9-T10	1	T9-T11	2	T9-T12	3	T9-T13	4	T9-T14	5
T9-T15	6	T9-T16	7	T9-T17	1	T9-T18	1.41	T9-T19	2.24
T9-T20	3.16	T9-T21	4.12	T9-T22	5.1	T9-T23	6.08	T9-T24	7.07
T9-T25	2	T9-T26	2.24	T9-T27	2.83	T9-T28	3.61	T9-T29	4.47
T9-T30	5.39	T9-T31	6.32	T9-T32	7.28	T10-T1	1.41	T10-T2	1
T10-T3	1.41	T10-T4	2.24	T10-T5	3.16	T10-T6	4.12	T10-T7	5.1
T10-T8	6.08	T10-T9	1	T10-T10	0	T10-T11	1	T10-T12	2
T10-T13	3	T10-T14	4	T10-T15	5	T10-T16	6	T10-T17	1.41
T10-T18	1	T10-T19	1.41	T10-T20	2.24	T10-T21	3.16	T10-T22	4.12
T10-T23	5.1	T10-T24	6.08	T10-T25	2.24	T10-T26	2	T10-T27	2.24
T10-T28	2.83	T10-T29	3.61	T10-T30	4.47	T10-T31	5.39	T10-T32	6.32
T11-T1	2.24	T11-T2	1.41	T11-T3	1	T11-T4	1.41	T11-T5	2.24
T11-T6	3.16	T11-T7	4.12	T11-T8	5.1	T11-T9	2	T11-T10	1
T11-T11	0	T11-T12	1	T11-T13	2	T11-T14	3	T11-T15	4
T11-T16	5	T11-T17	2.24	T11-T18	1.41	T11-T19	1	T11-T20	1.41
T11-T21	2.24	T11-T22	3.16	T11-T23	4.12	T11-T24	5.1	T11-T25	2.83
T11-T26	2.24	T11-T27	2	T11-T28	2.24	T11-T29	2.83	T11-T30	3.61
T11-T31	4.47	T11-T32	5.39	T12-T1	3.16	T12-T2	2.24	T12-T3	1.41
T12-T4	1	T12-T5	1.41	T12-T6	2.24	T12-T7	3.16	T12-T8	4.12
T12-T9	3	T12-T10	2	T12-T11	1	T12-T12	0	T12-T13	1
T12-T14	2	T12-T15	3	T12-T16	4	T12-T17	3.16	T12-T18	2.24
T12-T19	1.41	T12-T20	1	T12-T21	1.41	T12-T22	2.24	T12-T23	3.16
T12-T24	4.12	T12-T25	3.61	T12-T26	2.83	T12-T27	2.24	T12-T28	2
T12-T29	2.24	T12-T30	2.83	T12-T31	3.61	T12-T32	4.47	T13-T1	4.12
T13-T2	3.16	T13-T3	2.24	T13-T4	1.41	T13-T5	1	T13-T6	1.41
T13-T7	2.24	T13-T8	3.16	T13-T9	4	T13-T10	3	T13-T11	2
T13-T12	1	T13-T13	0	T13-T14	1	T13-T15	2	T13-T16	3
T13-T17	4.12	T13-T18	3.16	T13-T19	2.24	T13-T20	1.41	T13-T21	1
T13-T22	1.41	T13-T23	2.24	T13-T24	3.16	T13-T25	4.47	T13-T26	3.61
T13-T27	2.83	T13-T28	2.24	T13-T29	2	T13-T30	2.24	T13-T31	2.83
T13-T32	3.61	T14-T1	5.1	T14-T2	4.12	T14-T3	3.16	T14-T4	2.24
T14-T5	1.41	T14-T6	1	T14-T7	1.41	T14-T8	2.24	T14-T9	5
T14-T10	4	T14-T11	3	T14-T12	2	T14-T13	1	T14-T14	0
T14-T15	1	T14-T16	2	T14-T17	5.1	T14-T18	4.12	T14-T19	3.16
T14-T20	2.24	T14-T21	1.41	T14-T22	1	T14-T23	1.41	T14-T24	2.24
T14-T25	5.39	T14-T26	4.47	T14-T27	3.61	T14-T28	2.83	T14-T29	2.24
T14-T30	2	T14-T31	2.24	T14-T32	2.83	T15-T1	6.08	T15-T2	5.1
T15-T3	4.12	T15-T4	3.16	T15-T5	2.24	T15-T6	1.41	T15-T7	1
T15-T8	1.41	T15-T9	6	T15-T10	5	T15-T11	4	T15-T12	3
T15-T13	2	T15-T14	1	T15-T15	0	T15-T16	1	T15-T17	6.08
T15-T18	5.1	T15-T19	4.12	T15-T20	3.16	T15-T21	2.24	T15-T22	1.41
T15-T23	1	T15-T24	1.41	T15-T25	6.32	T15-T26	5.39	T15-T27	4.47

A Input-Daten

T15-T28	3.61	T15-T29	2.83	T15-T30	2.24	T15-T31	2	T15-T32	2.24
T16-T1	7.07	T16-T2	6.08	T16-T3	5.1	T16-T4	4.12	T16-T5	3.16
T16-T6	2.24	T16-T7	1.41	T16-T8	1	T16-T9	7	T16-T10	6
T16-T11	5	T16-T12	4	T16-T13	3	T16-T14	2	T16-T15	1
T16-T16	0	T16-T17	7.07	T16-T18	6.08	T16-T19	5.1	T16-T20	4.12
T16-T21	3.16	T16-T22	2.24	T16-T23	1.41	T16-T24	1	T16-T25	7.28
T16-T26	6.32	T16-T27	5.39	T16-T28	4.47	T16-T29	3.61	T16-T30	2.83
T16-T31	2.24	T16-T32	2	T17-T1	2	T17-T2	2.24	T17-T3	2.83
T17-T4	3.61	T17-T5	4.47	T17-T6	5.39	T17-T7	6.32	T17-T8	7.28
T17-T9	1	T17-T10	1.41	T17-T11	2.24	T17-T12	3.16	T17-T13	4.12
T17-T14	5.1	T17-T15	6.08	T17-T16	7.07	T17-T17	0	T17-T18	1
T17-T19	2	T17-T20	3	T17-T21	4	T17-T22	5	T17-T23	6
T17-T24	7	T17-T25	1	T17-T26	1.41	T17-T27	2.24	T17-T28	3.16
T17-T29	4.12	T17-T30	5.1	T17-T31	6.08	T17-T32	7.07	T18-T1	2.24
T18-T2	2	T18-T3	2.24	T18-T4	2.83	T18-T5	3.61	T18-T6	4.47
T18-T7	5.39	T18-T8	6.32	T18-T9	1.41	T18-T10	1	T18-T11	1.41
T18-T12	2.24	T18-T13	3.16	T18-T14	4.12	T18-T15	5.1	T18-T16	6.08
T18-T17	1	T18-T18	0	T18-T19	1	T18-T20	2	T18-T21	3
T18-T22	4	T18-T23	5	T18-T24	6	T18-T25	1.41	T18-T26	1
T18-T27	1.41	T18-T28	2.24	T18-T29	3.16	T18-T30	4.12	T18-T31	5.1
T18-T32	6.08	T19-T1	2.83	T19-T2	2.24	T19-T3	2	T19-T4	2.24
T19-T5	2.83	T19-T6	3.61	T19-T7	4.47	T19-T8	5.39	T19-T9	2.24
T19-T10	1.41	T19-T11	1	T19-T12	1.41	T19-T13	2.24	T19-T14	3.16
T19-T15	4.12	T19-T16	5.1	T19-T17	2	T19-T18	1	T19-T19	0
T19-T20	1	T19-T21	2	T19-T22	3	T19-T23	4	T19-T24	5
T19-T25	2.24	T19-T26	1.41	T19-T27	1	T19-T28	1.41	T19-T29	2.24
T19-T30	3.16	T19-T31	4.12	T19-T32	5.1	T20-T1	3.61	T20-T2	2.83
T20-T3	2.24	T20-T4	2	T20-T5	2.24	T20-T6	2.83	T20-T7	3.61
T20-T8	4.47	T20-T9	3.16	T20-T10	2.24	T20-T11	1.41	T20-T12	1
T20-T13	1.41	T20-T14	2.24	T20-T15	3.16	T20-T16	4.12	T20-T17	3
T20-T18	2	T20-T19	1	T20-T20	0	T20-T21	1	T20-T22	2
T20-T23	3	T20-T24	4	T20-T25	3.16	T20-T26	2.24	T20-T27	1.41
T20-T28	1	T20-T29	1.41	T20-T30	2.24	T20-T31	3.16	T20-T32	4.12
T21-T1	4.47	T21-T2	3.61	T21-T3	2.83	T21-T4	2.24	T21-T5	2
T21-T6	2.24	T21-T7	2.83	T21-T8	3.61	T21-T9	4.12	T21-T10	3.16
T21-T11	2.24	T21-T12	1.41	T21-T13	1	T21-T14	1.41	T21-T15	2.24
T21-T16	3.16	T21-T17	4	T21-T18	3	T21-T19	2	T21-T20	1
T21-T21	0	T21-T22	1	T21-T23	2	T21-T24	3	T21-T25	4.12
T21-T26	3.16	T21-T27	2.24	T21-T28	1.41	T21-T29	1	T21-T30	1.41
T21-T31	2.24	T21-T32	3.16	T22-T1	5.39	T22-T2	4.47	T22-T3	3.61
T22-T4	2.83	T22-T5	2.24	T22-T6	2	T22-T7	2.24	T22-T8	2.83
T22-T9	5.1	T22-T10	4.12	T22-T11	3.16	T22-T12	2.24	T22-T13	1.41
T22-T14	1	T22-T15	1.41	T22-T16	2.24	T22-T17	5	T22-T18	4
T22-T19	3	T22-T20	2	T22-T21	1	T22-T22	0	T22-T23	1
T22-T24	2	T22-T25	5.1	T22-T26	4.12	T22-T27	3.16	T22-T28	2.24

T22-T29	1.41	T22-T30	1	T22-T31	1.41	T22-T32	2.24	T23-T1	6.32
T23-T2	5.39	T23-T3	4.47	T23-T4	3.61	T23-T5	2.83	T23-T6	2.24
T23-T7	2	T23-T8	2.24	T23-T9	6.08	T23-T10	5.1	T23-T11	4.12
T23-T12	3.16	T23-T13	2.24	T23-T14	1.41	T23-T15	1	T23-T16	1.41
T23-T17	6	T23-T18	5	T23-T19	4	T23-T20	3	T23-T21	2
T23-T22	1	T23-T23	0	T23-T24	1	T23-T25	6.08	T23-T26	5.1
T23-T27	4.12	T23-T28	3.16	T23-T29	2.24	T23-T30	1.41	T23-T31	1
T23-T32	1.41	T24-T1	7.28	T24-T2	6.32	T24-T3	5.39	T24-T4	4.47
T24-T5	3.61	T24-T6	2.83	T24-T7	2.24	T24-T8	2	T24-T9	7.07
T24-T10	6.08	T24-T11	5.1	T24-T12	4.12	T24-T13	3.16	T24-T14	2.24
T24-T15	1.41	T24-T16	1	T24-T17	7	T24-T18	6	T24-T19	5
T24-T20	4	T24-T21	3	T24-T22	2	T24-T23	1	T24-T24	0
T24-T25	7.07	T24-T26	6.08	T24-T27	5.1	T24-T28	4.12	T24-T29	3.16
T24-T30	2.24	T24-T31	1.41	T24-T32	1	T25-T1	3	T25-T2	3.16
T25-T3	3.61	T25-T4	4.24	T25-T5	5	T25-T6	5.83	T25-T7	6.71
T25-T8	7.62	T25-T9	2	T25-T10	2.24	T25-T11	2.83	T25-T12	3.61
T25-T13	4.47	T25-T14	5.39	T25-T15	6.32	T25-T16	7.28	T25-T17	1
T25-T18	1.41	T25-T19	2.24	T25-T20	3.16	T25-T21	4.12	T25-T22	5.1
T25-T23	6.08	T25-T24	7.07	T25-T25	0	T25-T26	1	T25-T27	2
T25-T28	3	T25-T29	4	T25-T30	5	T25-T31	6	T25-T32	7
T26-T1	3.16	T26-T2	3	T26-T3	3.16	T26-T4	3.61	T26-T5	4.24
T26-T6	5	T26-T7	5.83	T26-T8	6.71	T26-T9	2.24	T26-T10	2
T26-T11	2.24	T26-T12	2.83	T26-T13	3.61	T26-T14	4.47	T26-T15	5.39
T26-T16	6.32	T26-T17	1.41	T26-T18	1	T26-T19	1.41	T26-T20	2.24
T26-T21	3.16	T26-T22	4.12	T26-T23	5.1	T26-T24	6.08	T26-T25	1
T26-T26	0	T26-T27	1	T26-T28	2	T26-T29	3	T26-T30	4
T26-T31	5	T26-T32	6	T27-T1	3.61	T27-T2	3.16	T27-T3	3
T27-T4	3.16	T27-T5	3.61	T27-T6	4.24	T27-T7	5	T27-T8	5.83
T27-T9	2.83	T27-T10	2.24	T27-T11	2	T27-T12	2.24	T27-T13	2.83
T27-T14	3.61	T27-T15	4.47	T27-T16	5.39	T27-T17	2.24	T27-T18	1.41
T27-T19	1	T27-T20	1.41	T27-T21	2.24	T27-T22	3.16	T27-T23	4.12
T27-T24	5.1	T27-T25	2	T27-T26	1	T27-T27	0	T27-T28	1
T27-T29	2	T27-T30	3	T27-T31	4	T27-T32	5	T28-T1	4.24
T28-T2	3.61	T28-T3	3.16	T28-T4	3	T28-T5	3.16	T28-T6	3.61
T28-T7	4.24	T28-T8	5	T28-T9	3.61	T28-T10	2.83	T28-T11	2.24
T28-T12	2	T28-T13	2.24	T28-T14	2.83	T28-T15	3.61	T28-T16	4.47
T28-T17	3.16	T28-T18	2.24	T28-T19	1.41	T28-T20	1	T28-T21	1.41
T28-T22	2.24	T28-T23	3.16	T28-T24	4.12	T28-T25	3	T28-T26	2
T28-T27	1	T28-T28	0	T28-T29	1	T28-T30	2	T28-T31	3
T28-T32	4	T29-T1	5	T29-T2	4.24	T29-T3	3.61	T29-T4	3.16
T29-T5	3	T29-T6	3.16	T29-T7	3.61	T29-T8	4.24	T29-T9	4.47
T29-T10	3.61	T29-T11	2.83	T29-T12	2.24	T29-T13	2	T29-T14	2.24
T29-T15	2.83	T29-T16	3.61	T29-T17	4.12	T29-T18	3.16	T29-T19	2.24
T29-T20	1.41	T29-T21	1	T29-T22	1.41	T29-T23	2.24	T29-T24	3.16
T29-T25	4	T29-T26	3	T29-T27	2	T29-T28	1	T29-T29	0
T29-T30	1	T29-T31	2	T29-T32	3	T30-T1	5.83	T30-T2	5

T30-T3	4.24	T30-T4	3.61	T30-T5	3.16	T30-T6	3	T30-T7	3.16
T30-T8	3.61	T30-T9	5.39	T30-T10	4.47	T30-T11	3.61	T30-T12	2.83
T30-T13	2.24	T30-T14	2	T30-T15	2.24	T30-T16	2.83	T30-T17	5.1
T30-T18	4.12	T30-T19	3.16	T30-T20	2.24	T30-T21	1.41	T30-T22	1
T30-T23	1.41	T30-T24	2.24	T30-T25	5	T30-T26	4	T30-T27	3
T30-T28	2	T30-T29	1	T30-T30	0	T30-T31	1	T30-T32	2
T31-T1	6.71	T31-T2	5.83	T31-T3	5	T31-T4	4.24	T31-T5	3.61
T31-T6	3.16	T31-T7	3	T31-T8	3.16	T31-T9	6.32	T31-T10	5.39
T31-T11	4.47	T31-T12	3.61	T31-T13	2.83	T31-T14	2.24	T31-T15	2
T31-T16	2.24	T31-T17	6.08	T31-T18	5.1	T31-T19	4.12	T31-T20	3.16
T31-T21	2.24	T31-T22	1.41	T31-T23	1	T31-T24	1.41	T31-T25	6
T31-T26	5	T31-T27	4	T31-T28	3	T31-T29	2	T31-T30	1
T31-T31	0	T31-T32	1	T32-T1	7.62	T32-T2	6.71	T32-T3	5.83
T32-T4	5	T32-T5	4.24	T32-T6	3.61	T32-T7	3.16	T32-T8	3
T32-T9	7.28	T32-T10	6.32	T32-T11	5.39	T32-T12	4.47	T32-T13	3.61
T32-T14	2.83	T32-T15	2.24	T32-T16	2	T32-T17	7.07	T32-T18	6.08
T32-T19	5.1	T32-T20	4.12	T32-T21	3.16	T32-T22	2.24	T32-T23	1.41
T32-T24	1	T32-T25	7	T32-T26	6	T32-T27	5	T32-T28	4
T32-T29	3	T32-T30	2	T32-T31	1	T32-T32	0		

A.2.5 Layouts der Testinstanzen 1 und 2 - 3x3- und 2x4-Layout

3x3-Layout für Testinstanz 1

Tabelle 44: Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 3x3-Layout

T1	1.12	T2	0.5	T3	1.12	T4	1.8	T5	1.5	T6	1.8	T7	2.69
T8	2.5	T9	2.69										

Tabelle 45: Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 3x3-Layout

T1	4.72	T2	5.59	T3	4.72	T4	4.27	T5	5.22	T6	4.27	T7	4.03
T8	5.02	T9	4.03										

Tabelle 46: Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 3x3-Layout

T1	2.12	T2	2.92	T3	2.12	T4	1.58	T5	2.55	T6	1.58	T7	1.58
T8	2.55	T9	1.58										

Tabelle 47: Distanzen d_{jl} zwischen den Tasten für das vertikale 3x3-Layout

T1-T1	0	T1-T2	1	T1-T3	2	T1-T4	1	T1-T5	1.41
T1-T6	2.24	T1-T7	2	T1-T8	2.24	T1-T9	2.83	T2-T1	1
T2-T2	0	T2-T3	1	T2-T4	1.41	T2-T5	1	T2-T6	1.41
T2-T7	2.24	T2-T8	2	T2-T9	2.24	T3-T1	2	T3-T2	1
T3-T3	0	T3-T4	2.24	T3-T5	1.41	T3-T6	1	T3-T7	2.83
T3-T8	2.24	T3-T9	2	T4-T1	1	T4-T2	1.41	T4-T3	2.24
T4-T4	0	T4-T5	1	T4-T6	2	T4-T7	1	T4-T8	1.41
T4-T9	2.24	T5-T1	1.41	T5-T2	1	T5-T3	1.41	T5-T4	1
T5-T5	0	T5-T6	1	T5-T7	1.41	T5-T8	1	T5-T9	1.41
T6-T1	2.24	T6-T2	1.41	T6-T3	1	T6-T4	2	T6-T5	1
T6-T6	0	T6-T7	2.24	T6-T8	1.41	T6-T9	1	T7-T1	2
T7-T2	2.24	T7-T3	2.83	T7-T4	1	T7-T5	1.41	T7-T6	2.24
T7-T7	0	T7-T8	1	T7-T9	2	T8-T1	2.24	T8-T2	2
T8-T3	2.24	T8-T4	1.41	T8-T5	1	T8-T6	1.41	T8-T7	1
T8-T8	0	T8-T9	1	T9-T1	2.83	T9-T2	2.24	T9-T3	2
T9-T4	2.24	T9-T5	1.41	T9-T6	1	T9-T7	2	T9-T8	1
T9-T9	0								

2x4-Layout für Testinstanz 2

Tabelle 48: Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 2x4-Layout

T1	1.58	T2	0.71	T3	0.71	T4	1.58	T5	2.12	T6	1.58	T7	1.58
T8	2.12												

Tabelle 49: Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 2x4-Layout

T1	4.3	T2	5.15	T3	5.15	T4	4.3	T5	3.81	T6	4.74	T7	4.74
T8	3.81												

Tabelle 50: Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 2x4-Layout

T1	1.8	T2	2.5	T3	2.5	T4	1.8	T5	1.12	T6	2.06	T7	2.06
T8	1.12												

Tabelle 51: Distanzen d_{jl} zwischen den Tasten für das vertikale 2x4-Layout

T1-T1	0	T1-T2	1	T1-T3	2	T1-T4	3	T1-T5	1
T1-T6	1.41	T1-T7	2.24	T1-T8	3.16	T2-T1	1	T2-T2	0
T2-T3	1	T2-T4	2	T2-T5	1.41	T2-T6	1	T2-T7	1.41
T2-T8	2.24	T3-T1	2	T3-T2	1	T3-T3	0	T3-T4	1
T3-T5	2.24	T3-T6	1.41	T3-T7	1	T3-T8	1.41	T4-T1	3
T4-T2	2	T4-T3	1	T4-T4	0	T4-T5	3.16	T4-T6	2.24
T4-T7	1.41	T4-T8	1	T5-T1	1	T5-T2	1.41	T5-T3	2.24
T5-T4	3.16	T5-T5	0	T5-T6	1	T5-T7	2	T5-T8	3
T6-T1	1.41	T6-T2	1	T6-T3	1.41	T6-T4	2.24	T6-T5	1
T6-T6	0	T6-T7	1	T6-T8	2	T7-T1	2.24	T7-T2	1.41
T7-T3	1	T7-T4	1.41	T7-T5	2	T7-T6	1	T7-T7	0
T7-T8	1	T8-T1	3.16	T8-T2	2.24	T8-T3	1.41	T8-T4	1
T8-T5	3	T8-T6	2	T8-T7	1	T8-T8	0		

A.2.6 Zusätzliche Layouts - 5x6- und 5x7-Layout

5x6-Layout

- `numberofkeys = 30`, `Rows = 5`, `Columns = 6`

Tabelle 52: Tastenbelegungen für das 5x6-Layout

Belegung	Input-Daten
<i>FITALY</i>	(10 24 3 20 16 7 19 4 8 26 6 11 28 15 21 29 25 22 23 9 27 2 5 30 12 1)
<i>OPTI II</i>	(17 26 3 11 16 27 4 15 9 6 2 23 18 10 21 29 1 22 8 14 20 5 13 30 28 25)

Tabelle 53: Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 5x6-Layout

T1	2.55	T2	1.58	T3	0.71	T4	0.71	T5	1.58	T6	2.55	T7	2.92
T8	2.12	T9	1.58	T10	1.58	T11	2.12	T12	2.92	T13	3.54	T14	2.92
T15	2.55	T16	2.55	T17	2.92	T18	3.54	T19	4.3	T20	3.81	T21	3.54
T22	3.54	T23	3.81	T24	4.3	T25	5.15	T26	4.74	T27	4.53	T28	4.53
T29	4.74	T30	5.15										

Tabelle 54: Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 5x6-Layout

T1	4.3	T2	4.95	T3	5.7	T4	5.7	T5	4.95	T6	4.3	T7	3.54
T8	4.3	T9	5.15	T10	5.15	T11	4.3	T12	3.54	T13	2.92	T14	3.81
T15	4.74	T16	4.74	T17	3.81	T18	2.92	T19	2.55	T20	3.54	T21	4.53
T22	4.53	T23	3.54	T24	2.55	T25	2.55	T26	3.54	T27	4.53	T28	4.53
T29	3.54	T30	2.55										

Tabelle 55: Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 5x6-Layout

T1	2	T2	2.24	T3	2.83	T4	2.83	T5	2.24	T6	2	T7	1
T8	1.41	T9	2.24	T10	2.24	T11	1.41	T12	1	T13	0	T14	1
T15	2	T16	2	T17	1	T18	0	T19	1	T20	1.41	T21	2.24
T22	2.24	T23	1.41	T24	1	T25	2	T26	2.24	T27	2.83	T28	2.83
T29	2.24	T30	2										

Tabelle 56: Distanzen d_{jl} zwischen den Tasten für das vertikale 5x6-Layout

T1-T1	0	T1-T2	1	T1-T3	2	T1-T4	3	T1-T5	4
T1-T6	5	T1-T7	1	T1-T8	1.41	T1-T9	2.24	T1-T10	3.16
T1-T11	4.12	T1-T12	5.1	T1-T13	2	T1-T14	2.24	T1-T15	2.83
T1-T16	3.61	T1-T17	4.47	T1-T18	5.39	T1-T19	3	T1-T20	3.16
T1-T21	3.61	T1-T22	4.24	T1-T23	5	T1-T24	5.83	T1-T25	4
T1-T26	4.12	T1-T27	4.47	T1-T28	5	T1-T29	5.66	T1-T30	6.4
T2-T1	1	T2-T2	0	T2-T3	1	T2-T4	2	T2-T5	3
T2-T6	4	T2-T7	1.41	T2-T8	1	T2-T9	1.41	T2-T10	2.24
T2-T11	3.16	T2-T12	4.12	T2-T13	2.24	T2-T14	2	T2-T15	2.24
T2-T16	2.83	T2-T17	3.61	T2-T18	4.47	T2-T19	3.16	T2-T20	3
T2-T21	3.16	T2-T22	3.61	T2-T23	4.24	T2-T24	5	T2-T25	4.12
T2-T26	4	T2-T27	4.12	T2-T28	4.47	T2-T29	5	T2-T30	5.66
T3-T1	2	T3-T2	1	T3-T3	0	T3-T4	1	T3-T5	2
T3-T6	3	T3-T7	2.24	T3-T8	1.41	T3-T9	1	T3-T10	1.41
T3-T11	2.24	T3-T12	3.16	T3-T13	2.83	T3-T14	2.24	T3-T15	2
T3-T16	2.24	T3-T17	2.83	T3-T18	3.61	T3-T19	3.61	T3-T20	3.16
T3-T21	3	T3-T22	3.16	T3-T23	3.61	T3-T24	4.24	T3-T25	4.47
T3-T26	4.12	T3-T27	4	T3-T28	4.12	T3-T29	4.47	T3-T30	5
T4-T1	3	T4-T2	2	T4-T3	1	T4-T4	0	T4-T5	1
T4-T6	2	T4-T7	3.16	T4-T8	2.24	T4-T9	1.41	T4-T10	1
T4-T11	1.41	T4-T12	2.24	T4-T13	3.61	T4-T14	2.83	T4-T15	2.24
T4-T16	2	T4-T17	2.24	T4-T18	2.83	T4-T19	4.24	T4-T20	3.61
T4-T21	3.16	T4-T22	3	T4-T23	3.16	T4-T24	3.61	T4-T25	5
T4-T26	4.47	T4-T27	4.12	T4-T28	4	T4-T29	4.12	T4-T30	4.47
T5-T1	4	T5-T2	3	T5-T3	2	T5-T4	1	T5-T5	0
T5-T6	1	T5-T7	4.12	T5-T8	3.16	T5-T9	2.24	T5-T10	1.41
T5-T11	1	T5-T12	1.41	T5-T13	4.47	T5-T14	3.61	T5-T15	2.83
T5-T16	2.24	T5-T17	2	T5-T18	2.24	T5-T19	5	T5-T20	4.24
T5-T21	3.61	T5-T22	3.16	T5-T23	3	T5-T24	3.16	T5-T25	5.66
T5-T26	5	T5-T27	4.47	T5-T28	4.12	T5-T29	4	T5-T30	4.12
T6-T1	5	T6-T2	4	T6-T3	3	T6-T4	2	T6-T5	1
T6-T6	0	T6-T7	5.1	T6-T8	4.12	T6-T9	3.16	T6-T10	2.24
T6-T11	1.41	T6-T12	1	T6-T13	5.39	T6-T14	4.47	T6-T15	3.61
T6-T16	2.83	T6-T17	2.24	T6-T18	2	T6-T19	5.83	T6-T20	5
T6-T21	4.24	T6-T22	3.61	T6-T23	3.16	T6-T24	3	T6-T25	6.4
T6-T26	5.66	T6-T27	5	T6-T28	4.47	T6-T29	4.12	T6-T30	4
T7-T1	1	T7-T2	1.41	T7-T3	2.24	T7-T4	3.16	T7-T5	4.12
T7-T6	5.1	T7-T7	0	T7-T8	1	T7-T9	2	T7-T10	3
T7-T11	4	T7-T12	5	T7-T13	1	T7-T14	1.41	T7-T15	2.24
T7-T16	3.16	T7-T17	4.12	T7-T18	5.1	T7-T19	2	T7-T20	2.24
T7-T21	2.83	T7-T22	3.61	T7-T23	4.47	T7-T24	5.39	T7-T25	3
T7-T26	3.16	T7-T27	3.61	T7-T28	4.24	T7-T29	5	T7-T30	5.83

T8-T1	1.41	T8-T2	1	T8-T3	1.41	T8-T4	2.24	T8-T5	3.16
T8-T6	4.12	T8-T7	1	T8-T8	0	T8-T9	1	T8-T10	2
T8-T11	3	T8-T12	4	T8-T13	1.41	T8-T14	1	T8-T15	1.41
T8-T16	2.24	T8-T17	3.16	T8-T18	4.12	T8-T19	2.24	T8-T20	2
T8-T21	2.24	T8-T22	2.83	T8-T23	3.61	T8-T24	4.47	T8-T25	3.16
T8-T26	3	T8-T27	3.16	T8-T28	3.61	T8-T29	4.24	T8-T30	5
T9-T1	2.24	T9-T2	1.41	T9-T3	1	T9-T4	1.41	T9-T5	2.24
T9-T6	3.16	T9-T7	2	T9-T8	1	T9-T9	0	T9-T10	1
T9-T11	2	T9-T12	3	T9-T13	2.24	T9-T14	1.41	T9-T15	1
T9-T16	1.41	T9-T17	2.24	T9-T18	3.16	T9-T19	2.83	T9-T20	2.24
T9-T21	2	T9-T22	2.24	T9-T23	2.83	T9-T24	3.61	T9-T25	3.61
T9-T26	3.16	T9-T27	3	T9-T28	3.16	T9-T29	3.61	T9-T30	4.24
T10-T1	3.16	T10-T2	2.24	T10-T3	1.41	T10-T4	1	T10-T5	1.41
T10-T6	2.24	T10-T7	3	T10-T8	2	T10-T9	1	T10-T10	0
T10-T11	1	T10-T12	2	T10-T13	3.16	T10-T14	2.24	T10-T15	1.41
T10-T16	1	T10-T17	1.41	T10-T18	2.24	T10-T19	3.61	T10-T20	2.83
T10-T21	2.24	T10-T22	2	T10-T23	2.24	T10-T24	2.83	T10-T25	4.24
T10-T26	3.61	T10-T27	3.16	T10-T28	3	T10-T29	3.16	T10-T30	3.61
T11-T1	4.12	T11-T2	3.16	T11-T3	2.24	T11-T4	1.41	T11-T5	1
T11-T6	1.41	T11-T7	4	T11-T8	3	T11-T9	2	T11-T10	1
T11-T11	0	T11-T12	1	T11-T13	4.12	T11-T14	3.16	T11-T15	2.24
T11-T16	1.41	T11-T17	1	T11-T18	1.41	T11-T19	4.47	T11-T20	3.61
T11-T21	2.83	T11-T22	2.24	T11-T23	2	T11-T24	2.24	T11-T25	5
T11-T26	4.24	T11-T27	3.61	T11-T28	3.16	T11-T29	3	T11-T30	3.16
T12-T1	5.1	T12-T2	4.12	T12-T3	3.16	T12-T4	2.24	T12-T5	1.41
T12-T6	1	T12-T7	5	T12-T8	4	T12-T9	3	T12-T10	2
T12-T11	1	T12-T12	0	T12-T13	5.1	T12-T14	4.12	T12-T15	3.16
T12-T16	2.24	T12-T17	1.41	T12-T18	1	T12-T19	5.39	T12-T20	4.47
T12-T21	3.61	T12-T22	2.83	T12-T23	2.24	T12-T24	2	T12-T25	5.83
T12-T26	5	T12-T27	4.24	T12-T28	3.61	T12-T29	3.16	T12-T30	3
T13-T1	2	T13-T2	2.24	T13-T3	2.83	T13-T4	3.61	T13-T5	4.47
T13-T6	5.39	T13-T7	1	T13-T8	1.41	T13-T9	2.24	T13-T10	3.16
T13-T11	4.12	T13-T12	5.1	T13-T13	0	T13-T14	1	T13-T15	2
T13-T16	3	T13-T17	4	T13-T18	5	T13-T19	1	T13-T20	1.41
T13-T21	2.24	T13-T22	3.16	T13-T23	4.12	T13-T24	5.1	T13-T25	2
T13-T26	2.24	T13-T27	2.83	T13-T28	3.61	T13-T29	4.47	T13-T30	5.39
T14-T1	2.24	T14-T2	2	T14-T3	2.24	T14-T4	2.83	T14-T5	3.61
T14-T6	4.47	T14-T7	1.41	T14-T8	1	T14-T9	1.41	T14-T10	2.24
T14-T11	3.16	T14-T12	4.12	T14-T13	1	T14-T14	0	T14-T15	1
T14-T16	2	T14-T17	3	T14-T18	4	T14-T19	1.41	T14-T20	1
T14-T21	1.41	T14-T22	2.24	T14-T23	3.16	T14-T24	4.12	T14-T25	2.24
T14-T26	2	T14-T27	2.24	T14-T28	2.83	T14-T29	3.61	T14-T30	4.47
T15-T1	2.83	T15-T2	2.24	T15-T3	2	T15-T4	2.24	T15-T5	2.83
T15-T6	3.61	T15-T7	2.24	T15-T8	1.41	T15-T9	1	T15-T10	1.41
T15-T11	2.24	T15-T12	3.16	T15-T13	2	T15-T14	1	T15-T15	0

A Input-Daten

T15-T16	1	T15-T17	2	T15-T18	3	T15-T19	2.24	T15-T20	1.41
T15-T21	1	T15-T22	1.41	T15-T23	2.24	T15-T24	3.16	T15-T25	2.83
T15-T26	2.24	T15-T27	2	T15-T28	2.24	T15-T29	2.83	T15-T30	3.61
T16-T1	3.61	T16-T2	2.83	T16-T3	2.24	T16-T4	2	T16-T5	2.24
T16-T6	2.83	T16-T7	3.16	T16-T8	2.24	T16-T9	1.41	T16-T10	1
T16-T11	1.41	T16-T12	2.24	T16-T13	3	T16-T14	2	T16-T15	1
T16-T16	0	T16-T17	1	T16-T18	2	T16-T19	3.16	T16-T20	2.24
T16-T21	1.41	T16-T22	1	T16-T23	1.41	T16-T24	2.24	T16-T25	3.61
T16-T26	2.83	T16-T27	2.24	T16-T28	2	T16-T29	2.24	T16-T30	2.83
T17-T1	4.47	T17-T2	3.61	T17-T3	2.83	T17-T4	2.24	T17-T5	2
T17-T6	2.24	T17-T7	4.12	T17-T8	3.16	T17-T9	2.24	T17-T10	1.41
T17-T11	1	T17-T12	1.41	T17-T13	4	T17-T14	3	T17-T15	2
T17-T16	1	T17-T17	0	T17-T18	1	T17-T19	4.12	T17-T20	3.16
T17-T21	2.24	T17-T22	1.41	T17-T23	1	T17-T24	1.41	T17-T25	4.47
T17-T26	3.61	T17-T27	2.83	T17-T28	2.24	T17-T29	2	T17-T30	2.24
T18-T1	5.39	T18-T2	4.47	T18-T3	3.61	T18-T4	2.83	T18-T5	2.24
T18-T6	2	T18-T7	5.1	T18-T8	4.12	T18-T9	3.16	T18-T10	2.24
T18-T11	1.41	T18-T12	1	T18-T13	5	T18-T14	4	T18-T15	3
T18-T16	2	T18-T17	1	T18-T18	0	T18-T19	5.1	T18-T20	4.12
T18-T21	3.16	T18-T22	2.24	T18-T23	1.41	T18-T24	1	T18-T25	5.39
T18-T26	4.47	T18-T27	3.61	T18-T28	2.83	T18-T29	2.24	T18-T30	2
T19-T1	3	T19-T2	3.16	T19-T3	3.61	T19-T4	4.24	T19-T5	5
T19-T6	5.83	T19-T7	2	T19-T8	2.24	T19-T9	2.83	T19-T10	3.61
T19-T11	4.47	T19-T12	5.39	T19-T13	1	T19-T14	1.41	T19-T15	2.24
T19-T16	3.16	T19-T17	4.12	T19-T18	5.1	T19-T19	0	T19-T20	1
T19-T21	2	T19-T22	3	T19-T23	4	T19-T24	5	T19-T25	1
T19-T26	1.41	T19-T27	2.24	T19-T28	3.16	T19-T29	4.12	T19-T30	5.1
T20-T1	3.16	T20-T2	3	T20-T3	3.16	T20-T4	3.61	T20-T5	4.24
T20-T6	5	T20-T7	2.24	T20-T8	2	T20-T9	2.24	T20-T10	2.83
T20-T11	3.61	T20-T12	4.47	T20-T13	1.41	T20-T14	1	T20-T15	1.41
T20-T16	2.24	T20-T17	3.16	T20-T18	4.12	T20-T19	1	T20-T20	0
T20-T21	1	T20-T22	2	T20-T23	3	T20-T24	4	T20-T25	1.41
T20-T26	1	T20-T27	1.41	T20-T28	2.24	T20-T29	3.16	T20-T30	4.12
T21-T1	3.61	T21-T2	3.16	T21-T3	3	T21-T4	3.16	T21-T5	3.61
T21-T6	4.24	T21-T7	2.83	T21-T8	2.24	T21-T9	2	T21-T10	2.24
T21-T11	2.83	T21-T12	3.61	T21-T13	2.24	T21-T14	1.41	T21-T15	1
T21-T16	1.41	T21-T17	2.24	T21-T18	3.16	T21-T19	2	T21-T20	1
T21-T21	0	T21-T22	1	T21-T23	2	T21-T24	3	T21-T25	2.24
T21-T26	1.41	T21-T27	1	T21-T28	1.41	T21-T29	2.24	T21-T30	3.16
T22-T1	4.24	T22-T2	3.61	T22-T3	3.16	T22-T4	3	T22-T5	3.16
T22-T6	3.61	T22-T7	3.61	T22-T8	2.83	T22-T9	2.24	T22-T10	2
T22-T11	2.24	T22-T12	2.83	T22-T13	3.16	T22-T14	2.24	T22-T15	1.41
T22-T16	1	T22-T17	1.41	T22-T18	2.24	T22-T19	3	T22-T20	2
T22-T21	1	T22-T22	0	T22-T23	1	T22-T24	2	T22-T25	3.16
T22-T26	2.24	T22-T27	1.41	T22-T28	1	T22-T29	1.41	T22-T30	2.24
T23-T1	5	T23-T2	4.24	T23-T3	3.61	T23-T4	3.16	T23-T5	3

T23-T6	3.16	T23-T7	4.47	T23-T8	3.61	T23-T9	2.83	T23-T10	2.24
T23-T11	2	T23-T12	2.24	T23-T13	4.12	T23-T14	3.16	T23-T15	2.24
T23-T16	1.41	T23-T17	1	T23-T18	1.41	T23-T19	4	T23-T20	3
T23-T21	2	T23-T22	1	T23-T23	0	T23-T24	1	T23-T25	4.12
T23-T26	3.16	T23-T27	2.24	T23-T28	1.41	T23-T29	1	T23-T30	1.41
T24-T1	5.83	T24-T2	5	T24-T3	4.24	T24-T4	3.61	T24-T5	3.16
T24-T6	3	T24-T7	5.39	T24-T8	4.47	T24-T9	3.61	T24-T10	2.83
T24-T11	2.24	T24-T12	2	T24-T13	5.1	T24-T14	4.12	T24-T15	3.16
T24-T16	2.24	T24-T17	1.41	T24-T18	1	T24-T19	5	T24-T20	4
T24-T21	3	T24-T22	2	T24-T23	1	T24-T24	0	T24-T25	5.1
T24-T26	4.12	T24-T27	3.16	T24-T28	2.24	T24-T29	1.41	T24-T30	1
T25-T1	4	T25-T2	4.12	T25-T3	4.47	T25-T4	5	T25-T5	5.66
T25-T6	6.4	T25-T7	3	T25-T8	3.16	T25-T9	3.61	T25-T10	4.24
T25-T11	5	T25-T12	5.83	T25-T13	2	T25-T14	2.24	T25-T15	2.83
T25-T16	3.61	T25-T17	4.47	T25-T18	5.39	T25-T19	1	T25-T20	1.41
T25-T21	2.24	T25-T22	3.16	T25-T23	4.12	T25-T24	5.1	T25-T25	0
T25-T26	1	T25-T27	2	T25-T28	3	T25-T29	4	T25-T30	5
T26-T1	4.12	T26-T2	4	T26-T3	4.12	T26-T4	4.47	T26-T5	5
T26-T6	5.66	T26-T7	3.16	T26-T8	3	T26-T9	3.16	T26-T10	3.61
T26-T11	4.24	T26-T12	5	T26-T13	2.24	T26-T14	2	T26-T15	2.24
T26-T16	2.83	T26-T17	3.61	T26-T18	4.47	T26-T19	1.41	T26-T20	1
T26-T21	1.41	T26-T22	2.24	T26-T23	3.16	T26-T24	4.12	T26-T25	1
T26-T26	0	T26-T27	1	T26-T28	2	T26-T29	3	T26-T30	4
T27-T1	4.47	T27-T2	4.12	T27-T3	4	T27-T4	4.12	T27-T5	4.47
T27-T6	5	T27-T7	3.61	T27-T8	3.16	T27-T9	3	T27-T10	3.16
T27-T11	3.61	T27-T12	4.24	T27-T13	2.83	T27-T14	2.24	T27-T15	2
T27-T16	2.24	T27-T17	2.83	T27-T18	3.61	T27-T19	2.24	T27-T20	1.41
T27-T21	1	T27-T22	1.41	T27-T23	2.24	T27-T24	3.16	T27-T25	2
T27-T26	1	T27-T27	0	T27-T28	1	T27-T29	2	T27-T30	3
T28-T1	5	T28-T2	4.47	T28-T3	4.12	T28-T4	4	T28-T5	4.12
T28-T6	4.47	T28-T7	4.24	T28-T8	3.61	T28-T9	3.16	T28-T10	3
T28-T11	3.16	T28-T12	3.61	T28-T13	3.61	T28-T14	2.83	T28-T15	2.24
T28-T16	2	T28-T17	2.24	T28-T18	2.83	T28-T19	3.16	T28-T20	2.24
T28-T21	1.41	T28-T22	1	T28-T23	1.41	T28-T24	2.24	T28-T25	3
T28-T26	2	T28-T27	1	T28-T28	0	T28-T29	1	T28-T30	2
T29-T1	5.66	T29-T2	5	T29-T3	4.47	T29-T4	4.12	T29-T5	4
T29-T6	4.12	T29-T7	5	T29-T8	4.24	T29-T9	3.61	T29-T10	3.16
T29-T11	3	T29-T12	3.16	T29-T13	4.47	T29-T14	3.61	T29-T15	2.83
T29-T16	2.24	T29-T17	2	T29-T18	2.24	T29-T19	4.12	T29-T20	3.16
T29-T21	2.24	T29-T22	1.41	T29-T23	1	T29-T24	1.41	T29-T25	4
T29-T26	3	T29-T27	2	T29-T28	1	T29-T29	0	T29-T30	1
T30-T1	6.4	T30-T2	5.66	T30-T3	5	T30-T4	4.47	T30-T5	4.12
T30-T6	4	T30-T7	5.83	T30-T8	5	T30-T9	4.24	T30-T10	3.61
T30-T11	3.16	T30-T12	3	T30-T13	5.39	T30-T14	4.47	T30-T15	3.61
T30-T16	2.83	T30-T17	2.24	T30-T18	2	T30-T19	5.1	T30-T20	4.12
T30-T21	3.16	T30-T22	2.24	T30-T23	1.41	T30-T24	1	T30-T25	5
T30-T26	4	T30-T27	3	T30-T28	2	T30-T29	1	T30-T30	0

5x7-Layout

- numberofkeys = 35, Rows = 5, Columns = 7

Tabelle 57: Tastenbelegungen für das 5x7-Layout

Belegung	Input-Daten
OPTI I	(19 15 5 26 18 2 32 12 24 29 6 33 4 25 10 30 1 17 16 11 3 31 20 21 34 7)

Tabelle 58: Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 5x7-Layout

T1	3.04	T2	2.06	T3	1.12	T4	0.5	T5	1.12	T6	2.06	T7	3.04
T8	3.35	T9	2.5	T10	1.8	T11	1.5	T12	1.8	T13	2.5	T14	3.35
T15	3.91	T16	3.2	T17	2.69	T18	2.5	T19	2.69	T20	3.2	T21	3.91
T22	4.61	T23	4.03	T24	3.64	T25	3.5	T26	3.64	T27	4.03	T28	4.61
T29	5.41	T30	4.92	T31	4.61	T32	4.5	T33	4.61	T34	4.92	T35	5.41

Tabelle 59: Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 5x7-Layout

T1	4.03	T2	4.61	T3	5.32	T4	6.1	T5	5.32	T6	4.61	T7	4.03
T8	3.2	T9	3.91	T10	4.72	T11	5.59	T12	4.72	T13	3.91	T14	3.2
T15	2.5	T16	3.35	T17	4.27	T18	5.22	T19	4.27	T20	3.35	T21	2.5
T22	2.06	T23	3.04	T24	4.03	T25	5.02	T26	4.03	T27	3.04	T28	2.06
T29	2.06	T30	3.04	T31	4.03	T32	5.02	T33	4.03	T34	3.04	T35	2.06

Tabelle 60: Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 5x7-Layout

T1	2.06	T2	2.06	T3	2.5	T4	3.2	T5	2.5	T6	2.06	T7	2.06
T8	1.12	T9	1.12	T10	1.8	T11	2.69	T12	1.8	T13	1.12	T14	1.12
T15	0.5	T16	0.5	T17	1.5	T18	2.5	T19	1.5	T20	0.5	T21	0.5
T22	1.12	T23	1.12	T24	1.8	T25	2.69	T26	1.8	T27	1.12	T28	1.12
T29	2.06	T30	2.06	T31	2.5	T32	3.2	T33	2.5	T34	2.06	T35	2.06

Tabelle 61: Distanzen d_{jl} zwischen den Tasten für das vertikale 5x7-Layout

T1-T1	0	T1-T2	1	T1-T3	2	T1-T4	3	T1-T5	4
T1-T6	5	T1-T7	6	T1-T8	1	T1-T9	1.41	T1-T10	2.24
T1-T11	3.16	T1-T12	4.12	T1-T13	5.1	T1-T14	6.08	T1-T15	2
T1-T16	2.24	T1-T17	2.83	T1-T18	3.61	T1-T19	4.47	T1-T20	5.39
T1-T21	6.32	T1-T22	3	T1-T23	3.16	T1-T24	3.61	T1-T25	4.24
T1-T26	5	T1-T27	5.83	T1-T28	6.71	T1-T29	4	T1-T30	4.12
T1-T31	4.47	T1-T32	5	T1-T33	5.66	T1-T34	6.4	T1-T35	7.21
T2-T1	1	T2-T2	0	T2-T3	1	T2-T4	2	T2-T5	3
T2-T6	4	T2-T7	5	T2-T8	1.41	T2-T9	1	T2-T10	1.41
T2-T11	2.24	T2-T12	3.16	T2-T13	4.12	T2-T14	5.1	T2-T15	2.24
T2-T16	2	T2-T17	2.24	T2-T18	2.83	T2-T19	3.61	T2-T20	4.47
T2-T21	5.39	T2-T22	3.16	T2-T23	3	T2-T24	3.16	T2-T25	3.61
T2-T26	4.24	T2-T27	5	T2-T28	5.83	T2-T29	4.12	T2-T30	4
T2-T31	4.12	T2-T32	4.47	T2-T33	5	T2-T34	5.66	T2-T35	6.4
T3-T1	2	T3-T2	1	T3-T3	0	T3-T4	1	T3-T5	2
T3-T6	3	T3-T7	4	T3-T8	2.24	T3-T9	1.41	T3-T10	1
T3-T11	1.41	T3-T12	2.24	T3-T13	3.16	T3-T14	4.12	T3-T15	2.83
T3-T16	2.24	T3-T17	2	T3-T18	2.24	T3-T19	2.83	T3-T20	3.61
T3-T21	4.47	T3-T22	3.61	T3-T23	3.16	T3-T24	3	T3-T25	3.16
T3-T26	3.61	T3-T27	4.24	T3-T28	5	T3-T29	4.47	T3-T30	4.12
T3-T31	4	T3-T32	4.12	T3-T33	4.47	T3-T34	5	T3-T35	5.66
T4-T1	3	T4-T2	2	T4-T3	1	T4-T4	0	T4-T5	1
T4-T6	2	T4-T7	3	T4-T8	3.16	T4-T9	2.24	T4-T10	1.41
T4-T11	1	T4-T12	1.41	T4-T13	2.24	T4-T14	3.16	T4-T15	3.61
T4-T16	2.83	T4-T17	2.24	T4-T18	2	T4-T19	2.24	T4-T20	2.83
T4-T21	3.61	T4-T22	4.24	T4-T23	3.61	T4-T24	3.16	T4-T25	3
T4-T26	3.16	T4-T27	3.61	T4-T28	4.24	T4-T29	5	T4-T30	4.47
T4-T31	4.12	T4-T32	4	T4-T33	4.12	T4-T34	4.47	T4-T35	5
T5-T1	4	T5-T2	3	T5-T3	2	T5-T4	1	T5-T5	0
T5-T6	1	T5-T7	2	T5-T8	4.12	T5-T9	3.16	T5-T10	2.24
T5-T11	1.41	T5-T12	1	T5-T13	1.41	T5-T14	2.24	T5-T15	4.47
T5-T16	3.61	T5-T17	2.83	T5-T18	2.24	T5-T19	2	T5-T20	2.24
T5-T21	2.83	T5-T22	5	T5-T23	4.24	T5-T24	3.61	T5-T25	3.16
T5-T26	3	T5-T27	3.16	T5-T28	3.61	T5-T29	5.66	T5-T30	5
T5-T31	4.47	T5-T32	4.12	T5-T33	4	T5-T34	4.12	T5-T35	4.47
T6-T1	5	T6-T2	4	T6-T3	3	T6-T4	2	T6-T5	1
T6-T6	0	T6-T7	1	T6-T8	5.1	T6-T9	4.12	T6-T10	3.16
T6-T11	2.24	T6-T12	1.41	T6-T13	1	T6-T14	1.41	T6-T15	5.39
T6-T16	4.47	T6-T17	3.61	T6-T18	2.83	T6-T19	2.24	T6-T20	2
T6-T21	2.24	T6-T22	5.83	T6-T23	5	T6-T24	4.24	T6-T25	3.61
T6-T26	3.16	T6-T27	3	T6-T28	3.16	T6-T29	6.4	T6-T30	5.66
T6-T31	5	T6-T32	4.47	T6-T33	4.12	T6-T34	4	T6-T35	4.12
T7-T1	6	T7-T2	5	T7-T3	4	T7-T4	3	T7-T5	2

A Input-Daten

T7-T6	1	T7-T7	0	T7-T8	6.08	T7-T9	5.1	T7-T10	4.12
T7-T11	3.16	T7-T12	2.24	T7-T13	1.41	T7-T14	1	T7-T15	6.32
T7-T16	5.39	T7-T17	4.47	T7-T18	3.61	T7-T19	2.83	T7-T20	2.24
T7-T21	2	T7-T22	6.71	T7-T23	5.83	T7-T24	5	T7-T25	4.24
T7-T26	3.61	T7-T27	3.16	T7-T28	3	T7-T29	7.21	T7-T30	6.4
T7-T31	5.66	T7-T32	5	T7-T33	4.47	T7-T34	4.12	T7-T35	4
T8-T1	1	T8-T2	1.41	T8-T3	2.24	T8-T4	3.16	T8-T5	4.12
T8-T6	5.1	T8-T7	6.08	T8-T8	0	T8-T9	1	T8-T10	2
T8-T11	3	T8-T12	4	T8-T13	5	T8-T14	6	T8-T15	1
T8-T16	1.41	T8-T17	2.24	T8-T18	3.16	T8-T19	4.12	T8-T20	5.1
T8-T21	6.08	T8-T22	2	T8-T23	2.24	T8-T24	2.83	T8-T25	3.61
T8-T26	4.47	T8-T27	5.39	T8-T28	6.32	T8-T29	3	T8-T30	3.16
T8-T31	3.61	T8-T32	4.24	T8-T33	5	T8-T34	5.83	T8-T35	6.71
T9-T1	1.41	T9-T2	1	T9-T3	1.41	T9-T4	2.24	T9-T5	3.16
T9-T6	4.12	T9-T7	5.1	T9-T8	1	T9-T9	0	T9-T10	1
T9-T11	2	T9-T12	3	T9-T13	4	T9-T14	5	T9-T15	1.41
T9-T16	1	T9-T17	1.41	T9-T18	2.24	T9-T19	3.16	T9-T20	4.12
T9-T21	5.1	T9-T22	2.24	T9-T23	2	T9-T24	2.24	T9-T25	2.83
T9-T26	3.61	T9-T27	4.47	T9-T28	5.39	T9-T29	3.16	T9-T30	3
T9-T31	3.16	T9-T32	3.61	T9-T33	4.24	T9-T34	5	T9-T35	5.83
T10-T1	2.24	T10-T2	1.41	T10-T3	1	T10-T4	1.41	T10-T5	2.24
T10-T6	3.16	T10-T7	4.12	T10-T8	2	T10-T9	1	T10-T10	0
T10-T11	1	T10-T12	2	T10-T13	3	T10-T14	4	T10-T15	2.24
T10-T16	1.41	T10-T17	1	T10-T18	1.41	T10-T19	2.24	T10-T20	3.16
T10-T21	4.12	T10-T22	2.83	T10-T23	2.24	T10-T24	2	T10-T25	2.24
T10-T26	2.83	T10-T27	3.61	T10-T28	4.47	T10-T29	3.61	T10-T30	3.16
T10-T31	3	T10-T32	3.16	T10-T33	3.61	T10-T34	4.24	T10-T35	5
T11-T1	3.16	T11-T2	2.24	T11-T3	1.41	T11-T4	1	T11-T5	1.41
T11-T6	2.24	T11-T7	3.16	T11-T8	3	T11-T9	2	T11-T10	1
T11-T11	0	T11-T12	1	T11-T13	2	T11-T14	3	T11-T15	3.16
T11-T16	2.24	T11-T17	1.41	T11-T18	1	T11-T19	1.41	T11-T20	2.24
T11-T21	3.16	T11-T22	3.61	T11-T23	2.83	T11-T24	2.24	T11-T25	2
T11-T26	2.24	T11-T27	2.83	T11-T28	3.61	T11-T29	4.24	T11-T30	3.61
T11-T31	3.16	T11-T32	3	T11-T33	3.16	T11-T34	3.61	T11-T35	4.24
T12-T1	4.12	T12-T2	3.16	T12-T3	2.24	T12-T4	1.41	T12-T5	1
T12-T6	1.41	T12-T7	2.24	T12-T8	4	T12-T9	3	T12-T10	2
T12-T11	1	T12-T12	0	T12-T13	1	T12-T14	2	T12-T15	4.12
T12-T16	3.16	T12-T17	2.24	T12-T18	1.41	T12-T19	1	T12-T20	1.41
T12-T21	2.24	T12-T22	4.47	T12-T23	3.61	T12-T24	2.83	T12-T25	2.24
T12-T26	2	T12-T27	2.24	T12-T28	2.83	T12-T29	5	T12-T30	4.24
T12-T31	3.61	T12-T32	3.16	T12-T33	3	T12-T34	3.16	T12-T35	3.61
T13-T1	5.1	T13-T2	4.12	T13-T3	3.16	T13-T4	2.24	T13-T5	1.41
T13-T6	1	T13-T7	1.41	T13-T8	5	T13-T9	4	T13-T10	3
T13-T11	2	T13-T12	1	T13-T13	0	T13-T14	1	T13-T15	5.1
T13-T16	4.12	T13-T17	3.16	T13-T18	2.24	T13-T19	1.41	T13-T20	1
T13-T21	1.41	T13-T22	5.39	T13-T23	4.47	T13-T24	3.61	T13-T25	2.83

T13-T26	2.24	T13-T27	2	T13-T28	2.24	T13-T29	5.83	T13-T30	5
T13-T31	4.24	T13-T32	3.61	T13-T33	3.16	T13-T34	3	T13-T35	3.16
T14-T1	6.08	T14-T2	5.1	T14-T3	4.12	T14-T4	3.16	T14-T5	2.24
T14-T6	1.41	T14-T7	1	T14-T8	6	T14-T9	5	T14-T10	4
T14-T11	3	T14-T12	2	T14-T13	1	T14-T14	0	T14-T15	6.08
T14-T16	5.1	T14-T17	4.12	T14-T18	3.16	T14-T19	2.24	T14-T20	1.41
T14-T21	1	T14-T22	6.32	T14-T23	5.39	T14-T24	4.47	T14-T25	3.61
T14-T26	2.83	T14-T27	2.24	T14-T28	2	T14-T29	6.71	T14-T30	5.83
T14-T31	5	T14-T32	4.24	T14-T33	3.61	T14-T34	3.16	T14-T35	3
T15-T1	2	T15-T2	2.24	T15-T3	2.83	T15-T4	3.61	T15-T5	4.47
T15-T6	5.39	T15-T7	6.32	T15-T8	1	T15-T9	1.41	T15-T10	2.24
T15-T11	3.16	T15-T12	4.12	T15-T13	5.1	T15-T14	6.08	T15-T15	0
T15-T16	1	T15-T17	2	T15-T18	3	T15-T19	4	T15-T20	5
T15-T21	6	T15-T22	1	T15-T23	1.41	T15-T24	2.24	T15-T25	3.16
T15-T26	4.12	T15-T27	5.1	T15-T28	6.08	T15-T29	2	T15-T30	2.24
T15-T31	2.83	T15-T32	3.61	T15-T33	4.47	T15-T34	5.39	T15-T35	6.32
T16-T1	2.24	T16-T2	2	T16-T3	2.24	T16-T4	2.83	T16-T5	3.61
T16-T6	4.47	T16-T7	5.39	T16-T8	1.41	T16-T9	1	T16-T10	1.41
T16-T11	2.24	T16-T12	3.16	T16-T13	4.12	T16-T14	5.1	T16-T15	1
T16-T16	0	T16-T17	1	T16-T18	2	T16-T19	3	T16-T20	4
T16-T21	5	T16-T22	1.41	T16-T23	1	T16-T24	1.41	T16-T25	2.24
T16-T26	3.16	T16-T27	4.12	T16-T28	5.1	T16-T29	2.24	T16-T30	2
T16-T31	2.24	T16-T32	2.83	T16-T33	3.61	T16-T34	4.47	T16-T35	5.39
T17-T1	2.83	T17-T2	2.24	T17-T3	2	T17-T4	2.24	T17-T5	2.83
T17-T6	3.61	T17-T7	4.47	T17-T8	2.24	T17-T9	1.41	T17-T10	1
T17-T11	1.41	T17-T12	2.24	T17-T13	3.16	T17-T14	4.12	T17-T15	2
T17-T16	1	T17-T17	0	T17-T18	1	T17-T19	2	T17-T20	3
T17-T21	4	T17-T22	2.24	T17-T23	1.41	T17-T24	1	T17-T25	1.41
T17-T26	2.24	T17-T27	3.16	T17-T28	4.12	T17-T29	2.83	T17-T30	2.24
T17-T31	2	T17-T32	2.24	T17-T33	2.83	T17-T34	3.61	T17-T35	4.47
T18-T1	3.61	T18-T2	2.83	T18-T3	2.24	T18-T4	2	T18-T5	2.24
T18-T6	2.83	T18-T7	3.61	T18-T8	3.16	T18-T9	2.24	T18-T10	1.41
T18-T11	1	T18-T12	1.41	T18-T13	2.24	T18-T14	3.16	T18-T15	3
T18-T16	2	T18-T17	1	T18-T18	0	T18-T19	1	T18-T20	2
T18-T21	3	T18-T22	3.16	T18-T23	2.24	T18-T24	1.41	T18-T25	1
T18-T26	1.41	T18-T27	2.24	T18-T28	3.16	T18-T29	3.61	T18-T30	2.83
T18-T31	2.24	T18-T32	2	T18-T33	2.24	T18-T34	2.83	T18-T35	3.61
T19-T1	4.47	T19-T2	3.61	T19-T3	2.83	T19-T4	2.24	T19-T5	2
T19-T6	2.24	T19-T7	2.83	T19-T8	4.12	T19-T9	3.16	T19-T10	2.24
T19-T11	1.41	T19-T12	1	T19-T13	1.41	T19-T14	2.24	T19-T15	4
T19-T16	3	T19-T17	2	T19-T18	1	T19-T19	0	T19-T20	1
T19-T21	2	T19-T22	4.12	T19-T23	3.16	T19-T24	2.24	T19-T25	1.41
T19-T26	1	T19-T27	1.41	T19-T28	2.24	T19-T29	4.47	T19-T30	3.61
T19-T31	2.83	T19-T32	2.24	T19-T33	2	T19-T34	2.24	T19-T35	2.83
T20-T1	5.39	T20-T2	4.47	T20-T3	3.61	T20-T4	2.83	T20-T5	2.24
T20-T6	2	T20-T7	2.24	T20-T8	5.1	T20-T9	4.12	T20-T10	3.16

A Input-Daten

T20-T11	2.24	T20-T12	1.41	T20-T13	1	T20-T14	1.41	T20-T15	5
T20-T16	4	T20-T17	3	T20-T18	2	T20-T19	1	T20-T20	0
T20-T21	1	T20-T22	5.1	T20-T23	4.12	T20-T24	3.16	T20-T25	2.24
T20-T26	1.41	T20-T27	1	T20-T28	1.41	T20-T29	5.39	T20-T30	4.47
T20-T31	3.61	T20-T32	2.83	T20-T33	2.24	T20-T34	2	T20-T35	2.24
T21-T1	6.32	T21-T2	5.39	T21-T3	4.47	T21-T4	3.61	T21-T5	2.83
T21-T6	2.24	T21-T7	2	T21-T8	6.08	T21-T9	5.1	T21-T10	4.12
T21-T11	3.16	T21-T12	2.24	T21-T13	1.41	T21-T14	1	T21-T15	6
T21-T16	5	T21-T17	4	T21-T18	3	T21-T19	2	T21-T20	1
T21-T21	0	T21-T22	6.08	T21-T23	5.1	T21-T24	4.12	T21-T25	3.16
T21-T26	2.24	T21-T27	1.41	T21-T28	1	T21-T29	6.32	T21-T30	5.39
T21-T31	4.47	T21-T32	3.61	T21-T33	2.83	T21-T34	2.24	T21-T35	2
T22-T1	3	T22-T2	3.16	T22-T3	3.61	T22-T4	4.24	T22-T5	5
T22-T6	5.83	T22-T7	6.71	T22-T8	2	T22-T9	2.24	T22-T10	2.83
T22-T11	3.61	T22-T12	4.47	T22-T13	5.39	T22-T14	6.32	T22-T15	1
T22-T16	1.41	T22-T17	2.24	T22-T18	3.16	T22-T19	4.12	T22-T20	5.1
T22-T21	6.08	T22-T22	0	T22-T23	1	T22-T24	2	T22-T25	3
T22-T26	4	T22-T27	5	T22-T28	6	T22-T29	1	T22-T30	1.41
T22-T31	2.24	T22-T32	3.16	T22-T33	4.12	T22-T34	5.1	T22-T35	6.08
T23-T1	3.16	T23-T2	3	T23-T3	3.16	T23-T4	3.61	T23-T5	4.24
T23-T6	5	T23-T7	5.83	T23-T8	2.24	T23-T9	2	T23-T10	2.24
T23-T11	2.83	T23-T12	3.61	T23-T13	4.47	T23-T14	5.39	T23-T15	1.41
T23-T16	1	T23-T17	1.41	T23-T18	2.24	T23-T19	3.16	T23-T20	4.12
T23-T21	5.1	T23-T22	1	T23-T23	0	T23-T24	1	T23-T25	2
T23-T26	3	T23-T27	4	T23-T28	5	T23-T29	1.41	T23-T30	1
T23-T31	1.41	T23-T32	2.24	T23-T33	3.16	T23-T34	4.12	T23-T35	5.1
T24-T1	3.61	T24-T2	3.16	T24-T3	3	T24-T4	3.16	T24-T5	3.61
T24-T6	4.24	T24-T7	5	T24-T8	2.83	T24-T9	2.24	T24-T10	2
T24-T11	2.24	T24-T12	2.83	T24-T13	3.61	T24-T14	4.47	T24-T15	2.24
T24-T16	1.41	T24-T17	1	T24-T18	1.41	T24-T19	2.24	T24-T20	3.16
T24-T21	4.12	T24-T22	2	T24-T23	1	T24-T24	0	T24-T25	1
T24-T26	2	T24-T27	3	T24-T28	4	T24-T29	2.24	T24-T30	1.41
T24-T31	1	T24-T32	1.41	T24-T33	2.24	T24-T34	3.16	T24-T35	4.12
T25-T1	4.24	T25-T2	3.61	T25-T3	3.16	T25-T4	3	T25-T5	3.16
T25-T6	3.61	T25-T7	4.24	T25-T8	3.61	T25-T9	2.83	T25-T10	2.24
T25-T11	2	T25-T12	2.24	T25-T13	2.83	T25-T14	3.61	T25-T15	3.16
T25-T16	2.24	T25-T17	1.41	T25-T18	1	T25-T19	1.41	T25-T20	2.24
T25-T21	3.16	T25-T22	3	T25-T23	2	T25-T24	1	T25-T25	0
T25-T26	1	T25-T27	2	T25-T28	3	T25-T29	3.16	T25-T30	2.24
T25-T31	1.41	T25-T32	1	T25-T33	1.41	T25-T34	2.24	T25-T35	3.16
T26-T1	5	T26-T2	4.24	T26-T3	3.61	T26-T4	3.16	T26-T5	3
T26-T6	3.16	T26-T7	3.61	T26-T8	4.47	T26-T9	3.61	T26-T10	2.83
T26-T11	2.24	T26-T12	2	T26-T13	2.24	T26-T14	2.83	T26-T15	4.12
T26-T16	3.16	T26-T17	2.24	T26-T18	1.41	T26-T19	1	T26-T20	1.41
T26-T21	2.24	T26-T22	4	T26-T23	3	T26-T24	2	T26-T25	1
T26-T26	0	T26-T27	1	T26-T28	2	T26-T29	4.12	T26-T30	3.16

T26-T31	2.24	T26-T32	1.41	T26-T33	1	T26-T34	1.41	T26-T35	2.24
T27-T1	5.83	T27-T2	5	T27-T3	4.24	T27-T4	3.61	T27-T5	3.16
T27-T6	3	T27-T7	3.16	T27-T8	5.39	T27-T9	4.47	T27-T10	3.61
T27-T11	2.83	T27-T12	2.24	T27-T13	2	T27-T14	2.24	T27-T15	5.1
T27-T16	4.12	T27-T17	3.16	T27-T18	2.24	T27-T19	1.41	T27-T20	1
T27-T21	1.41	T27-T22	5	T27-T23	4	T27-T24	3	T27-T25	2
T27-T26	1	T27-T27	0	T27-T28	1	T27-T29	5.1	T27-T30	4.12
T27-T31	3.16	T27-T32	2.24	T27-T33	1.41	T27-T34	1	T27-T35	1.41
T28-T1	6.71	T28-T2	5.83	T28-T3	5	T28-T4	4.24	T28-T5	3.61
T28-T6	3.16	T28-T7	3	T28-T8	6.32	T28-T9	5.39	T28-T10	4.47
T28-T11	3.61	T28-T12	2.83	T28-T13	2.24	T28-T14	2	T28-T15	6.08
T28-T16	5.1	T28-T17	4.12	T28-T18	3.16	T28-T19	2.24	T28-T20	1.41
T28-T21	1	T28-T22	6	T28-T23	5	T28-T24	4	T28-T25	3
T28-T26	2	T28-T27	1	T28-T28	0	T28-T29	6.08	T28-T30	5.1
T28-T31	4.12	T28-T32	3.16	T28-T33	2.24	T28-T34	1.41	T28-T35	1
T29-T1	4	T29-T2	4.12	T29-T3	4.47	T29-T4	5	T29-T5	5.66
T29-T6	6.4	T29-T7	7.21	T29-T8	3	T29-T9	3.16	T29-T10	3.61
T29-T11	4.24	T29-T12	5	T29-T13	5.83	T29-T14	6.71	T29-T15	2
T29-T16	2.24	T29-T17	2.83	T29-T18	3.61	T29-T19	4.47	T29-T20	5.39
T29-T21	6.32	T29-T22	1	T29-T23	1.41	T29-T24	2.24	T29-T25	3.16
T29-T26	4.12	T29-T27	5.1	T29-T28	6.08	T29-T29	0	T29-T30	1
T29-T31	2	T29-T32	3	T29-T33	4	T29-T34	5	T29-T35	6
T30-T1	4.12	T30-T2	4	T30-T3	4.12	T30-T4	4.47	T30-T5	5
T30-T6	5.66	T30-T7	6.4	T30-T8	3.16	T30-T9	3	T30-T10	3.16
T30-T11	3.61	T30-T12	4.24	T30-T13	5	T30-T14	5.83	T30-T15	2.24
T30-T16	2	T30-T17	2.24	T30-T18	2.83	T30-T19	3.61	T30-T20	4.47
T30-T21	5.39	T30-T22	1.41	T30-T23	1	T30-T24	1.41	T30-T25	2.24
T30-T26	3.16	T30-T27	4.12	T30-T28	5.1	T30-T29	1	T30-T30	0
T30-T31	1	T30-T32	2	T30-T33	3	T30-T34	4	T30-T35	5
T31-T1	4.47	T31-T2	4.12	T31-T3	4	T31-T4	4.12	T31-T5	4.47
T31-T6	5	T31-T7	5.66	T31-T8	3.61	T31-T9	3.16	T31-T10	3
T31-T11	3.16	T31-T12	3.61	T31-T13	4.24	T31-T14	5	T31-T15	2.83
T31-T16	2.24	T31-T17	2	T31-T18	2.24	T31-T19	2.83	T31-T20	3.61
T31-T21	4.47	T31-T22	2.24	T31-T23	1.41	T31-T24	1	T31-T25	1.41
T31-T26	2.24	T31-T27	3.16	T31-T28	4.12	T31-T29	2	T31-T30	1
T31-T31	0	T31-T32	1	T31-T33	2	T31-T34	3	T31-T35	4
T32-T1	5	T32-T2	4.47	T32-T3	4.12	T32-T4	4	T32-T5	4.12
T32-T6	4.47	T32-T7	5	T32-T8	4.24	T32-T9	3.61	T32-T10	3.16
T32-T11	3	T32-T12	3.16	T32-T13	3.61	T32-T14	4.24	T32-T15	3.61
T32-T16	2.83	T32-T17	2.24	T32-T18	2	T32-T19	2.24	T32-T20	2.83
T32-T21	3.61	T32-T22	3.16	T32-T23	2.24	T32-T24	1.41	T32-T25	1
T32-T26	1.41	T32-T27	2.24	T32-T28	3.16	T32-T29	3	T32-T30	2
T32-T31	1	T32-T32	0	T32-T33	1	T32-T34	2	T32-T35	3
T33-T1	5.66	T33-T2	5	T33-T3	4.47	T33-T4	4.12	T33-T5	4
T33-T6	4.12	T33-T7	4.47	T33-T8	5	T33-T9	4.24	T33-T10	3.61
T33-T11	3.16	T33-T12	3	T33-T13	3.16	T33-T14	3.61	T33-T15	4.47

T33-T16	3.61	T33-T17	2.83	T33-T18	2.24	T33-T19	2	T33-T20	2.24
T33-T21	2.83	T33-T22	4.12	T33-T23	3.16	T33-T24	2.24	T33-T25	1.41
T33-T26	1	T33-T27	1.41	T33-T28	2.24	T33-T29	4	T33-T30	3
T33-T31	2	T33-T32	1	T33-T33	0	T33-T34	1	T33-T35	2
T34-T1	6.4	T34-T2	5.66	T34-T3	5	T34-T4	4.47	T34-T5	4.12
T34-T6	4	T34-T7	4.12	T34-T8	5.83	T34-T9	5	T34-T10	4.24
T34-T11	3.61	T34-T12	3.16	T34-T13	3	T34-T14	3.16	T34-T15	5.39
T34-T16	4.47	T34-T17	3.61	T34-T18	2.83	T34-T19	2.24	T34-T20	2
T34-T21	2.24	T34-T22	5.1	T34-T23	4.12	T34-T24	3.16	T34-T25	2.24
T34-T26	1.41	T34-T27	1	T34-T28	1.41	T34-T29	5	T34-T30	4
T34-T31	3	T34-T32	2	T34-T33	1	T34-T34	0	T34-T35	1
T35-T1	7.21	T35-T2	6.4	T35-T3	5.66	T35-T4	5	T35-T5	4.47
T35-T6	4.12	T35-T7	4	T35-T8	6.71	T35-T9	5.83	T35-T10	5
T35-T11	4.24	T35-T12	3.61	T35-T13	3.16	T35-T14	3	T35-T15	6.32
T35-T16	5.39	T35-T17	4.47	T35-T18	3.61	T35-T19	2.83	T35-T20	2.24
T35-T21	2	T35-T22	6.08	T35-T23	5.1	T35-T24	4.12	T35-T25	3.16
T35-T26	2.24	T35-T27	1.41	T35-T28	1	T35-T29	6	T35-T30	5
T35-T31	4	T35-T32	3	T35-T33	2	T35-T34	1	T35-T35	0

B Quellcode

```
1 model KeyboardOptimization !Modellname
2 options noimplicit, explterm; !IVE Optionen
3 uses 'mxxprs', 'mmsystem', 'mmive'; !gain access to the Xpress-Optimizer solver
4 !PARAMETER
5 parameters
6 Task = 'optimization'; !Aufgabe des Programms (optimization, evaluation)
7 Objective = 'travel-distance'; !Zielfunktion (travel-distance, travel-time)
8 Position = 'vertical'; !Geraeteposition (vertical, horizontal, hybrid)
9 Writing_method = 'realistic'; !Schreibweise (simple,semi-realistic,realistic,hover-over)
10 datafile = 'QWERTY-like-Layout(26B).dat'; !Einzulesende Datei
11 !Grad der Ausgabe
12 MESSLEV = 3; !Message-Level (1,2,3,4,5,6)
13 !Grundlegende Parameter
14 tol = 0.000001; !Toleranz
15 aver_text_length =100; !Durchschnittliche Textlaenge
16 fitts_law_param_a =0.083; !Parameter a fuer Fitts Law
17 fitts_law_param_b =1/7.9; !Parameter b fuer Fitts Law
18 key_width =1; !Tastengroesse
19 alpha =0.5; !Anteil der vertikalen Schreibweise im Hybrid-Fall
20 beta =0.5; !Anteil der horiz. Schreibweise im Hybrid-Fall
21
22 !REDUKTIONSMETHODEN
23 PROB_HANDLING = 'threshold';
24 threshold = 0.6;
25 min_entries_per_row = 0;
26 global_best = 0;
27 max_entries_per_row = 0;
28 entries_per_row = 0;
29 !none = keine Verarbeitung, threshold = alle Eintraege unter einem Schwellenwert werden gleich Null gesetzt,
threshold_but_min_entries_per_row = wie threshold, aber es bleibt eine Mindestanzahl pro Zeile bestehen,
global_best = verwende die besten n Eintraege, global_best_but_max_entries_per_row = wie global_best aber nicht
mehr als n Eintraege aus einer Reihe, entries_per_row = die n besten Eintraege einer Reihe,
entries_per_row_plus_global_best = wie entries_per_row und zusätzlich die besten n der verbleibenden Eintraege
30
31 !OPTIMIERUNGSEINSTELLUNGEN FUER (semi-real.,real.,hover-over)
32 Objfot_extra_term = 'on'; !OBJFCT (zusätzlicher Term in der Zielfunktion)
33 Implicit_zero = 'on'; !IMPL_ZERO (implizites Nullsetzen)
34 X_LowB = 'on'; ! - (untere Schranke an x)
35 X_UpB = 'off'; !UPBND (obere Schranke an x)
36 AJ_Con = 'on'; !AJCON (Adams-Johnson-Nebenbedingungen)
37 Letter_Letter_Con = 'off'; !CUTS(1a)
38 Key_Key_Con = 'off'; !CUTS(1b)
39 X_Equ = 'on'; !CUTS(2)
40 XY_Equ = 'off'; !CUTS(3)
41 Letter_Key_Con = 'off'; !CUTS(4)
42 Binaryness = 'off'; !NOBin (Ganzzahligkeit von x)
43 Fix_Letter = 5; !SYMM (Buchstabe auf rechte Haelfte fixieren)
44
45 !POST-OPTIMIERUNGS-OPTIONEN
46 Heuristic_Iter = 0; !Anzahl der 2-opt-Iterationen
47 test_keyboard_in_alternative_situations = 'off'; !Eigenschaften der optimierten Tastatur fuer andere Position oder
Schreibweise betrachten
48 examine_own_keyboard = 'off'; !eigene Tastenbelegung nach der Optimierung auswerten
49 improve_own_keyboard = 'off'; !eigene Tastenbelegung verbessern
50 show_assignment_array = 'off'; !Belegungsvektoren anzeigen
51 end-parameters
52
53 !SOLVEREINSTELLUNGEN
54 !setparam('XPRS_presolve',0);
55 setparam('XPRS_cutstrategy', 0);
```

B Quellcode

```
56 setparam('XPRS_heurstrategy', 0);
57
58 !DEKLARATIONEN - I
59 declarations
60 Keyboard, Shape: string; !Name und Form der Tastatur
61 Rows: integer; !Anzahl der Zeilen der Tastatur
62 numberofletters, numberofkeys: integer; !Anzahl der Tasten,$ Anzahl der Buchstaben
63 Letters, Keys: range; !Vektoren der Buchstaben und Tasten
64 end-declarations
65
66 !EINLESEN - I
67 initializations from datafile
68 Keyboard; Shape; Rows; numberofletters; numberofkeys;
69 end-initializations
70
71 !DEFINITION DER ZUM EINLESEN BENOETIGTEN VARIABLEN
72 Letters :=1..numberofletters;
73 Keys :=1..numberofkeys;
74
75 !DEKLARATIONEN - II
76 declarations
77 !Die eingelesenen Daten
78 Lettername: array(Letters) of string; !Buchstabennamen
79 Keyname: array(Keys) of string; !Tastennamen
80 Columns_free: array(1..Rows) of integer; !Anzahl der Spalten bei einem freien Tastaturlayout
81 Columns_rect: integer; !Anzahl der Spalten bei einem rechteckigen Tastaturlayout
82 x: array(Letters,Keys,Letters,Keys) of mpvar; !Optimierungsvariablen - 1
83 y: array(Letters,Keys) of mpvar; !Optimierungsvariablen - 2
84 objfct: lincpr; !Zielfunktion
85 dist_keys_orig: array(Keys,Keys) of real; !urspruengliche Distanzen zwischen den Tasten
86 dist_start_vert_orig: array(Keys) of real; !urspruengliche Distanzen zum vertikalen Startpunkt
87 dist_start_hor_orig: array(Keys) of real; !urspruengliche Distanzen zum horizontalen Startpunkt
88 dist_mid_hor_orig: array(Keys) of real; !urspruengliche Distanzen zum Mittelpunkt
89 bigram_prob_orig: array(Letters,Letters) of real; !originale Bigramm-Wahrscheinlichkeiten
90 letter_prob_orig: array(Letters) of real; !originale Buchstaben-Wahrscheinlichkeiten
91 assignment_letters_keys: array(Letters) of integer; !eigene Belegung, welche ausgewertet werden soll
92
93 !Die Daten, die in der Optimierung verwendet werden
94 letter_prob: array(Letters) of real; !neue Buchstaben-Wahrscheinlichkeiten
95 bigram_prob: array(Letters,Letters) of real; !neue Bigramm-Wahrscheinlichkeiten
96 dist_keys: array(Keys,Keys) of real; !Distanzen zwischen den Tasten
97 dist_start: array(Keys) of real; !Distanzen zum Startpunkt
98 time_keys: array(Keys,Keys) of real; !Zeit zwischen den Tasten
99 time_start: array(Keys) of real; !Zeit zum Startpunkt
100
101 !Hilfsvariablen
102 individual_keyboard_vector: array(Letters,Keys) of real; !Vektor zur Auswertung der eigenen Tastenbelegung
103 Leftkeys, Rightkeys: set of integer; !Menge der linken, rechten Tasten
104 numberofconstraints_equ, numberofconstraints_inequ: integer; !Anzahl der Nebenbedingungen
105 numberofvar_x, numberofvar_y, numberofvar_implzero: integer; !Anzahl der Optimierungsvariablen
106 starttime, runtime: real; !Variablen zur Laufzeitbestimmung
107 error: boolean; !Fehlervariable
108 better_sol: boolean; !Variable fuer Heuristik
109 number_nonzero_entries, number_nonzero_entries_orig: integer; !Variablen fuer Analyse der
    Bigrammwahrscheinlichkeiten
110 final_solution: array(Letters,Keys) of real; !Finale Loesung
111 sol_optimization: array(Letters,Keys) of real; !Loesung der Optimierung
112 sol_heuristic: array(Letters,Keys) of real; !Loesung der Heuristik
113 end-declarations
114
115 !EINLESEN - II
116 initialize
117 Lettername; Keyname; Columns_free; Columns_rect; dist_keys_orig; dist_start_vert_orig;
118 dist_start_hor_orig; dist_mid_hor_orig; bigram_prob_orig; letter_prob_orig; assignment_letters_keys;
119 end-initializations
120
121 !DEFINITION DER FUNKTIONEN UND PROZEDUREN
122 forward procedure check_data
123 forward procedure check_setting
124 forward procedure create_dist_start
125 forward procedure create_time_start
126 forward procedure create_letter_prob
127 forward procedure assign_keys_to_side
```

```

128 forward procedure create_dist_keys
129 forward procedure create_time_keys
130 forward procedure create_probability_matrix
131 forward procedure show_probability_matrix(Matrix: array(Letters,Letters) of real)
132 forward procedure analyse_probability_matrix(Matrix: array(Letters,Letters) of real)
133 forward procedure show_dist_or_time_matrix(Matrix: array(Keys,Keys) of real)
134 forward procedure show_setting
135 forward procedure show_analysis
136 forward procedure show_model
137 forward procedure show_result
138 forward procedure show_solution_part
139 forward procedure show_solution_complete
140 forward procedure create_constraints
141 forward procedure create_objfct
142 forward function keyboard_quality_dist(sol:array(Letters,Keys) of real): real
143 forward function keyboard_quality_time(sol:array(Letters,Keys) of real): real
144 forward function keyboard_quality_wpm(sol:array(Letters,Keys) of real): real
145 forward function movement_time_key2point(j:integer,point: string): real
146 forward function movement_time_key2key(j:integer,l:integer): real
147 forward procedure visualize_keyboard(sol: array(Letters,Keys) of real)
148 forward procedure show_keyboard_characteristics(sol:array(Letters,Keys) of real)
149 forward function two_opt_heuristic(sol:array(Letters,Keys) of real): array(Letters,Keys) of real
150 forward function apply_heuristic(max_iter:integer, sol: array(Letters,Keys) of real): array(Letters,Keys) of real
151 forward procedure do_keyboard_test_in_alternative_situations(sol:array(Letters,Keys) of real)
152 forward function create_individual_keyboard_vector(assignment_set:array(Letters) of integer): array(Letters,Keys) of
    real
153 forward function create_assignment_array(sol:array(Letters,Keys) of real): array(Letters) of integer
154
155 !UEBERPRUEFE DIE DATEN UND EINSTELLUNGEN
156 error := false;
157 check_data;
158 check_setting;
159
160 !OPTIMIERUNG UND EVTL. AUSWERTUNG
161 if(Task = 'optimization') then
162     !INITIALISIERUNG DER VARIABLEN
163     starttime := gettime;
164     numberofconstraints_equ := 0;
165     numberofconstraints_inequ := 0;
166     numberofvar_x := 0;
167     numberofvar_y := 0;
168     numberofvar_implzero:= 0;
169     number_nonzero_entries := 0;
170     better_sol := false;
171
172     if(error = false) then
173         !TEILE DIE TASTATUR
174         if(Position <> 'simple' or Fix_Letter<>0 or test_keyboard_in_alternative_situations = 'on') then
175             assign_keys_to_side;
176         end-if
177
178         !ERZEUGE DIE DATEN FUER DIE OPTIMIERUNG
179         create_dist_start;
180         create_time_start;
181         create_letter_prob;
182         if(Writing_method <> 'simple') then
183             create_dist_keys;
184             create_time_keys;
185             create_probability_matrix;
186         end-if
187
188         !AUSGABE DER EINSTELLUNGEN UND ANALYSE DER WAHRSCHEINLICHKEITEN
189         show_setting;
190         show_analysis;
191         if (MESSLEV >=2) then
192             writeln(' ');
193             writeln('-----OPTIMIERUNG-----');
194         end-if
195
196         !ERZEUGE DIE NEBENBEDINGUNGEN
197         create_constraints;
198
199         !ERZEUGE DIE ZIELFUNKTION

```

B Quellcode

```
200 create_objfct;
201
202 !OPTIMIERUNG
203 minimize(objfct);
204
205 !OPTIMIERUNGSLOESUNG SPEICHERN
206 forall(i in Letters,j in Keys) do
207     sol_optimization(i,j) := y(i,j).sol;
208 end-do
209
210 !AUSGABE DER ERGEBNISSE
211 if (MESSLEV >=2) then
212     writeln('Optimierung abgeschlossen...');
213     show_result;
214     writeln('-----ENDE OPTIMIERUNG----');
215 end-if
216
217 !HEURISTIK
218 if(MESSLEV >=2) then
219     writeln(' ');
220     writeln('-----HEURISTIC-----');
221     if(Heuristic_Iter >0) then
222         sol_heuristic := apply_heuristic(Heuristic_Iter,sol_optimization);
223     else
224         writeln('Keine Heuristik verwendet');
225     end-if
226     writeln('-----END HEURISTIC----');
227 end-if
228
229 !GESAMTLOESUNG
230 writeln(' ');
231 writeln('-----SOLUTION-----');
232 if(better_sol = true) then
233     final_solution := sol_heuristic;
234 else
235     final_solution := sol_optimization;
236 end-if
237 writeln('Loesung:');
238 visualize_keyboard(final_solution);
239 if (show_assignment_array = 'on' and MESSLEV>=2) then
240     writeln(' ');
241     writeln('Finale Belegung: ',create_assignment_array(final_solution));
242     writeln(' ');
243 end-if
244 show_keyboard_characteristics(final_solution);
245 writeln('-----END SOLUTION-----');
246
247 !TESTE DIE OPTIMIERTE TASTENBELEGUNG IN VERSCHIEDENEN SITUATIONEN
248 if(test_keyboard_in_alternative_situations = 'on') then
249     writeln(' ');
250     writeln('-----EVALUATION-----');
251     do_keyboard_test_in_alternative_situations(final_solution);
252     writeln('-----END EVALUATION-----');
253 end-if
254
255 !UNTERSUCHE DIE EIGENE EINGELESENE TASTENBELEGUNG
256 if(examine_own_keyboard = 'on') then
257     writeln(' ');
258     writeln('-----OWN KEYBOARD-----');
259     if(MESSLEV>=3)then
260         writeln('Eingelesene Belegung: ',assignment_letters_keys);
261     end-if
262     writeln('Auswertung der selbst-erstellten Tastatur:');
263     individual_keyboard_vector := create_individual_keyboard_vector(assignment_letters_keys);
264     visualize_keyboard(individual_keyboard_vector);
265     show_keyboard_characteristics(individual_keyboard_vector);
266     if(test_keyboard_in_alternative_situations = 'on') then
267         writeln(' ');
268         do_keyboard_test_in_alternative_situations(individual_keyboard_vector);
269     end-if
270
271 !EVTL. HEURISTIK AUF EIGENE TASTENBELEGUNG ANWENDEN
272 if(improve_own_keyboard = 'on') then
```

```

273     writeln(' ');
274     better_sol := false;
275     sol_heuristic := apply_heuristic(Heuristic_Iter, individual_keyboard_vector);
276     if(better_sol = true) then
277         visualize_keyboard(sol_heuristic);
278         show_keyboard_characteristics(sol_heuristic);
279         if(test_keyboard_in_alternative_situations = 'on') then
280             writeln(' ');
281             do_keyboard_test_in_alternative_situations(sol_heuristic);
282         end-if
283     else
284         writeln(' Tastatur konnte nicht verbessert werden');
285     end-if
286 end-if
287 writeln('-----END OWN KEYBOARD-----');
288 end-if
289 writeln(' ');
290 writeln('Gesamte Laufzeit: ', gettime-starttime, 's');
291 else
292     writeln(' ');
293     writeln('Keine Optimierung: Bitte Fehler korrigieren!');
294 end-if
295
296 !NUR AUSWERTUNG; KEINE OPTIMIERUNG
297 elif(Task = 'evaluation') then
298
299 !TEILE DIE TASTATUR AUF
300 if(Position <> 'simple' or test_keyboard_in_alternative_situations = 'on') then
301     assign_keys_to_side;
302 end-if
303
304 !ERZEUGE DIE DATEN FUEER DIE AUSWERTUNG
305 create_dist_start;
306 create_time_start;
307 create_dist_keys;
308 create_time_keys;
309
310 !WERTE DIE EIGENE EINGELESENE TASTENBELEGUNG AUS
311 writeln('-----EVALUATION-----');
312 writeln('Auswertung der selbst-erstellten Tastatur:');
313 individual_keyboard_vector := create_individual_keyboard_vector(assignment_letters_keys);
314 visualize_keyboard(individual_keyboard_vector);
315 show_keyboard_characteristics(individual_keyboard_vector);
316 if(test_keyboard_in_alternative_situations = 'on') then
317     writeln(' ');
318     do_keyboard_test_in_alternative_situations(individual_keyboard_vector);
319 end-if
320 writeln('-----END EVALUATION-----');
321
322 !VERBESSERE DIE EIGENE TASTENBELEGUNG
323 if(improve_own_keyboard = 'on' and Heuristic_Iter>0) then
324     writeln(' ');
325     writeln('-----HEURISTIC-----');
326     better_sol := false;
327     sol_heuristic := apply_heuristic(Heuristic_Iter, individual_keyboard_vector);
328     if(better_sol = true) then
329         visualize_keyboard(sol_heuristic);
330         show_keyboard_characteristics(sol_heuristic);
331         if(test_keyboard_in_alternative_situations = 'on') then
332             writeln(' ');
333             do_keyboard_test_in_alternative_situations(sol_heuristic);
334         end-if
335     else
336         writeln(' Tastatur konnte nicht verbessert werden');
337     end-if
338     writeln('-----END HEURISTIC----');
339 end-if
340 end-if
341
342
343 !*****
344 !Prozeduren und Funktionen
345 !*****

```

B Quellcode

```
346
347 !*****
348 !Funktion um eingelesene Tastenbelegung in einen Vektor zu uebersetzen
349 function create_assignment_array(sol:array(Letters,Keys) of real): array(Letters) of integer
350   declarations
351     assignment: array(Letters) of integer;
352   end-declarations
353   forall(i in Letters) do
354     forall(j in Keys | abs(sol(i,j)-1)<tol) assignment(i) := j;
355   end-do
356   returned := assignment;
357 end-function
358
359 !*****
360 !Prozedur um die Eigenschaften einer Tastenbelegung auszugeben
361 procedure show_keyboard_characteristics(sol:array(Letters,Keys) of real)
362   writeln('Ergebnisse fuer die gewaehlten Optionen (' ,Writing_method,'$, $ ',Position,') und ',aver_text_length,'
363     Buchstaben: ');
364   writeln(' -Zurueckzulegende Distanz (LE): ',keyboard_quality_dist(sol));
365   writeln(' -Aufzuwendende Zeit (s): ',keyboard_quality_time(sol));
366   writeln('Theoretische Leistungsfahigkeit (wpm): ',keyboard_quality_wpm(sol));
367 end-procedure
368
369 !*****
370 !Prozedur um Tastenbelegung in verschiedenen Situationen zu testen
371 procedure do_keyboard_test_in_alternative_situations(sol:array(Letters,Keys) of real)
372   declarations
373     orig_Writing_method, orig_Position: string;
374     iter: integer;
375     possible_WM, alternative_Pos: set of string;
376   end-declarations
377   iter :=1;
378   orig_Writing_method := Writing_method;
379   orig_Position := Position;
380   if(Position = 'vertical') then
381     possible_WM := {'simple','realistic'};
382     alternative_Pos := {'horizontal','hybrid'};
383   elif(Position = 'horizontal') then
384     possible_WM := {'simple','semi-realistic','realistic','hover-over'};
385     alternative_Pos := {'vertical','hybrid'};
386   elif(Position = 'hybrid') then
387     possible_WM := {'realistic','hover-over'};
388     alternative_Pos := {'vertical','horizontal'};
389   end-if
390   forall(WM in possible_WM | WM <> orig_Writing_method) do
391     Writing_method := WM;
392     create_dist_start;
393     create_time_start;
394     create_dist_keys;
395     create_time_keys;
396     writeln(iter,'. alternative Verwendung (' ,Writing_method,'$, $ ',Position,'$, $ ',aver_text_length,' Buchstaben):');
397     writeln(' -Zurueckzulegende Distanz (LE): ',keyboard_quality_dist(sol));
398     writeln(' -Aufzuwendende Zeit (s): ',keyboard_quality_time(sol));
399     iter += 1;
400   end-do
401   Writing_method := orig_Writing_method;
402   forall(Pos in alternative_Pos | Pos <> orig_Position) do
403     Position := Pos;
404     create_dist_start;
405     create_time_start;
406     create_dist_keys;
407     create_time_keys;
408     writeln(iter,'. alternative Verwendung (' ,Writing_method,'$, $ ',Position,'$, $ ',aver_text_length,' Buchstaben):');
409     writeln(' -Zurueckzulegende Distanz (LE): ',keyboard_quality_dist(sol));
410     writeln(' -Aufzuwendende Zeit (s): ',keyboard_quality_time(sol));
411     writeln('Theoretische Leistungsfahigkeit (wpm): ',keyboard_quality_wpm(sol));
412     iter += 1;
413   end-do
414   Writing_method := orig_Writing_method;
415   Position := orig_Position;
416   create_dist_start;
417   create_time_start;
418   create_dist_keys;
```

```

418 create_time_keys;
419 end-procedure
420
421 !*****
422 !Funktion um die Schreibgeschwindigkeit einer Tastenbelgung zu ermitteln
423 function keyboard_quality_wpm(sol:array(Letters,Keys) of real): real
424   declarations
425     wpm, time_per_char: real;
426     prob_jl: real;
427     best_time_keys: array(Keys,Keys) of real;
428   end-declarations
429   time_per_char := 0;
430   wpm := 0;
431   if(Position = 'vertical') then
432     forall(j in Keys,l in Keys) do
433       best_time_keys(j,l) := movement_time_key2key(j,l);
434     end-do
435   end-if
436   if(Position = 'horizontal') then
437     forall(j in Keys,l in Keys) do
438       best_time_keys(j,l) := movement_time_key2key(j,l);
439     end-do
440     forall(j in Keys,l in Keys | (j in Leftkeys and l in Rightkeys) or (l in Leftkeys and j in Rightkeys)) do
441       best_time_keys(j,l) := maxlist(movement_time_key2point(j,'middle'),movement_time_key2point(l,'middle'));
442     end-do
443   end-if
444   if(Position = 'hybrid') then
445     forall(j in Keys,l in Keys) do
446       best_time_keys(j,l) := movement_time_key2key(j,l);
447     end-do
448     forall(j in Keys,l in Keys | (j in Leftkeys and l in Rightkeys) or (l in Leftkeys and j in Rightkeys)) do
449       best_time_keys(j,l) := alpha*movement_time_key2key(j,l) + beta*maxlist(movement_time_key2point(j,'middle'),
450         movement_time_key2point(l,'middle'));
451     end-do
452   end-if
453   forall(j in Keys,l in Keys) do
454     prob_jl := 0;
455     forall(i in Letters,k in Letters | abs(1-sol(i,j))<tol and abs(1-sol(k,l))<tol) do
456       prob_jl := bigram_prob_orig(i,k);
457     end-do
458     time_per_char += prob_jl*best_time_keys(j,l);
459   end-do
460   time_per_char := time_per_char/(sum(i in Letters,k in Letters) bigram_prob_orig(i,k));
461   wpm := 60/(5*time_per_char);
462   returned := wpm;
463 end-function
464
465 !*****
466 !Funktion um die zurueckzulegende Distanz einer Tastenbelgung zu ermitteln
467 function keyboard_quality_dist(sol:array(Letters,Keys) of real): real
468   declarations
469     quality: real;
470   end-declarations
471   quality := (1/sum(i in Letters) letter_prob_orig(i))*2*(sum(i in Letters,j in Keys) dist_start(j)*letter_prob_orig(i)*
472     sol(i,j))*aver_text_length;
473   if(Writing_method <> 'simple') then
474     quality := quality/aver_text_length + ((sum(i in Letters,j in Keys,k in Letters,l in Keys) sol(i,j)*sol(k,l)*
475       dist_keys(j,l)*bigram_prob_orig(i,k))*(aver_text_length-1)*(1/sum(i in Letters,k in Letters) bigram_prob_orig(i,k)
476     ));
477   end-if
478   returned := quality;
479 end-function
480
481 !*****
482 !Funktion um die Schreibgeschwindigkeit einer Tastenbelgung zu ermitteln
483 function keyboard_quality_time(sol:array(Letters,Keys) of real): real
484   declarations
485     quality: real;
486   end-declarations
487   quality := (1/sum(i in Letters) letter_prob_orig(i))*2*(sum(i in Letters,j in Keys) time_start(j)*letter_prob_orig(i)*
488     sol(i,j))*aver_text_length;
489   if(Writing_method <> 'simple') then

```

B Quellcode

```
485     quality := quality/aver_text_length + ((sum(i in Letters,j in Keys,k in Letters,l in Keys) sol(i,j)*sol(k,l)*
         time_keys(j,l)*bigram_prob_orig(i,k))*(aver_text_length-1)*(1/sum(i in Letters,k in Letters) bigram_prob_orig(i,k)
         ));
486   end-if
487   returned := quality;
488 end-function
489
490 !*****
491 !Prozedur um die Zielfunktion zu erstellen
492 procedure create_objfct
493   if(Objective = 'travel-distance') then
494     if(Objfct_extra_term = 'on' or Writing_method='simple') then
495       objfct := aver_text_length*(1/sum(i in Letters) letter_prob_orig(i))*2*(sum(i in Letters,j in Keys) dist_start(j)*
         letter_prob(i)*y(i,j));
496       if(Writing_method<>'simple') then
497         objfct := objfct/aver_text_length + ((sum(i in Letters,j in Keys,k in Letters,l in Keys) x(i,j,k,l)*dist_keys(j,
         l)*bigram_prob(i,k))*(aver_text_length-1)*(1/sum(i in Letters,k in Letters) bigram_prob(i,k)));
498       end-if
499     else
500       objfct := (aver_text_length-1)*((sum(i in Letters,j in Keys,k in Letters,l in Keys) x(i,j,k,l)*dist_keys(j,l)*
         bigram_prob(i,k))*(1/sum(i in Letters,k in Letters) bigram_prob(i,k)));
501     end-if
502   elif(Objective = 'travel-time') then
503     if(Objfct_extra_term = 'on' or Writing_method='simple') then
504       objfct := aver_text_length*(1/sum(i in Letters) letter_prob_orig(i))*2*(sum(i in Letters,j in Keys) time_start(j)*
         letter_prob(i)*y(i,j));
505       if(Writing_method<>'simple') then
506         objfct := objfct/aver_text_length + ((sum(i in Letters,j in Keys,k in Letters,l in Keys) x(i,j,k,l)*time_keys(j,
         l)*bigram_prob(i,k))*(aver_text_length-1)*(1/sum(i in Letters,k in Letters) bigram_prob(i,k)));
507       end-if
508     else
509       objfct := (aver_text_length-1)*((sum(i in Letters,j in Keys,k in Letters,l in Keys) x(i,j,k,l)*time_keys(j,l)*
         bigram_prob(i,k))*(1/sum(i in Letters,k in Letters) bigram_prob(i,k)));
510     end-if
511   end-if
512 end-procedure
513
514 !*****
515 !Prozedur um die Matrix der benoetigten Zeiten zwischen den Tasten zu erstellen
516 procedure create_time_keys
517   if(Position = 'vertical') then
518     forall(j in Keys,l in Keys) do
519       time_keys(j,l) := movement_time_key2key(j,l);
520     end-do
521   elif(Position = 'horizontal') then
522     forall(j in Keys,l in Keys) do
523       time_keys(j,l) := movement_time_key2key(j,l);
524     end-do
525   if(Writing_method = 'semi-realistic') then
526     forall(j in Keys,l in Keys | (j in Leftkeys and l in Rightkeys) or (l in Leftkeys and j in Rightkeys)) do
527       time_keys(j,l) := movement_time_key2point(j,'start') + movement_time_key2point(l,'start');
528     end-do
529   elif(Writing_method = 'realistic') then
530     forall(j in Keys,l in Keys | (j in Leftkeys and l in Rightkeys) or (l in Leftkeys and j in Rightkeys)) do
531       time_keys(j,l) := movement_time_key2point(j,'middle') + movement_time_key2point(l,'middle');
532     end-do
533   elif(Writing_method = 'hover-over') then
534     forall(j in Keys,l in Keys | (j in Leftkeys and l in Rightkeys) or (l in Leftkeys and j in Rightkeys)) do
535       time_keys(j,l) := maxlist(movement_time_key2point(j,'middle'),movement_time_key2point(l,'middle'));
536     end-do
537   end-if
538   elif(Position = 'hybrid') then
539     forall(j in Keys,l in Keys) do
540       time_keys(j,l) := movement_time_key2key(j,l);
541     end-do
542   if(Writing_method = 'realistic') then
543     forall(j in Keys,l in Keys | (j in Leftkeys and l in Rightkeys) or (l in Leftkeys and j in Rightkeys)) do
544       time_keys(j,l) := alpha * movement_time_key2key(j,l) + beta * (movement_time_key2point(j,'middle') +
         movement_time_key2point(l,'middle'));
545     end-do
546   elif(Writing_method = 'hover-over') then
547     forall(j in Keys,l in Keys | (j in Leftkeys and l in Rightkeys) or (l in Leftkeys and j in Rightkeys)) do
```

```

548     time_keys(j,l) := alpha * movement_time_key2key(j,l) + beta * (maxlist(movement_time_key2point(j,'middle'),
movement_time_key2point(l,'middle')));
549     end-do
550     end-if
551     end-if
552 end-procedure
553
554 !*****
555 !Funktion um die Dauer einer Bewegung zwischen einer Taste und einem Punkt zu bestimmen
556 function movement_time_key2point(j:integer,point: string): real
557     declarations
558         mov_time: real;
559     end-declarations
560     mov_time := 0;
561     if(point = 'start') then
562         if(dist_start(j)>0) then
563             if(Position = 'hybrid') then
564                 mov_time := alpha*(fitts_law_param_a + fitts_law_param_b*(log(1 + dist_start_vert_orig(j)/key_width)/log(2))) +
beta*(fitts_law_param_a + fitts_law_param_b*(log(1 + dist_start_hor_orig(j)/key_width)/log(2)));
565             else
566                 mov_time := fitts_law_param_a + fitts_law_param_b*(log(1 + dist_start(j)/key_width)/log(2));
567             end-if
568         else
569             mov_time := fitts_law_param_a;
570         end-if
571     elif(point = 'middle') then
572         if(dist_mid_hor_orig(j)>0) then
573             mov_time := fitts_law_param_a + fitts_law_param_b*(log(1 + dist_mid_hor_orig(j)/key_width)/log(2));
574         elif(dist_mid_hor_orig(j)=0) then
575             mov_time := fitts_law_param_a;
576         end-if
577     end-if
578     returned := mov_time;
579 end-function
580
581 !*****
582 !Funktion um die Dauer einer Bewegung zwischen zwei Tasten zu bestimmen
583 function movement_time_key2key(j:integer,l:integer): real
584     declarations
585         mov_time: real;
586         dist_jl: real;
587     end-declarations
588     mov_time := 0;
589     if(j<>l) then
590         dist_jl := dist_keys_orig(j,l);
591         mov_time := fitts_law_param_a + fitts_law_param_b*(log(1 + dist_jl/key_width)/log(2));
592     else
593         mov_time := fitts_law_param_a;
594     end-if
595     returned := mov_time;
596 end-function
597
598 !*****
599 !Prozedur um die Matrix der Distanzen zwischen den Tasten zu erstellen
600 procedure create_dist_keys
601     if(Position = 'vertical') then
602         forall(j in Keys,l in Keys) do
603             dist_keys(j,l) := dist_keys_orig(j,l);
604         end-do
605     elif(Position = 'horizontal') then
606         forall(j in Keys,l in Keys) do
607             dist_keys(j,l) := dist_keys_orig(j,l);
608         end-do
609     if(Writing_method = 'semi-realistic') then
610         forall(j in Keys,l in Keys | (j in Leftkeys and l in Rightkeys) or (l in Leftkeys and j in Rightkeys)) do
611             dist_keys(j,l) := dist_start_hor_orig(j) + dist_start_hor_orig(l);
612         end-do
613     elif(Writing_method = 'realistic') then
614         forall(j in Keys,l in Keys | (j in Leftkeys and l in Rightkeys) or (l in Leftkeys and j in Rightkeys)) do
615             dist_keys(j,l) := dist_mid_hor_orig(j) + dist_mid_hor_orig(l);
616         end-do
617     elif(Writing_method = 'hover-over') then
618         forall(j in Keys,l in Keys | (j in Leftkeys and l in Rightkeys) or (l in Leftkeys and j in Rightkeys)) do

```

B Quellcode

```
619         dist_keys(j,l) := dist_mid_hor_orig(j) + dist_mid_hor_orig(l);
620     end-do
621 end-if
622 elif(Position = 'hybrid') then
623     forall(j in Keys,l in Keys) do
624         dist_keys(j,l) := dist_keys_orig(j,l);
625     end-do
626     forall(j in Keys,l in Keys | (j in Leftkeys and l in Rightkeys) or (l in Leftkeys and j in Rightkeys)) do
627         dist_keys(j,l) := alpha * dist_keys_orig(j,l) + beta * (dist_mid_hor_orig(j) + dist_mid_hor_orig(l));
628     end-do
629 end-if
630 end-procedure
631
632 !*****
633 !Prozedur um den Vektor der Distanzen zum Startpunkt zu erstellen
634 procedure create_dist_start
635     if(Position = 'vertical') then
636         forall(j in 1..numberofkeys) dist_start(j) := dist_start_vert_orig(j);
637     elif(Position = 'horizontal') then
638         forall(j in 1..numberofkeys) dist_start(j) := dist_start_hor_orig(j);
639     elif(Position = 'hybrid') then
640         forall(j in 1..numberofkeys) dist_start(j) := alpha*dist_start_vert_orig(j)+beta*dist_start_hor_orig(j);
641     end-if
642 end-procedure
643
644 !*****
645 !Prozedur um den Vektor der Buchstabenwahrscheinlichkeiten zu erstellen
646 procedure create_letter_prob
647     forall(i in Letters) letter_prob(i) := letter_prob_orig(i);
648 end-procedure
649
650 !*****
651 !Prozedur um den Vektor der Bewegungszeiten zum Startpunkt zu erstellen
652 procedure create_time_start
653     forall(j in Keys) time_start(j) := movement_time_key2point(j,'start');
654 end-procedure
655
656 !*****
657 !Prozedur um Tasten auf die Seiten zu verteilen
658 procedure assign_keys_to_side
659     declarations
660         columns_per_side_rect: integer;
661         columns_per_side_free: array(1..Rows) of integer;
662     end-declarations
663     Leftkeys := {};
664     if(Shape = 'rectangular') then
665         columns_per_side_rect := floor(Columns_rect/2);
666         forall(r in 1..Rows,c in 1..columns_per_side_rect) do
667             Leftkeys += {(r-1)*Columns_rect+c};
668         end-do
669     elif(Shape = 'free') then
670         forall(r in 1..Rows) columns_per_side_free(r) := floor(Columns_free(r)/2);
671         forall(r in 1..Rows,c in 1..columns_per_side_free(r)) do
672             Leftkeys += {c+sum(z in 1..r-1) Columns_free(z)};
673         end-do
674     end-if
675     Rightkeys := Keys - Leftkeys;
676 end-procedure
677
678 !*****
679 !Prozedur um die eingelesenen Daten auf Fehler zu ueberpruefen
680 procedure check_data
681     !input ueberpruefen: Tasten und Tastaturlayout
682     if((Shape = 'free' and numberofkeys <> sum(i in 1..Rows) Columns_free(i)) or (Shape = 'rectangular' and numberofkeys <
683         > Rows*Columns_rect)) then
684         error := true;
685         writeln('input-Fehler: Tastenzahlen stimmen nicht mit dem angegebenen Tastaturlayout ueberein!');
686     elif (numberofkeys < numberofletters) then
687         error := true;
688         writeln('input-Fehler: Zu wenige Tasten um alle Buchstaben zu verteilen!');
689     elif (Shape <> 'rectangular' and Shape <> 'free') then
690         error := true;
691         writeln('input-Fehler: ueberpruefe Tastaturlayout!');
```

```

691 end-if
692 !input ueberpruefen: Wahrscheinlichkeiten
693 if(or(i in Letters,k in Letters) (bigram_prob_orig(i,k)<0 or bigram_prob_orig(i,k)>100)) then
694     writeln('input-Fehler: Bigramm-Wahrscheinlichkeiten ueberpruefen!');
695     error := true;
696 end-if
697 if(or(i in Letters) (letter_prob_orig(i)<0 or letter_prob_orig(i)>100)) then
698     writeln('input-Fehler: Buchstaben-Wahrscheinlichkeiten ueberpruefen!');
699     error := true;
700 end-if
701 !input ueberpruefen: Distanzen
702 if(or(j in Keys,l in Keys) dist_keys_orig(j,l)<0 ) then
703     writeln('input-Fehler: Distanzen zwischen den Tasten ueberpruefen!');
704     error := true;
705 end-if
706 if(or(i in 1..numberofkeys) (dist_start_vert_orig(i)<0 or dist_start_hor_orig(i)<0)) then
707     writeln('input-Fehler: Distanzen zum Startpunkt ueberpruefen!');
708     error := true;
709 end-if
710 if(or(i in 1..numberofkeys) dist_mid_hor_orig(i)<0) then
711     writeln('input-Fehler: Distanzen zur Mitte ueberpruefen!');
712     error := true;
713 end-if
714 end-procedure
715
716 !*****
717 !Prozedur um die Einstellungen auf Fehler zu ueberpruefen
718 procedure check_setting
719     if(Task not in {'optimization','evaluation'}) then
720         writeln('Fehler $in$ den Einstellungen: ueberpruefe die Aufgabe des Tools');
721         error := true;
722     end-if
723     if(Position not in {'vertical','horizontal','hybrid'}) then
724         writeln('Fehler $in$ den Einstellungen: Position des Geraets');
725         error := true;
726     end-if
727     if(Position = 'vertical' and Writing_method not in {'simple','realistic'}) then
728         writeln('Fehler $in$ den Einstellungen: Diese Schreibweise existiert nicht fuer vertikales Geraet');
729         error := true;
730     end-if
731     if(Position = 'hybrid' and Writing_method not in {'realistic','hover-over'}) then
732         writeln('Fehler $in$ den Einstellungen: Diese Schreibweise ist fuer hybrid-positioniertes Geraet nicht verfuegbar');
733         error := true;
734     end-if
735     if(Writing_method not in {'simple','semi-realistic','realistic','hover-over'}) then
736         writeln('Fehler $in$ den Einstellungen: Schreibweise ueberpruefen');
737         error := true;
738     end-if
739     if(aver_text_length<1) then
740         writeln('Fehler $in$ den Einstellungen: Durchschnittliche Textlaenge ueberpruefen!');
741         error := true;
742     end-if
743     if(fitts_low_param_a<0 or fitts_low_param_b<0 or key_width<=0 or alpha<0 or beta<0 or (alpha+beta<>1)) then
744         writeln('Fehler $in$ den Einstellungen: Fitts'-Law- und Hybrid-Parameter pruefen!');
745         error := true;
746     end-if
747     if(Heuristic_Iter<0) then
748         writeln('Fehler $in$ den Einstellungen: Maximale Anzahl an Heuristik-Iterationen ueberpruefen!');
749         error := true;
750     end-if
751     if(Task = 'evaluation') then
752         if((or(i in Letters) ((assignment_letters_keys(i)>numberofkeys) or (assignment_letters_keys(i)<=0))) or (or(i in
753             Letters, j in Letters | i<> j) (assignment_letters_keys(i)=assignment_letters_keys(j)))) then
754             writeln('Fehler $in$ den Einstellungen: Eingelesene Tastenbelegung ueberpruefen!');
755             error := true;
756         end-if
757         if(improve_own_keyboard = 'on' and Objective not in {'travel-time','travel-distance'}) then
758             writeln('Fehler $in$ den Einstellungen: Zielfunktion');
759             error := true;
760         end-if
761     elif(Task = 'optimization') then
762         if(Objective not in {'travel-time','travel-distance'}) then
763             writeln('Fehler $in$ den Einstellungen: Zielfunktion');

```

B Quellcode

```
763     error := true;
764 end-if
765 if(Writing_method = 'simple' and (PROB_HANDLING <> 'none' or (or(z in {Objfct_extra_term,Implicit_zero,X_LowB,X_UpB,
AJ_Con,X_Equ,XY_Equ,Letter_Letter_Con,Key_Key_Con,Letter_Key_Con,Binaryness}) z <> 'off'))) then
766     writeln('Fehler $in$ den Einstellungen: Einfache Schreibweise nicht mit gewaehlten Optionen kombinierbar');
767     error := true;
768 end-if
769 if(MESSLEV<=0 or MESSLEV>6) then
770     writeln('Fehler $in$ den Einstellungen: Message-Level ueberpruefen!');
771     error := true;
772 end-if
773 if(tol<=0 or tol>=1) then
774     writeln('Fehler $in$ den Einstellungen: Toleranz ueberpruefen!');
775     error := true;
776 end-if
777 if(Writing_method <> 'simple') then
778     if(PROB_HANDLING not in {'none','threshold','threshold_but_min_entries_per_row','global_best','
global_best_but_max_entries_per_row','entries_per_row','entries_per_row_plus_global_best'}) then
779         writeln('Fehler $in$ den Einstellungen: PROB_HANDLING ueberpruefen!');
780         error := true;
781     end-if
782     if(PROB_HANDLING='threshold' and ((threshold<0 or threshold>100) or (threshold>max(i in Letters,k in Letters)
bigram_prob_orig(i,k)))) then
783         writeln('Fehler $in$ den Einstellungen: WS-Schranke ueberpruefen!');
784         error := true;
785     elif(PROB_HANDLING='threshold_but_min_entries_per_row' and (threshold<0 or threshold>100 or min_entries_per_row<0
or min_entries_per_row>numberofletters)) then
786         writeln('Fehler $in$ den Einstellungen: threshold oder min_entries_per_row!');
787         error := true;
788     elif(PROB_HANDLING='global_best' and (global_best<0 or global_best>numberofletters*numberofletters)) then
789         writeln('Fehler $in$ den Einstellungen: global_best!');
790         error := true;
791     elif(PROB_HANDLING='global_best_but_max_entries_per_row' and (global_best<0 or global_best>numberofletters*
max_entries_per_row or max_entries_per_row<=0)) then
792         writeln('Fehler $in$ den Einstellungen: global_best oder max_entries_per_row!');
793         error := true;
794     elif(PROB_HANDLING='entries_per_row' and (entries_per_row<=0 or entries_per_row>numberofletters)) then
795         writeln('Fehler $in$ den Einstellungen: entries_per_row!');
796         error := true;
797     elif(PROB_HANDLING='entries_per_row_plus_global_best' and (entries_per_row<=0 or entries_per_row>numberofletters
or global_best<0 or entries_per_row*numberofletters+global_best>numberofletters*numberofletters)) then
798         writeln('Fehler $in$ den Einstellungen: entries_per_row oder global_best!');
799         error := true;
800     end-if
801     if(or(z in {Implicit_zero,X_LowB,X_UpB,AJ_Con,X_Equ,XY_Equ,Letter_Letter_Con,Key_Key_Con,Letter_Key_Con,Binaryness
}) (z<>'on' and z<>'off'))) then
802         writeln('Fehler $in$ den Einstellungen: Verwende on/off');
803         error := true;
804     end-if
805     if((Implicit_zero='off' and X_LowB='off' and (AJ_Con='off' or X_Equ='off')) or (Implicit_zero = 'on' and X_LowB='
off')) then
806         writeln('Fehler $in$ den Einstellungen: Die x-Variablen benoetigen eine untere Schranke');
807         error := true;
808     end-if
809     if(X_LowB='on' and X_UpB='on' and (and(z in {AJ_Con,Letter_Letter_Con,Key_Key_Con,Letter_Key_Con,Binaryness})z='
off')) then
810         writeln('Fehler $in$ den Einstellungen: Bitte Ganzzahligkeit fordern');
811     end-if
812 end-if
813 if(Fix_Letter not in 0..numberofletters) then
814     writeln('Fehler $in$ den Einstellungen: Zu fixierender Buchstabe');
815     error := true;
816 end-if
817 if(PROB_HANDLING='none' and Heuristic_Iter>0) then
818     writeln('Fehler $in$ den Einstellungen: Heuristik bringt keinen Gewinn bei vollstaendiger Optimierung!');
819     error := true;
820 end-if
821 if(examine_own_keyboard = 'on' and ((or(i in Letters)((assignment_letters_keys(i)>numberofkeys) or (
assignment_letters_keys(i)<=0))) or (or(i in Letters, j in Letters | i<> j)(assignment_letters_keys(i)=
assignment_letters_keys(j))))) then
822     writeln('Fehler $in$ den Einstellungen: Eingelesene Tastenbelegung ueberpruefen!');
823     error := true;
824 end-if
```

```

825 end-if
826 end-procedure
827
828 !*****
829 !Prozedur um Zeit- oder Distanz-Matrizen anzeigen zu koennen
830 procedure show_dist_or_time_matrix(Matrix: array(Keys,Keys) of real)
831   forall(j in 1..numberofkeys) do
832     forall(l in 1..numberofkeys) do
833       write(Matrix(j,l),' ');
834     end-do
835     writeln(' ');
836   end-do
837   writeln(' ');
838 end-procedure
839
840 !*****
841 !Prozedur um die Bigrammwahrscheinlichkeits-Matrix anzeigen zu koennen
842 procedure show_probability_matrix(Matrix: array(Letters,Letters) of real)
843   forall(i in Letters) do
844     forall(k in Letters) do
845       write(Matrix(i,k),' ');
846     end-do
847     writeln(' ');
848   end-do
849   writeln(' ');
850 end-procedure
851
852 !*****
853 !Prozedur um die Bigrammwahrscheinlichkeits-Matrix zu analysieren
854 procedure analyse_probability_matrix(Matrix: array(Letters,Letters) of real)
855   declarations
856     analysiscounter: integer;
857     min_val,max_val,entry_sum,aver: real;
858   end-declarations
859   analysiscounter := 0;
860   writeln(' <= | 0 1 2 3 4 5 6 7 8 9 ');
861   write(' -----|-----');
862   forall(i in 0..3) do
863     writeln(' ');
864     write(' ',i,'$',$.... | ');
865     forall(j in 0..9) do
866       analysiscounter := 0;
867       forall(k in Letters,l in Letters | i+0.1*(j-1) < Matrix(k,l) and Matrix(k,l) <= i+0.1*j) do
868         analysiscounter += 1;
869       end-do
870       write(analysiscounter,' ');
871     end-do
872   end-do
873   writeln(' ');
874   writeln(' ');
875   min_val := min(i in Letters, k in Letters) Matrix(i,k);
876   max_val := max(i in Letters, k in Letters) Matrix(i,k);
877   entry_sum := sum(i in Letters, k in Letters) Matrix(i,k);
878   writeln('Minimaler Eintrag: ',min_val);
879   writeln('Maximaler Eintrag: ',max_val);
880   writeln('Durchschnittlicher Eintrag: ',entry_sum/(numberofletters^2));
881   writeln('Summe der Eintraege: ',sum(i in Letters,k in Letters) Matrix(i,k));
882   writeln(' ');
883   min_val := min(i in Letters, k in Letters | Matrix(i,k)<>0) Matrix(i,k);
884   max_val := max(i in Letters, k in Letters | Matrix(i,k)<>0) Matrix(i,k);
885   aver := entry_sum/(sum(i in Letters, k in Letters | Matrix(i,k)<>0) 1);
886   writeln('Minimaler Nicht-Null-Eintrag: ',min_val);
887   writeln('Maximaler Nicht-Null-Eintrag: ',max_val);
888   writeln('Durchschnitt der Nicht-Null-Eintraege: ',aver);
889   writeln(' ');
890   writeln(' ');
891 end-procedure
892
893 !*****
894 !Prozedur um Nebenbedingungen zu erzeugen
895 procedure create_constraints
896   !AUSGABE
897   if (MESSLEV >= 2) then

```

B Quellcode

```
898     writeln('Erzeuge Nebenbedingungen...');
899 end-if
900 !NB(y): Ganzzahligkeit
901 forall(i in Letters,j in Keys) do
902     y(i,j) is_binary;
903     numberofvar_y += 1;
904 end-do
905 !NB(y): Jeder Buchstaben muss genau einmal vorkommen
906 forall(i in Letters) do
907     sum(j in Keys) y(i,j) = 1;
908     numberofconstraints_equ += 1;
909 end-do
910 !NB(y): Hoechstens oder genau ein Buchstabe pro Taste
911 if(numberofletters = numberofkeys) then
912     forall(j in Keys) do
913         sum(i in Letters) y(i,j) = 1;
914         numberofconstraints_equ += 1;
915     end-do
916 else
917     forall(j in Keys) do
918         sum(i in Letters) y(i,j) <= 1;
919         numberofconstraints_inequ += 1;
920     end-do
921 end-if
922 if(Writing_method <> 'simple') then
923     !NB(x): Ganzzahligkeit
924     if(Binaryness = 'on') then
925         forall(i in Letters,j in Keys,k in Letters,l in Keys) do
926             x(i,j,k,l) is_binary;
927             numberofvar_x += 1;
928         end-do
929     end-if
930     if(Implicit_zero = 'off') then
931         !NB(y,$x): Untere Schranke fuer x-Variablen
932         if(X_LowB = 'on') then
933             forall(i in Letters,j in Keys,k in Letters,l in Keys) do
934                 y(i,j) + y(k,l) - 1 <= x(i,j,k,l);
935                 numberofconstraints_inequ += 1;
936             end-do
937         end-if
938         !NB(y,$x): Obere Schranke fuer x-Variablen
939         if(X_UpB = 'on') then
940             forall(i in Letters,j in Keys,k in Letters,l in Keys) do
941                 2*x(i,j,k,l) <= y(i,j) + y(k,l);
942                 numberofconstraints_inequ += 1;
943             end-do
944         end-if
945         !NB(y,$x): Adams-Johnson-Nebenbedingungen
946         if(AJ_Con = 'on') then
947             if(numberofletters=numberofkeys) then
948                 forall(j in Keys,k in Letters,l in Keys) do
949                     sum(i in Letters) x(i,j,k,l) = y(k,l);
950                     numberofconstraints_equ += 1;
951                 end-do
952                 forall(i in Letters,k in Letters,l in Keys) do
953                     sum(j in Keys) x(i,j,k,l) = y(k,l);
954                     numberofconstraints_equ += 1;
955                 end-do
956             else
957                 forall(j in Keys,k in Letters,l in Keys) do
958                     sum(i in Letters) x(i,j,k,l) <= y(k,l);
959                     numberofconstraints_inequ += 1;
960                 end-do
961                 forall(i in Letters,k in Letters,l in Keys) do
962                     sum(j in Keys) x(i,j,k,l) = y(k,l);
963                     numberofconstraints_equ += 1;
964                 end-do
965             end-if
966         end-if
967     !NB(x): Paarweise Gleichheit
968     if(X_Equ = 'on') then
969         forall(i in Letters,j in Keys,k in Letters,l in Keys | i<=k and (i<>k or j<>l)) do
970             x(i,j,k,l) = x(k,l,i,j);
```

```

971     numberofconstraints_equ +=1;
972 end-do
973 end-if
974 !NB(y$, $x): Gleichheit von bestimmten x und y
975 if (XY_Equ = 'on') then
976     forall(i in Letters, j in Keys) do
977         x(i, j, i, j) = y(i, j);
978         numberofconstraints_equ +=1;
979     end-do
980 end-if
981 !NB(x): Genau ein Tastenpaar pro Buchstabenpaar
982 if (Letter_Letter_Con = 'on') then
983     forall(i in Letters, k in Letters) do
984         sum(j in Keys, l in Keys) x(i, j, k, l) = 1;
985         numberofconstraints_equ +=1;
986     end-do
987 end-if
988 !NB(x): Hoechstens oder genau ein Buchstabenpaar pro Tastenpaar
989 if (Key_Key_Con = 'on') then
990     if (numberofkeys=numberofletters) then
991         forall(j in Keys, l in Keys) do
992             sum(i in Letters, k in Letters) x(i, j, k, l) = 1;
993             numberofconstraints_equ +=1;
994         end-do
995     else
996         forall(j in Keys, l in Keys) do
997             sum(i in Letters, k in Letters) x(i, j, k, l) <= 1;
998             numberofconstraints_inequ +=1;
999         end-do
1000     end-if
1001 end-if
1002 !NB(x): Buchstabe-Taste-Paar
1003 if (Letter_Key_Con = 'on') then
1004     forall(k in Letters, l in Keys) do
1005         sum(i in Letters, j in Keys) x(i, j, k, l) = numberofletters*y(k, l);
1006         numberofconstraints_equ += 1;
1007     end-do
1008 end-if
1009 elif (Implicit_zero = 'on') then
1010     if (Objective = 'travel-distance') then
1011         !NB(x): Nullsetzen ausgewaehlter x-Variablen
1012         forall(i in Letters, j in Keys, k in Letters, l in Keys | i=k or j=l or bigram_prob(i, k)=0) do !and not(i$=$k and
1013             j$=$l)???
1014             x(i, j, k, l) = 0;
1015             numberofconstraints_equ +=1;
1016             numberofvar_implzero += 1;
1017         end-do
1018     else
1019         !NB(y$, $x): Untere Schranke fuer x-Variablen$, $ die nicht gleich Null gesetzt wurden
1020         if (X_LowB = 'on') then
1021             forall(i in Letters, j in Keys, k in Letters, l in Keys | not(i=k or j=l or bigram_prob(i, k)=0)) do !NB:
1022                 Coupling constraints
1023                 y(i, j) + y(k, l) -1 <= x(i, j, k, l);
1024                 numberofconstraints_inequ +=1;
1025             end-do
1026         end-if
1027     else
1028         !NB(y$, $x): Obere Schranke fuer x-Variablen$, $ die nicht gleich Null gesetzt wurden
1029         if (X_UpB = 'on') then
1030             forall(i in Letters, j in Keys, k in Letters, l in Keys | not(i=k or j=l or bigram_prob(i, k)=0)) do
1031                 2*x(i, j, k, l) <= y(i, j) + y(k, l);
1032                 numberofconstraints_inequ +=1;
1033             end-do
1034         end-if
1035     else
1036         !NB(y$, $x): Adams-Johnson-Nebenbedingungen
1037         if (AJ_Con = 'on') then
1038             forall(j in Keys, k in Letters, l in Keys) do
1039                 sum(i in Letters) x(i, j, k, l) <= y(k, l);
1040                 numberofconstraints_inequ += 1;
1041             end-do
1042         end-if
1043     else
1044         forall(i in Letters, k in Letters, l in Keys | i=k or bigram_prob(i, k)=0) do
1045             sum(j in Keys) x(i, j, k, l) <= y(k, l);
1046             numberofconstraints_inequ += 1;
1047         end-do
1048     else
1049         forall(i in Letters, k in Letters, l in Keys | i<>k and bigram_prob(i, k)>0) do

```

B Quellcode

```
1042     sum(j in Keys) x(i,j,k,l) = y(k,l);
1043     numberofconstraints_equ += 1;
1044 end-do
1045 end-if
1046 !NB(x): Paarweise Gleichheit
1047 if (X_Equ = 'on') then
1048     forall(i in Letters,j in Keys,k in Letters,l in Keys | i<k and j<>l and bigram_prob(i,k)>0 and bigram_prob(k,i
1049 )>0) do
1049         x(i,j,k,l) = x(k,l,i,j);
1050         numberofconstraints_equ +=1;
1051     end-do
1052 end-if
1053 !NB(yx): Gleichheit bestimmter y und x
1054 if (XY_Equ = 'on') then
1055     forall(i in Letters,j in Keys) do
1056         x(i,j,i,j) = y(i,j);
1057         numberofconstraints_equ +=1;
1058     end-do
1059 end-if
1060 !NB(x): Hoechstens oder genau ein Tastenpaar pro Buchstabenpaar
1061 if (Letter_Letter_Con = 'on') then
1062     forall(i in Letters,k in Letters | i=k or bigram_prob(i,k)=0) do
1063         sum(j in Keys,l in Keys) x(i,j,k,l) <= 1;
1064         numberofconstraints_equ +=1;
1065     end-do
1066     forall(i in Letters,k in Letters | i<>k and bigram_prob(i,k)>0) do
1067         sum(j in Keys,l in Keys) x(i,j,k,l) = 1;
1068         numberofconstraints_equ +=1;
1069     end-do
1070 end-if
1071 !NB(x): Hoechstens ein Buchstabenpaar pro Tastenpaar
1072 if (Key_Key_Con = 'on') then
1073     forall(j in Keys,l in Keys) do
1074         sum(i in Letters,k in Letters) x(i,j,k,l) <= 1;
1075         numberofconstraints_inequ +=1;
1076     end-do
1077 end-if
1078 !NB(y$, $x): Buchstabe-Taste-Paar
1079 if (Letter_Key_Con = 'on') then
1080     forall(k in Letters,l in Keys) do
1081         sum(i in Letters,j in Keys) x(i,j,k,l) <= numberofletters*y(k,l);
1082         numberofconstraints_inequ += 1;
1083     end-do
1084 end-if
1085
1086 elif(Objective = 'travel-time') then
1087     !NB(x): Nullsetzen ausgewaehlter x-Variablen
1088     forall(i in Letters,j in Keys,k in Letters,l in Keys | (i=k and j<>l) or (j=l and i<>k) or bigram_prob(i,k)=0)
1089 do
1089         x(i,j,k,l) = 0;
1090         numberofconstraints_equ +=1;
1091         numberofvar_implzero += 1;
1092     end-do
1093     !NB(y$, $x): Untere Schranke fuer x-Variablen$, $ die nicht gleich Null gesetzt wurden
1094     if (X_LowB = 'on') then
1095         forall(i in Letters,j in Keys,k in Letters,l in Keys | not((i=k and j<>l) or (j=l and i<>k) or bigram_prob(i,k
1096 )=0)) do !NB: Coupling constraints
1096             y(i,j) + y(k,l) -1 <= x(i,j,k,l);
1097             numberofconstraints_inequ +=1;
1098         end-do
1099     end-if
1100     !Obere Schranke fuer x-Variablen$, $ die nicht gleich Null gesetzt wurden
1101     if (X_UpB = 'on') then
1102         forall(i in Letters,j in Keys,k in Letters,l in Keys | not((i=k and j<>l) or (j=l and i<>k) or bigram_prob(i,k
1103 )=0)) do
1103             2*x(i,j,k,l) <= y(i,j) + y(k,l);
1104             numberofconstraints_inequ +=1;
1105         end-do
1106     end-if
1107     !Adams-Johnson-Nebenbedingungen
1108     if (AJ_Con = 'on') then
1109         forall(j in Keys,k in Letters,l in Keys) do
1109             sum(i in Letters) x(i,j,k,l) <= y(k,l);
```

```

1111         numberofconstraints_inequ += 1;
1112     end-do
1113     forall(i in Letters,k in Letters,l in Keys | bigram_prob(i,k)=0) do
1114         sum(j in Keys) x(i,j,k,l) <= y(k,l);
1115         numberofconstraints_inequ += 1;
1116     end-do
1117     forall(i in Letters,k in Letters,l in Keys | bigram_prob(i,k)>0) do
1118         sum(j in Keys) x(i,j,k,l) = y(k,l);
1119         numberofconstraints_equ += 1;
1120     end-do
1121     end-if
1122     !NB(x): Paarweise Gleichheit
1123     if (X_Equ = 'on') then
1124         forall(i in Letters,j in Keys,k in Letters,l in Keys | i<k and j<l and bigram_prob(i,k)>0 and bigram_prob(k,i
1125 )>0) do
1126             x(i,j,k,l) = x(k,l,i,j);
1127             numberofconstraints_equ +=1;
1128         end-do
1129     end-if
1130     !NB(y$, $x): Gleichheit bestimmter y und x
1131     if (XY_Equ = 'on') then
1132         forall(i in Letters,j in Keys) do
1133             x(i,j,i,j) = y(i,j);
1134             numberofconstraints_equ +=1;
1135         end-do
1136     end-if
1137     !NB(x): Hoechstens oder genau ein Tastenpaar pro Buchstabenpaar
1138     if(Letter_Letter_Con = 'on') then
1139         forall(i in Letters,k in Letters | bigram_prob(i,k)=0) do
1140             sum(j in Keys,l in Keys) x(i,j,k,l) <= 1;
1141             numberofconstraints_equ +=1;
1142         end-do
1143         forall(i in Letters,k in Letters | bigram_prob(i,k)>0) do
1144             sum(j in Keys,l in Keys) x(i,j,k,l) = 1;
1145             numberofconstraints_equ +=1;
1146         end-do
1147     end-if
1148     !NB(x): Hoechstens ein Buchstabenpaar pro Tastenpaar
1149     if(Key_Key_Con = 'on') then
1150         forall(j in Keys,l in Keys) do
1151             sum(i in Letters,k in Letters) x(i,j,k,l) <= 1;
1152             numberofconstraints_inequ +=1;
1153         end-do
1154     end-if
1155     !NB(y$, $x): Buchstabe-Taste-Paar
1156     if (Letter_Key_Con = 'on') then
1157         forall(k in Letters,l in Keys) do
1158             sum(i in Letters,j in Keys) x(i,j,k,l) <= numberofletters*y(k,l);
1159             numberofconstraints_inequ += 1;
1160         end-do
1161     end-if
1162 end-if
1163 end-if
1164 !NB(y): Buchstaben fixieren
1165 if (Fix_Letter > 0) then
1166     sum(j in Rightkeys) y(Fix_Letter,j) = 1;
1167     numberofconstraints_equ += 1;
1168 end-if
1169 !AUSGABE
1170 if (MESSLEV >= 2) then
1171     writeln('Nebenbedingungen erzeugt...');
1172     show_model;
1173     writeln('Optimierung gestartet...');
1174 end-if
1175 end-procedure
1176
1177 !*****
1178 !Prozedur um die Wahrscheinlichkeitsmatrix zu bearbeiten (Reduktionsmethoden)
1179 procedure create_probability_matrix
1180     declarations
1181         bigram_prob_copy: array(Letters,Letters) of real; !ehemals help_prob
1182         number_of_hidden_entries: integer;

```

B Quellcode

```
1183 help_count: array(Letters) of integer;
1184 missing_entries: integer;
1185 current_prob: array(Letters) of real;
1186 current_max: real;
1187 end-declarations
1188 number_of_hidden_entries := 0;
1189 missing_entries := 0;
1190 if(PROB_HANDLING = 'threshold') then
1191   forall(i in Letters, k in Letters) bigram_prob(i,k) := bigram_prob_orig(i,k);
1192   !Nullsetzen der Wahrscheinlichkeiten unter der Schranke
1193   forall(i in Letters, k in Letters | bigram_prob_orig(i,k) < threshold and bigram_prob_orig(i,k) > 0) do
1194     bigram_prob(i,k) := 0;
1195     number_of_hidden_entries +=1;
1196   end-do
1197 elif(PROB_HANDLING = 'threshold_but_min_entries_per_row') then
1198   forall(i in Letters) help_count(i) := 0;
1199   !Hilfsverfahren fuer Best-N
1200   forall(i in Letters, k in Letters | bigram_prob_orig(i,k) >= threshold) do
1201     bigram_prob(i,k) := bigram_prob_orig(i,k);
1202     help_count(i) += 1;
1203   end-do
1204   !Erzeugen von prob fuer Best-N
1205   forall(i in Letters | help_count(i) < min_entries_per_row) do
1206     missing_entries := min_entries_per_row - help_count(i);
1207     forall(k in Letters) do
1208       current_prob(k) := bigram_prob_orig(i,k);
1209     end-do
1210     while(missing_entries > 0) do
1211       current_max := max(k in Letters | current_prob(k) < threshold) current_prob(k);
1212       forall(k in Letters | current_prob(k) = current_max) do
1213         if(missing_entries > 0) then
1214           bigram_prob(i,k) := bigram_prob_orig(i,k);
1215           missing_entries -= 1;
1216           current_prob(k) := 0;
1217         end-if
1218       end-do
1219     end-do
1220   end-do
1221 elif(PROB_HANDLING = 'global_best') then
1222   forall(i in Letters, k in Letters) bigram_prob_copy(i,k) := bigram_prob_orig(i,k);
1223   missing_entries := global_best;
1224   while(missing_entries > 0) do
1225     current_max := max(i in Letters, k in Letters) bigram_prob_copy(i,k);
1226     forall(i in Letters, k in Letters | bigram_prob_copy(i,k) = current_max) do
1227       if(missing_entries > 0) then
1228         bigram_prob(i,k) := bigram_prob_orig(i,k);
1229         missing_entries -= 1;
1230         bigram_prob_copy(i,k) := 0;
1231       end-if
1232     end-do
1233   end-do
1234 elif(PROB_HANDLING = 'global_best_but_max_entries_per_row') then
1235   forall(i in Letters, k in Letters) bigram_prob_copy(i,k) := bigram_prob_orig(i,k);
1236   forall(i in Letters) help_count(i) := 0;
1237   missing_entries := global_best;
1238   while(missing_entries > 0) do
1239     current_max := max(i in Letters, k in Letters) bigram_prob_copy(i,k);
1240     forall(i in Letters, k in Letters | bigram_prob_copy(i,k) = current_max) do
1241       if (missing_entries > 0 and help_count(i) < max_entries_per_row) then
1242         bigram_prob(i,k) := bigram_prob_orig(i,k);
1243         missing_entries -= 1;
1244         help_count(i) += 1;
1245         bigram_prob_copy(i,k) := 0;
1246       else
1247         bigram_prob_copy(i,k) := 0;
1248       end-if
1249     end-do
1250   end-do
1251 elif(PROB_HANDLING = 'entries_per_row') then
1252   !Erzeugen von prob fuer Best-N
1253   forall(i in Letters) do
1254     missing_entries := entries_per_row;
1255     forall(k in Letters) do
```

```

1256     current_prob(k) := bigram_prob_orig(i,k);
1257 end-do
1258 while(missing_entries > 0) do
1259     current_max := max(k in Letters) current_prob(k);
1260     forall(k in Letters | current_prob(k) = current_max) do
1261         if(missing_entries > 0) then
1262             bigram_prob(i,k) := bigram_prob_orig(i,k);
1263             missing_entries -= 1;
1264             current_prob(k) := 0;
1265         end-if
1266     end-do
1267 end-do
1268 end-do
1269 elif(PROB_HANDLING = 'entries_per_row_plus_global_best') then
1270     forall(i in Letters,k in Letters) bigram_prob_copy(i,k) := bigram_prob_orig(i,k);
1271     forall(i in Letters) do
1272         missing_entries := entries_per_row;
1273         while(missing_entries > 0) do
1274             current_max := max(k in Letters) bigram_prob_copy(i,k);
1275             forall(k in Letters | bigram_prob_copy(i,k) = current_max) do
1276                 if(missing_entries > 0) then
1277                     bigram_prob(i,k) := bigram_prob_orig(i,k);
1278                     missing_entries -= 1;
1279                     bigram_prob_copy(i,k) := 0;
1280                 end-if
1281             end-do
1282         end-do
1283     end-do
1284     missing_entries := global_best;
1285     while(missing_entries > 0) do
1286         current_max := max(i in Letters,k in Letters) bigram_prob_copy(i,k);
1287         forall(i in Letters,k in Letters | bigram_prob_copy(i,k) = current_max) do
1288             if(missing_entries > 0) then
1289                 bigram_prob(i,k) := bigram_prob_orig(i,k);
1290                 missing_entries -= 1;
1291                 bigram_prob_copy(i,k) := 0;
1292             end-if
1293         end-do
1294     end-do
1295 else
1296     forall(i in Letters,k in Letters) bigram_prob(i,k) := bigram_prob_orig(i,k);
1297 end-if
1298 forall(i in Letters,k in Letters | bigram_prob_orig(i,k)<>0) do
1299     number_nonzero_entries_orig += 1;
1300 end-do
1301 forall(i in Letters,k in Letters | bigram_prob(i,k)<>0) do
1302     number_nonzero_entries += 1;
1303 end-do
1304 end-procedure
1305
1306 !*****
1307 !Prozedur um die Einstellungen anzuzeigen
1308 procedure show_setting
1309     writeln('-----SETTING-----');
1310     if(MESSLEV >=1) then
1311         writeln('Gegebenes Tastaturlayout:');
1312         writeln(' Name: ',Keyboard);
1313         writeln(' Form: ',Shape);
1314     end-if
1315     if(MESSLEV >=2) then
1316         writeln(' Anzahl an Buchstaben: ',numberofletters);
1317         writeln(' Anzahl der Tasten: ',numberofkeys);
1318     end-if
1319     if(MESSLEV >=1) then
1320         writeln(' ');
1321         writeln('Grundsatzliche Daten:');
1322         writeln(' Zielfunktion: ',Objective);
1323         write(' Geraeteposition: ',Position);
1324         if(Position='hybrid') then
1325             writeln(' (alpha$=$',alpha,'$, $ beta$=$',beta,'')');
1326         else
1327             writeln(' ');
1328         end-if

```

B Quellcode

```
1329     writeln(' Schreibweise: ',Writing_method);
1330 end-if
1331 if(MESSLEV >=2 and Writing_method <> 'simple') then
1332     writeln(' ');
1333     writeln('Einstellungen fuer die Optimierung: ');
1334     writeln(' Implizites Nullsetzen mancher Variablen: ',Implicit_zero);
1335     writeln(' Untere Schranke an x: ', X_LowB);
1336     writeln(' Obere Schranke an x: ', X_UpB);
1337     writeln(' Adams-Johnson-Nebenbedingungen: ', AJ_Con);
1338     writeln(' Paarweise Gleichheit von x: ', X_Equ);
1339     writeln(' Gleichheit von x und y: ', XY_Equ);
1340     writeln(' Nebenbedingung fuer Buchstabenpaare: ', Letter_Letter_Con);
1341     writeln(' Nebenbedingung fuer Tastenpaare: ', Key_Key_Con);
1342     writeln(' Nebenbedingung fuer Buchstabe-Taste-paare: ', Letter_Key_Con);
1343     writeln(' Ganzzahligkeit: ', Binaryness);
1344     writeln(' Iterationen der Heuristik: ', Heuristic_Iter);
1345     if(Fix_Letter <> 0 and MESSLEV>=2) then
1346         writeln(' ',Fix_Letter,'. Buchstaben (',Lettername(Fix_Letter),') auf die rechte Tastaturhaelfte fixieren');
1347     else
1348         writeln(' Keinen Buchstaben fixieren');
1349     end-if
1350 write(' Handhabung der Wahrscheinlichkeitsmatrix: ', PROB_HANDLING);
1351 if(PROB_HANDLING = 'threshold') then
1352     writeln(' mit WS-Schranke $=$ ',threshold,'%');
1353 elif(PROB_HANDLING = 'threshold_but_min_entries_per_row') then
1354     writeln(' mit WS-Schranke $=$ ',threshold,'% und mindestens ',min_entries_per_row,' Eintraegen pro Zeile');
1355 elif(PROB_HANDLING = 'global_best') then
1356     writeln(' mit den ',global_best,' besten Eintraegen');
1357 elif(PROB_HANDLING = 'global_best_but_max_entries_per_row') then
1358     writeln(' mit den ',global_best,' besten Eintraegen,$$ aber maximal ',max_entries_per_row,' pro Zeile');
1359 elif(PROB_HANDLING = 'entries_per_row') then
1360     writeln(' mit den ',entries_per_row,' besten Eintraege jeder Zeile');
1361 elif(PROB_HANDLING = 'entries_per_row_plus_global_best') then
1362     writeln(' mit den ',entries_per_row,' besten Eintraege jeder Zeile plus die ',global_best,' besten sonstigen
    Eintraege');
1363 else
1364     writeln(' ');
1365 end-if
1366 writeln(' (damit sind ',number_nonzero_entries,' Bigrammwahrscheinlichkeiten ungleich Null...');
1367 writeln(' und ',number_nonzero_entries_orig-number_nonzero_entries,' Bigrammwahrscheinlichkeiten wurden aktiv
    unterdrueckt');
1368 end-if
1369 writeln('-----ENDE SETTING-----');
1370 end-procedure
1371
1372 !*****
1373 !Prozedur um die Ergebnisse der Matrix-Analyse anzuzeigen
1374 procedure show_analysis
1375     if (MESSLEV >= 3) then
1376         writeln(' ');
1377         writeln('-----ANALYSE-----');
1378     end-if
1379     if(MESSLEV >= 3 and MESSLEV < 4 and Writing_method<>'simple') then
1380         writeln('Analyse der verwendeten Bigrammwahrscheinlichkeiten');
1381         analyse_probability_matrix(bigram_prob);
1382     end-if
1383     if(MESSLEV >=4) then
1384         writeln('Distanz zum Startpunkt:');
1385         writeln(dist_start);
1386         writeln(' ');
1387         writeln('Bewegungszeit zum Startpunkt:');
1388         writeln(time_start);
1389         writeln(' ');
1390         writeln('Buchstabenwahrscheinlichkeiten:');
1391         writeln(letter_prob);
1392         if(Writing_method <> 'simple') then
1393             writeln(' ');
1394             writeln('Distanz der Tasten untereinander:');
1395             show_dist_or_time_matrix(dist_keys);
1396             writeln('Bewegungszeit der Tasten untereinander:');
1397             show_dist_or_time_matrix(time_keys);
1398             if(PROB_HANDLING <> 'none') then
1399                 writeln('Urspruengliche Bigrammwahrscheinlichkeiten');
```

```

1400     show_probability_matrix(bigram_prob_orig);
1401     writeln('Analyse der urspruenglichen Bigrammwahrscheinlichkeiten');
1402     analyse_probability_matrix(bigram_prob_orig);
1403     writeln('Neue Bigrammwahrscheinlichkeiten');
1404     show_probability_matrix(bigram_prob);
1405     writeln('Analyse der neuen Bigrammwahrscheinlichkeiten');
1406     analyse_probability_matrix(bigram_prob);
1407     else
1408         writeln('Bigrammwahrscheinlichkeiten');
1409         show_probability_matrix(bigram_prob_orig);
1410         writeln('Analyse der Bigrammwahrscheinlichkeiten');
1411         analyse_probability_matrix(bigram_prob_orig);
1412     end-if
1413 end-if
1414 end-if
1415 if (MESSLEV >= 3) then
1416     writeln('-----ENDE ANALYSE-----');
1417 end-if
1418 end-procedure
1419
1420 !*****
1421 !Prozedur um das Modell anzuzeigen
1422 procedure show_model
1423     writeln(' ');
1424     writeln('Modell:');
1425     writeln(' Anzahl an y-Variablen: ', numberofvar_y);
1426     writeln(' Anzahl an x-Variablen: ', numberofvar_x, ' (davon ', numberofvar_implzero, ' per Gleichungs-NB auf Null gesetzt
1427     ');
1428     writeln(' ');
1429     writeln(' Anzahl an Ungleichungs-NB: ', numberofconstraints_inequ);
1430     writeln(' Anzahl an Gleichungs-NB: ', numberofconstraints_equ, ' (davon ', numberofvar_implzero, ' x-ist-Null-Bedingungen
1431     ');
1432     writeln(' ');
1433 end-procedure
1434
1435 !*****
1436 !Prozedur um die Ergebnisse der Optimierung anzuzeigen
1437 procedure show_result
1438     !AUSGABE: Status, Variablen und Tastatur
1439     if (MESSLEV >= 2) then
1440         !AUSGABE: Rechenzeit
1441         runtime := gettime - starttime;
1442         writeln(' ');
1443         writeln('Resultate (fuer obige Einstellungen): ');
1444         write(' Status: ');
1445         if getprobstat = XPRS_OPT then
1446             writeln('Solution optimal');
1447             writeln(' Zielfunktionswert: ', getobjval);
1448             writeln(' Benoetigte Rechenzeit: ', runtime, 's');
1449             if (Heuristic_Iter > 0) then
1450                 writeln(' ');
1451                 writeln(' Optimierte Tastatur:');
1452                 visualize_keyboard(sol_optimization);
1453                 if (show_assignment_array = 'on') then
1454                     writeln(' Errechnete Belegung: ', create_assignment_array(sol_optimization));
1455                     writeln(' ');
1456                 end-if
1457                 show_keyboard_characteristics(sol_optimization);
1458             end-if
1459             if (MESSLEV >= 5) then
1460                 show_solution_part;
1461             end-if
1462             if (MESSLEV >= 6) then
1463                 show_solution_complete;
1464             end-if
1465         elif getprobstat = XPRS_inF then
1466             writeln('Problem infeasible');
1467         elif getprobstat = XPRS_UNB then
1468             writeln('Problem unbounded');
1469         else
1470             writeln('No opt. sol.: Status $=$ ', getprobstat);
1471             if (Heuristic_Iter > 0) then

```

B Quellcode

```
1471     writeln(' ');
1472     writeln(' Beste gefundene Tastatur (suboptimal):');
1473     visualize_keyboard(sol_optimization);
1474     if (show_assignment_array = 'on') then
1475         writeln(' ');
1476         writeln('Errechnete (suboptimale) Belegung: ',create_assignment_array(sol_optimization));
1477         writeln(' ');
1478     end-if
1479     show_keyboard_characteristics(sol_optimization);
1480 end-if
1481 writeln(' Eigenschaften der suboptimalen Loesung:');
1482 if (MESSLEV >= 5) then
1483     show_solution_part;
1484 end-if
1485 if (MESSLEV >= 6) then
1486     show_solution_complete;
1487 end-if
1488 end-if
1489 end-if
1490 end-procedure
1491
1492 !*****
1493 !Prozedur um eine Tastenbelegung anzuzeigen
1494 procedure visualize_keyboard(sol: array(Letters,Keys) of real) !sodass ich alte und neue Loesung grafisch habe)
1495     declarations
1496         helpidx_row, helpidx_col: integer;
1497         max_col, spacenumber: integer;
1498     end-declarations
1499     helpidx_row := 0;
1500     helpidx_col := 0;
1501     if (Shape = 'free') then
1502         max_col := max(i in 1..Rows) Columns_free(i);
1503         helpidx_row += 1;
1504         forall(k in Keys) do
1505             spacenumber := max_col - Columns_free(helpidx_row);
1506             helpidx_col += 1;
1507             if helpidx_col = 1 then
1508                 if(spacenumber = 1) then write(' '); end-if
1509                 if(spacenumber = 2) then write(' '); end-if
1510                 if(spacenumber = 3) then write(' '); end-if
1511                 if(spacenumber = 4) then write(' '); end-if
1512                 if(spacenumber = 5) then write(' '); end-if
1513                 if(spacenumber = 6) then write(' '); end-if
1514                 write(' |');
1515             end-if
1516             if(Position = 'horizontal' and helpidx_col = 1+floor(Columns_free(helpidx_row)/2)) then
1517                 write(' |');
1518             end-if
1519             forall(l in Letters | abs(1-sol(l,k)) < tol ) do
1520                 write(Lettername(l),'|');
1521             end-do
1522             if abs(sum(l in Letters) sol(l,k)) < tol then
1523                 write('-|');
1524             end-if
1525             if(helpidx_col = Columns_free(helpidx_row)) then
1526                 writeln(' ');
1527                 helpidx_col := 0;
1528                 helpidx_row += 1;
1529             end-if
1530         end-do
1531     end-if
1532     if (Shape = 'rectangular') then
1533         forall(k in Keys) do
1534             helpidx_col += 1;
1535             if helpidx_col = 1 then
1536                 write(' |');
1537             end-if
1538             if(Position = 'horizontal' and helpidx_col = 1+floor(Columns_rect/2)) then
1539                 write(' |');
1540             end-if
1541             forall(l in Letters | abs(1-sol(l,k)) < tol) do
1542                 write(Lettername(l),'|');
1543             end-do
1544         end-do
1545     end-if
1546 end-procedure
```

```

1544     if(abs(sum(l in Letters) sol(l,k)) < tol) then
1545         write('-|');
1546     end-if
1547     if(helpidx_col = Columns_rect) then
1548         writeln(' ');
1549         helpidx_col := 0;
1550     end-if
1551 end-do
1552 end-if
1553 end-procedure
1554
1555 !*****
1556 !Funktion um die Belegung von zwei Tasten zu vertauschen (fuer $2$-opt)
1557 function two_opt_heuristic(sol:array(Letters,Keys) of real): array(Letters,Keys) of real
1558     declarations
1559         new_sol: array(Letters,Keys) of real;
1560         key1, key2: integer;
1561         occupied: boolean;
1562     end-declarations
1563     occupied := false;
1564     while(occupied = false) do
1565         key1 := ceil(random*numberofkeys);
1566         if( or(i in Letters) abs(1-sol(i,key1)) < tol) then
1567             occupied := true;
1568         end-if
1569     end-do
1570     key2 := ceil(random*numberofkeys);
1571     while(key2 = key1) do
1572         key2 := ceil(random*numberofkeys);
1573     end-do
1574     forall(i in Letters) do
1575         forall(j in Keys | j<>key1 and j<>key2) do
1576             new_sol(i,j) := sol(i,j);
1577         end-do
1578         new_sol(i,key1) := sol(i,key2);
1579         new_sol(i,key2) := sol(i,key1);
1580     end-do
1581     returned := new_sol;
1582 end-function
1583
1584 !*****
1585 !Funktion um die 2-opt-Heuristik anzuwenden
1586 function apply_heuristic(max_iter:integer, sol: array(Letters,Keys) of real): array(Letters,Keys) of real
1587     declarations
1588         iter: integer;
1589         current_best_sol: array(Letters,Keys) of real;
1590         sol_candidate: array(Letters,Keys) of real;
1591     end-declarations
1592     iter := 0;
1593     current_best_sol := sol;
1594     if(MESSLEV>=2) then
1595         writeln('Heuristik gestartet...');
1596     end-if
1597     if(numberofkeys=1) then
1598         iter := max_iter+1;
1599         writeln('Heuristik abgebrochen,$ da nur eine Taste verfuegbar.');
```

B Quellcode

```
1617     if(Objective = 'travel-distance') then
1618         if(keyboard_quality_dist(sol_candidate) < keyboard_quality_dist(current_best_sol)) then
1619             current_best_sol := sol_candidate;
1620             better_sol := true;
1621         end-if
1622     elif(Objective = 'travel-time') then
1623         if(keyboard_quality_time(sol_candidate) < keyboard_quality_time(current_best_sol)) then
1624             current_best_sol := sol_candidate;
1625             better_sol := true;
1626         end-if
1627     end-if
1628 end-do
1629 if(numberofkeys <> 1) then
1630     if(MESSLEV=1) then
1631         writeln('Bessere Loesung gefunden');
1632     elif(MESSLEV>=2) then
1633         writeln('Heuristik abgeschlossen...');
1634         writeln(' ');
1635         writeln('Resultat der Heuristik: ');
1636         if(better_sol = true) then
1637             writeln(' Bessere Loesung gefunden');
1638         elif(better_sol = false) then
1639             writeln(' Keine bessere Loesung gefunden');
1640         end-if
1641         writeln(' Laufzeit der Heuristik: ', gettime-runtime-starttime, 's');
1642     end-if
1643 end-if
1644 returned := current_best_sol;
1645 end-function
1646
1647 !*****
1648 !Prozedur um Teile der Optimierungsloesung anzuzeigen
1649 procedure show_solution_part
1650     writeln(' ');
1651     writeln(' Variablen mit Loesungswert $1$:');
1652     forall(i in Letters, j in Keys | abs(1-y(i,j).sol) < tol) do
1653         writeln(' y(' , i, '$, $', j, ') : ', y(i,j).sol);
1654     end-do
1655     forall(i in Letters, j in Keys, k in Letters, l in Keys | abs(1-x(i,j,k,l).sol) < tol) do
1656         writeln(' x(' , i, '$, $', j, '$, $', k, '$, $', l, ') : ', x(i,j,k,l).sol);
1657     end-do
1658 end-procedure
1659
1660 !*****
1661 !Prozedur um die komplette Optimierungsloesung anzuzeigen
1662 procedure show_solution_complete
1663     writeln(' ');
1664     writeln(' Loesungswerte der Variablen im Detail');
1665     forall(i in Letters, j in Keys) do
1666         writeln(' y(' , i, '$, $', j, ') : ', y(i,j).sol);
1667     end-do
1668     forall(i in Letters, j in Keys, k in Letters, l in Keys) do
1669         writeln(' x(' , i, '$, $', j, '$, $', k, '$, $', l, ') : ', x(i,j,k,l).sol);
1670     end-do
1671 end-procedure
1672
1673 !*****
1674 !Funktion um aus der eingelesenen Belegung einen Loesungsvektor zu erstellen
1675 function create_individual_keyboard_vector(assignment:array(Letters) of integer): array(Letters,Keys) of real
1676     declarations
1677         keyboard_vector: array(Letters,Keys) of real;
1678     end-declarations
1679     forall(i in Letters, j in Keys) do
1680         keyboard_vector(i,j) := 0;
1681     end-do
1682     forall(i in Letters) do
1683         keyboard_vector((i),assignment(i)) := 1;
1684     end-do
1685     returned := keyboard_vector;
1686 end-function
1687 !*****
1688 end-model
```

Literaturverzeichnis

- [Adams et al. 2004] ADAMS, Warren P. ; FORRESTER, Richard J. ; GLOVER, Fred W.: Comparisons and enhancement strategies for linearizing mixed 0-1 quadratic programs. In: *Discrete Optimization 1* (2004), Nr. 2, S. 99–120
- [Adams u. Johnson 1994] ADAMS, Warren P. ; JOHNSON, Terri A.: Improved linear programming-based lower bounds for the quadratic assignment problem. In: *DIMACS series in discrete mathematics and theoretical computer science 16* (1994), S. 43–77
- [Al Faraj et al. 2009] AL FARAJ, Khaldoun ; MOJAHID, Mustapha ; VIGOUROUX, Nadine: Bigkey: A virtual keyboard for mobile devices. In: *Human-Computer Interaction. Ambient, Ubiquitous and Intelligent Interaction*. Springer, 2009, S. 3–10
- [Bao et al. 2011] BAO, Patti ; PIERCE, Jeffrey ; WHITTAKER, Stephen ; ZHAI, Shumin: Smart phone use by non-mobile business users. In: *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* ACM, 2011, S. 445–454
- [Barkow 2007] BARKOW, Ruben: *ENTI-Key++*. <http://entiskey.z11.de/>. Version: 2007, Abruf: 14.11.2013
- [Behrens 2004] BEHRENS, Hanno: *Neo – ein ergonomisches Tastaturlayout*. <http://www.neo-layout.org/>. Version: 2004, Abruf: 30.11.2013
- [Bi et al. 2012] BI, Xiaojun ; SMITH, Barton A. ; ZHAI, Shumin: Multilingual touchscreen keyboard design and optimization. In: *Human-Computer Interaction 27* (2012), Nr. 4, S. 352–382
- [Billera u. Sarangarajan 1996] BILLERA, Louis J. ; SARANGARAJAN, A.: The combinatorics of permutation polytopes. In: *Formal power series and algebraic combinatorics (New Brunswick, NJ, 1994)* Bd. 24. Amer. Math. Soc., Providence, RI, 1996, S. 1–23
- [Brewbaker 2008] BREWBAKER, Chad R.: Optimizing stylus keyboard layouts with a genetic algorithm: customization and internationalization / Dept. of Computer Science, Iowa State University. 2008. – Forschungsbericht

- [Burkard et al. 1998] BURKARD, Rainer E. ; ÇELA, Eranda ; PARDALOS, Panos M. ; PITSOULIS, Leonidas S.: The Quadratic Assignment Problem. In: *Handbook of combinatorial optimization* 4 (1998), S. 241–337
- [Burkard u. Offermann 1977] BURKARD, Rainer E. ; OFFERMANN, J: Entwurf von Schreibmaschinentastaturen mittels quadratischer Zuordnungsprobleme. In: *Mathematical Methods of Operations Research* 21 (1977), Nr. 4, S. 121–132
- [Buzing 2003] BUZING, Pieter: Comparing different keyboard layouts: aspects of qwerty, dvorak and alphabetical keyboards / Delft University of Technology. 2003. – Forschungsbericht
- [Cashmore et al. 2008] CASHMORE, Lisa ; UOMINI, Natalie ; CHAPELAIN, Amandine: The evolution of handedness in humans and great apes: a review and current issues. In: *Journal of anthropological sciences* 86 (2008), S. 7–35
- [Clarkson et al. 2007] CLARKSON, Edward ; LYONS, Kent ; CLAWSON, James ; STARNER, Thad: Revisiting and validating a model of two-thumb text entry. In: *Proceedings of the SIGCHI conference on Human factors in computing systems ACM*, 2007, S. 163–166
- [Coleman 2006] COLEMAN, Shai: *Colemak keyboard layout*. <http://colemak.com/>. Version: 2006, Abruf: 30.11.2013
- [Curran et al. 2005] CURRAN, Kevin ; WOODS, Derek ; RIORDAN, Barry O.: Investigating text input methods for mobile phones. In: *Telematics and Informatics* 23 (2005), Nr. 1, S. 1–21
- [David 1985] DAVID, Paul A.: Clio and the Economics of QWERTY. In: *The American economic review* 75 (1985), Nr. 2, S. 332–337
- [David 1997] DAVID, Paul A.: Path Dependence and the Quest for Historical Economics: One More chorus of Ballad of QWERTY / Economics Group, Nuffield College, University of Oxford. 1997. – Forschungsbericht
- [Davis 2005] DAVIS, Will: *The Union Typewriter Company*. <http://www.willdavis.org/Union.html>. Version: November 2005, Abruf: 05.11.2013
- [Dell'Amico et al. 2009] DELL'AMICO, Mauro ; DÍAZ, José Carlos Díaz ; IORI, Manuel ; MONTANARI, Roberto: The single-finger keyboard layout problem. In: *Computers & Operations Research* 36 (2009), Nr. 11, S. 3002–3012
- [Dittmann et al. 2007] DITTMANN, Jürgen ; SIEBERT, Hedy ; STAIGER-ANLAUF, Yvonne: Medium und Kommunikationsform – am Beispiel der SMS. In: *Networx* 50 (2007), S. 5–55

- [Du et al. 2005] DU, Huan ; OGGIER, Thierry ; LUSTENBERGER, Felix ; CHARBON, Edoardo: A Virtual Keyboard Based on True-3D Optical Ranging. In: *Proceedings of the British Machine Vision Conference*, 2005
- [Dunlop u. Levine 2012] DUNLOP, Mark ; LEVINE, John: Multidimensional pareto optimization of touchscreen keyboards for speed, familiarity and improved spell checking. In: *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems* ACM, 2012, S. 2669–2678
- [Dvorak 1943] DVORAK, August: There is a better typewriter keyboard. In: *National Business Education Quarterly* 11 (1943), S. 51–58
- [Fitts 1954] FITTS, Paul M.: The information capacity of the human motor system in controlling the amplitude of movement. In: *Journal of experimental psychology* 47 (1954), Nr. 6, S. 381–391
- [Francis u. Rash 2003] FRANCIS, Gregory ; RASH, Clarence: Optimization of keyboard design for specialized text entry. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* Bd. 47 SAGE Publications, 2003, S. 734–736
- [Getschow et al. 1986] GETSCHOW, C. O. ; ROSEN, M. J. ; GOODENOUGH-TREPAGNIER, C: A systematic approach to design a minimum distance alphabetical keyboard. In: *Proc. RESNA (Rehabilitation Engineering Society of North America) 9th Annual Conference*, 1986, S. 396–398
- [Goel et al. 2012] GOEL, Mayank ; FINDLATER, Leah ; WOBROCK, Jacob: WalkType: using accelerometer data to accomodate situational impairments in mobile touch screen text entry. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* ACM, 2012, S. 2687–2696
- [Hirsch 1970] HIRSCH, Richard S.: Effects of standard versus alphabetical keyboard formats on typing performance. In: *Journal of Applied Psychology* 54 (1970), Nr. 6, S. 484–490
- [Hunter et al. 2000] HUNTER, Michael ; ZHAI, Shumin ; SMITH, Barton A.: Physics-based graphical keyboard design. In: *CHI'00 extended abstracts on Human factors in computing systems* ACM, 2000, S. 157–158
- [Ichbiah 1996] ICHBIAH, Jean D.: *Method for designing an ergonomic one-finger keyboard and apparatus therefor*. Jan 1996. – US Patent 5,487,616
- [Jain u. Bhattacharya 2010] JAIN, Siddharth ; BHATTACHARYA, Samit: Virtual keyboard layout optimization. In: *Students' Technology Symposium (TechSym)* IEEE, 2010, S. 312–317

- [Jüngst 2008] JÜNGST, Johannes: *Strokes: Entwicklung einer optimierten Miniatur-Tastatur für die Eingabe mit zwei Daumen*, Ludwig-Maximilians-Universität München, Diss., 2008
- [Kinkead 1975] KINKEAD, Robin: Typing speed, keying rates, and optimal keyboard layouts. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* Bd. 19 SAGE Publications, 1975, S. 159–161
- [Kölsch u. Turk 2002] KÖLSCH, Mathias ; TURK, Matthew: Keyboards without keyboards: A survey of virtual keyboards. In: *Workshop on Sensing and Input for Media-centric Systems, Santa Barbara, CA, 2002*
- [Koopmans u. Beckmann 1957] KOOPMANS, Tjalling C. ; BECKMANN, Martin: Assignment problems and the location of economic activities. In: *Econometrica: Journal of the Econometric Society* 25 (1957), S. 53–76
- [Kroemer 2001] KROEMER, Karl H.: Keyboards and keying. An annotated bibliography of the literature from 1878 to 1999. In: *Universal Access in the Information Society* 1 (2001), Nr. 2, S. 99–160
- [Lee u. Ma 2004] LEE, Chi-Guhn ; MA, Zhong: The generalized quadratic assignment problem / Dept. Mechanical & Industrial Engineering, Univ. Toronto, Canada. 2004. – Forschungsbericht. – 20 S.
- [Leshner u. Moulton 2000] LESHER, Gregory W. ; MOULTON, Bryan J.: A method for optimizing single-finger keyboards. In: *Proceedings of the RESNA 2000 Annual Conference*, 2000, S. 91–93
- [Leshner et al. 1998] LESHER, Gregory W. ; MOULTON, Bryan J. ; HIGGINBOTHAM, D J.: Optimal character arrangements for ambiguous keyboards. In: *IEEE Transactions on Rehabilitation Engineering* 6 (1998), Nr. 4, S. 415–423
- [Lessing 2010] LESSING, Hans-Erhard: *Karl Drais: zwei Räder statt vier Hufe*. Braun, 2010
- [Lewis et al. 1999a] LEWIS, James R. ; KENNEDY, Peter J. ; LALOMIA, Mary J.: Development of a digram-based typing key layout for single-finger/stylus input. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* Bd. 43 SAGE Publications, 1999, S. 415–419
- [Lewis et al. 1999b] LEWIS, James R. ; LALOMIA, Mary J. ; KENNEDY, Peter J.: Evaluation of typing key layouts for stylus input. In: *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* Bd. 43 SAGE Publications, 1999, S. 420–424

- [Li et al. 2011] LI, Frank Chun Y. ; GUY, Richard T. ; YATANI, Koji ; TRUONG, Khai N.: The 1line keyboard: A QWERTY layout in a single line. In: *Proceedings of the 24th annual ACM symposium on User interface software and technology* ACM, 2011, S. 461–470
- [Li et al. 2006] LI, Yanzhi ; CHEN, Lijuan ; GOONETILLEKE, Ravindra S.: A heuristic-based approach to optimize keyboard design for single-finger keying applications. In: *International journal of industrial ergonomics* 36 (2006), Nr. 8, S. 695–704
- [Liebowitz u. Margolis 1990] LIEBOWITZ, Stan J. ; MARGOLIS, Stephen E.: The fable of the keys. In: *Journal of Law and Economics* 33 (1990), Nr. 1, S. 1–25
- [Light u. Anderson 1993] LIGHT, Lissa ; ANDERSON, Peter: Typewriter keyboards via simulated annealing. In: *AI Expert* 8 (1993), Nr. 9, S. 20–27
- [MacKenzie u. Soukoreff 2002a] MACKENZIE, I S. ; SOUKOREFF, R W.: A model of two-thumb text entry. In: *Graphics interface*, 2002, S. 117–124
- [MacKenzie u. Soukoreff 2002b] MACKENZIE, I S. ; SOUKOREFF, R W.: Text entry for mobile computing: Models and methods, theory and practice. In: *Human–Computer Interaction* 17 (2002), Nr. 2, S. 147–198
- [MacKenzie u. Zhang 1999] MACKENZIE, I S. ; ZHANG, Shawn X.: The design and evaluation of a high-performance soft keyboard. In: *Proceedings of the SIGCHI conference on Human factors in computing systems* ACM, 1999, S. 25–31
- [MacKenzie et al. 1999] MACKENZIE, I S. ; ZHANG, Shawn X. ; SOUKOREFF, R W.: Text entry using soft keyboards. In: *Behaviour & information technology* 18 (1999), Nr. 4, S. 235–244
- [MacKenzie 1989] MACKENZIE, Scott I.: A note on the information-theoretic basis for Fitts law. In: *Journal of Motor Behavior* 21 (1989), Nr. 3, S. 323–330
- [Magnien et al. 2004] MAGNIEN, Laurent ; BOURAOUI, Jean L. ; VIGOUROUX, Nadine: Mobile text input with soft keyboards: optimization by means of visual clues. In: *Mobile Human-Computer Interaction-MobileHCI 2004*. Springer, 2004, S. 337–341
- [Malt 1977] MALT, Lillian G.: Keyboard design in the electronic era. In: *PIRA Symposium, Developments in Data Capture and Photocomposition*, 1977
- [Menzel et al. 2005] MENZEL, Justus ; RIEGER, Torsten ; STOYE, Nils P.: *RISTOME -Die Revolution der Tastatur*. <http://www.ristome.de/>. Version:2005, Abruf: 15.11.2013
- [Michaels 1971] MICHAELS, S E.: QWERTY versus alphabetic keyboards as a function of typing skill. In: *Human Factors: The Journal of the Human Factors and Ergonomics Society* 13 (1971), Nr. 5, S. 419–426

- [Mill 1714] MILL, Henry: *Machine for Transcribing Letters*. Patent no. 395 (GB), 1714
- [Noyes 1998] NOYES, Jan: QWERTY - the immortal keyboard. In: *Computing & Control Engineering Journal* 9 (1998), Nr. 3, S. 117–122
- [Oommen et al. 1991] OOMMEN, B J. ; VALIVETI, Radhakrishna S. ; ZGIERSKI, Jack R.: An adaptive learning solution to the keyboard optimization problem. In: *IEEE Transactions on Systems, Man and Cybernetics* 21 (1991), Nr. 6, S. 1608–1618
- [Oulasvirta et al. 2013] OULASVIRTA, Antti ; REICHEL, Anna ; LI, Wenbin ; ZHANG, Yan ; BACHYNSKYI, Myroslav ; VERTANEN, Keith ; KRISTENSSON, Per O.: Improving two-thumb text entry on touchscreen devices. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* ACM, 2013, S. 2765–2774
- [Pols 2013] POLS, Axel: *New sales record for smartphones in Germany projected*. Press Release, September 2013
- [Raynal u. Vigouroux 2005] RAYNAL, Mathieu ; VIGOUROUX, Nadine: Genetic algorithm to generate optimized soft keyboard. In: *CHI'05 extended abstracts on Human factors in computing systems* ACM, 2005, S. 1729–1732
- [Robert 2001] ROBERT, Paul: *The Virtual Typewriter Museum*. <http://www.typewritermuseum.org>. Version: Juni 2001, Abruf: 28.10.2013
- [Sahni u. Gonzalez 1976] SAHNI, Sartaj ; GONZALEZ, Teofilo: P-complete approximation problems. In: *Journal of the ACM (JACM)* 23 (1976), Nr. 3, S. 555–565
- [Schenk u. Rigoll 2010] SCHENK, Joachim ; RIGOLL, Gerhard: *Mensch-Maschine-Kommunikation: Grundlagen von sprach-und bildbasierten Benutzerschnittstellen*. Springer, 2010
- [Schukat-Talamazzini 1995] SCHUKAT-TALAMAZZINI, Ernst G.: *Automatische Spracherkennung*. Vieweg Verlag, 1995
- [Silfverberg et al. 2000] SILFVERBERG, Miika ; MACKENZIE, I S. ; KORHONEN, Panu: Predicting text entry speed on mobile phones. In: *Proceedings of the SIGCHI conference on Human factors in computing systems* ACM, 2000, S. 9–16
- [Smith u. Zhai 2001] SMITH, Barton A. ; ZHAI, Shumin: Optimised virtual keyboards with and without alphabetical ordering-a novice user study. In: HIROSE, Michitake (Hrsg.) ; IOS Press (Veranst.): *Human-computer Interaction: INTERACT'01: IFIP TC. 13 International Conference on Human-Computer Interaction, 9th-13th July 2001, Tokyo, Japan* IOS Press, 2001, S. 92

- [Soukoreff 2002] SOUKOREFF, Robert W.: *Text Entry for Mobile Systems: Models, measures and analyses for text entry research*, York University, Toronto, Canada, Diss., 2002
- [Swype-Inc. 2010] SWYPE-INC.: *Swype Input Method*. <http://www.swype.com/>.
Version: 2010, Abruf: 01.12.2013
- [Trost 2013] TROST, Stefan: *WordCreator*. sttmedia.com/downloads/WordCreatorWin.zip. Version: 2013, Abruf: 27.10.2013
- [Universität Leipzig 2010] UNIVERSITÄT LEIPZIG: *Leipzig Corpora Collection*. <http://corpora.uni-leipzig.de/>. Version: 2010. Datei: deu_news_2010_10K-text.tar.gz, Abruf: 27.10.2013
- [Wagner et al. 2003] WAGNER, Marc O. ; YANNOU, Bernard ; KEHL, Steffen ; FEILLET, Dominique ; EGGERS, Jan: Ergonomic modelling and optimization of the keyboard arrangement with an ant colony algorithm. In: *Journal of Engineering Design* 14 (2003), Nr. 2, S. 187–208
- [West 1998] WEST, Leonard J.: The standard and dvorak keyboards revisited: Direct measures of speed / Santa Fe Institute. 1998. – Forschungsbericht
- [Wienecke 2003] WIENECKE, Markus: *Videobasierte Handschrifterkennung*, Universität Bielefeld, Diss., 2003
- [William Soukoreff u. Scott Mackenzie 1995] WILLIAM SOUKOREFF, R ; SCOTT MACKENZIE, I: Theoretical upper and lower bounds on typing speed using a stylus and a soft keyboard. In: *Behaviour & Information Technology* 14 (1995), Nr. 6, S. 370–379
- [Yamada 1980] YAMADA, Hisao: A historical study of typewriters and typing methods: from the position of planning Japanese parallels. In: *Journal of Information Processing* 2 (1980), Nr. 4, S. 175–202
- [Yasuoka u. Yasuoka 2011] YASUOKA, Koichi ; YASUOKA, Motoko: On the Prehistory of QWERTY. In: *ZINBUN* 42 (2011), S. 161–174
- [Zhai et al. 2000] ZHAI, Shumin ; HUNTER, Michael ; SMITH, Barton A.: The metropolis keyboard—an exploration of quantitative techniques for virtual keyboard design. In: *Proceedings of the 13th annual ACM symposium on User interface software and technology ACM*, 2000, S. 119–128
- [Zhai et al. 2002a] ZHAI, Shumin ; HUNTER, Michael ; SMITH, Barton A.: Performance optimization of virtual keyboards. In: *Human–Computer Interaction* 17 (2002), Nr. 2, S. 229–269

[Zhai et al. 2002b] ZHAI, Shumin ; SUE, Alison ; ACCOT, Johnny: Movement model, hits distribution and learning in virtual keyboarding. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves* ACM, 2002, S. 17–24

[Zhang 1998] ZHANG, Shawn X.: *A high performance soft keyboard for mobile systems*, The University of Guelph, Diplomarbeit, 1998

Abbildungsverzeichnis

1	<i>QWERTY-like</i> -Layout	22
2	3x9-Layout	23
3	3x10-Layout	23
4	4x8-Layout	24
5	Aufteilung und spezielle Punkte des <i>QWERTY-like</i> -Layouts in vertikaler Position.	27
6	Aufteilung und spezielle Punkte des <i>QWERTY-like</i> -Layouts in horizontaler Position.	27
7	Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der vereinfachten Schreibweise in vertikaler Geräteposition.	32
8	Distanz-optimale Tastenbelegung für das <i>QWERTY-like</i> -Layout bei vereinfachter Schreibweise im vertikalen Fall.	34
9	Distanz-optimale Tastenbelegung für das 3x9-Layout bei vereinfachter Schreibweise im vertikalen Fall.	34
10	Distanz-optimale Tastenbelegung für das 3x10-Layout bei vereinfachter Schreibweise im vertikalen Fall.	35
11	Distanz-optimale Tastenbelegung für das 4x8-Layout bei vereinfachter Schreibweise im vertikalen Fall.	35
12	Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der realistischen Schreibweise in vertikaler Geräteposition.	36
13	Grafische Darstellung (in <i>IVE XPress</i>) des Verlaufs der Optimierung von Test-Instanz 1 bei ausgeschaltetem Presolving, Heuristiken und Cutstrategien unter Verwendung von <i>VirtualKeyboardOptimization.mos</i>	46
14	Schematische Darstellung der grundlegenden Idee des Ansatzes CUTS. Dabei sind P und P' das ursprüngliche bzw. neue, durch die (rote) Nebenbedingung entstehende Polytop und P_I die konvexe Hülle der zulässigen Punkte (grau) und der Lösung (grün).	47
15	Vier symmetrische Tastenbelegungen (bezüglich der Zielfunktion (3.4) und im geometrischen Sinne).	52

16	Schematische Darstellung der grundlegenden Idee des Ansatzes SYMM und Einfluss der Wahl von T auf die zusätzlichen Nebenbedingungen (rot oder grün) und die ursprüngliche Lösungsmenge (grüne Punkte). Dabei sind P und P' das ursprüngliche bzw. neue Polytop und rote Punkte ehemalige Lösungen.	54
17	Schematische Darstellung der grundlegenden Idee des Ansatzes OBJFCT und Einfluss der Variation des Zielfunktionsvektors ($-c$) auf die ursprüngliche Lösungsmenge (grün). P und P' sind das ursprüngliche bzw. neue Polytop und rote Punkte ehemalige Lösungen.	55
18	Schematische Darstellung der grundlegenden Idee des Ansatzes UPBND. P' ist das neue Polytop, welches aus dem ursprünglichen Polytop P durch die Relaxierung (von NB 2) entsteht. Zulässige Punkte sind grau, Lösungen grün.	57
19	Schematische Darstellung der Kompatibilität der Ansätze CUTS (magenta, orange) und UPBND anhand der Auswirkungen von CUTS auf (P4) (links) und (P5) (rechts). Die ursprünglichen Lösungen (grün) bleiben bei Verwendung von CUTS und/oder UPBND optimale Punkte. Die Punkte, welche durch UPBND zu Lösungen wurden (rot), werden durch CUTS potentiell abgeschnitten.	64
20	Schematische Darstellung der grundlegenden Idee des Ansatzes NOBIN. Die Relaxation der Binärbedingungen (rot) der z_2 -Variable vergrößert den Zulässigkeitsbereich (grau und grün), beeinflusst aber nicht den Wert der z_1 -Komponente von Lösungen (grün).	65
21	Schematische Darstellung der grundlegenden Idee des Ansatzes AJLIN. Dabei sind NB 1' und NB 2' die Nebenbedingungen der neuen Linearisierung, P' das zugehörige Polytop und grüne Punkte Lösungen des Problems.	71
22	Schematische Darstellung der grundlegenden Idee des Ansatzes IMPLZERO. Die nicht-ganzzahlige Variable z_2 wird auf 0 gesetzt, da sie keinen Einfluss auf den Zielfunktionswert hat. Damit verändert sich der Zulässigkeitsbereich (grau und grün), die Menge der Lösungen (grün) und das urspr. Polytop P wird zu P' (Binärbedingungen sind rot).	80
23	Distanz-optimierte Tastenbelegung für das <i>QWERTY-like</i> -Layout bei realistischer Schreibweise im vertikalen Fall.	90
24	Distanz-optimierte Tastenbelegung für das <i>3x9</i> -Layout bei realistischer Schreibweise im vertikalen Fall.	94
25	Distanz-optimierte Tastenbelegung für das <i>3x10</i> -Layout bei realistischer Schreibweise im vertikalen Fall.	94
26	Distanz-optimierte Tastenbelegung für das <i>4x8</i> -Layout bei realistischer Schreibweise im vertikalen Fall.	95

27	Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der vereinfachten Schreibweise in horizontaler Geräteposition.	98
28	Distanz-optimale Tastenbelegung für das <i>QWERTY-like</i> -Layout bei vereinfachter Schreibweise im horizontalen Fall.	99
29	Distanz-optimale Tastenbelegung für das <i>3x9</i> -Layout bei vereinfachter Schreibweise im horizontalen Fall.	99
30	Distanz-optimale Tastenbelegung für das <i>3x10</i> -Layout bei vereinfachter Schreibweise im horizontalen Fall.	99
31	Distanz-optimale Tastenbelegung für das <i>4x8</i> -Layout bei vereinfachter Schreibweise im horizontalen Fall.	100
32	Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der semirealistischen Schreibweise in horizontaler Geräteposition. Dabei kehrt der linke Finger für das zweiseitige Bigramm "AJ" auf den (roten) Startpunkt zurück.	101
33	Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der realistischen Schreibweise in horizontaler Geräteposition. Zwischen der zweiten und dritten Bewegung wird der linke (grüne) Wartepunkt benutzt, da "A" und "J" auf verschiedenen Seiten liegen.	103
34	Distanz-optimierte Tastenbelegung für das <i>QWERTY-like</i> -Layout bei realistischer Schreibweise im horizontalen Fall.	104
35	Distanz-optimierte Tastenbelegung für das <i>3x9</i> -Layout bei realistischer Schreibweise im horizontalen Fall.	104
36	Distanz-optimierte Tastenbelegung für das <i>3x10</i> -Layout bei realistischer Schreibweise im horizontalen Fall.	105
37	Distanz-optimierte Tastenbelegung für das <i>4x8</i> -Layout bei realistischer Schreibweise im horizontalen Fall.	105
38	Schematische Darstellung der grundlegenden Idee hinter der Modellierung des Hybrid-Falls. Durch eine Kombination der Zielfunktionsvektoren des vertikalen (magenta) und horizontalen (blau) Problems erhält man einen neuen Zielfunktionsvektor (grün), der für die Suche nach einer Hybrid-Tastenbelegung geeignet ist. Die farbigen Punkte sind dabei die Lösungen der jeweiligen Probleme. . . .	108
39	Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der realistischen Schreibweisen in vertikaler und horizontaler Geräteposition im Verhältnis $\alpha : \beta$	109

40	Distanz-optimierte Tastenbelegung für das <i>QWERTY-like</i> -Layout bei realistischer Schreibweise im Hybrid-Fall.	110
41	Distanz-optimierte Tastenbelegung für das <i>3x9</i> -Layout bei realistischer Schreibweise im Hybrid-Fall.	110
42	Distanz-optimierte Tastenbelegung für das <i>3x10</i> -Layout bei realistischer Schreibweise im Hybrid-Fall.	110
43	Distanz-optimierte Tastenbelegung für das <i>4x8</i> -Layout bei realistischer Schreibweise im Hybrid-Fall.	111
44	Zeit-optimierte Tastenbelegung für das <i>QWERTY-like</i> -Layout bei realistischer Schreibweise im vertikalen Fall.	116
45	Zeit-optimierte Tastenbelegung für das <i>3x9</i> -Layout bei realistischer Schreibweise im vertikalen Fall.	117
46	Zeit-optimierte Tastenbelegung für das <i>3x10</i> -Layout bei realistischer Schreibweise im vertikalen Fall.	117
47	Zeit-optimierte Tastenbelegung für das <i>4x8</i> -Layout bei realistischer Schreibweise im vertikalen Fall.	117
48	Schematische Darstellung der zurückzulegenden Wege und ihrer Reihenfolge für die Übertragung des Textes "AJRN" bei Verwendung der Hover-over-Schreibweise in horizontaler Geräteposition. Dabei erfolgen die Bewegung 1.a und 1.b zur gleichen Zeit, jedoch muss der rechte Daumen mit der Eingabe von "J" warten, bis der linke Daumen "A" getippt hat. Im Anschluss daran folgen (gleichzeitig) die Bewegungen 2.a und 2.b.	119
49	Zeit-optimierte Tastenbelegung für das <i>QWERTY-like</i> -Layout und die Hover-over-Schreibweise im horizontalen Fall.	120
50	Zeit-optimierte Tastenbelegung für das <i>3x9</i> -Layout und die Hover-over-Schreibweise im horizontalen Fall.	121
51	Zeit-optimierte Tastenbelegung für das <i>3x10</i> -Layout und die Hover-over-Schreibweise im horizontalen Fall.	121
52	Zeit-optimierte Tastenbelegung für das <i>4x8</i> -Layout und die Hover-over-Schreibweise im horizontalen Fall.	121
53	Zeit-optimierte Tastenbelegung für das <i>QWERTY-like</i> -Layout und die Hover-over-Schreibweise im Hybrid-Fall.	123
54	Zeit-optimierte Tastenbelegung für das <i>3x9</i> -Layout und die Hover-over-Schreibweise im Hybrid-Fall.	123
55	Zeit-optimierte Tastenbelegung für das <i>3x10</i> -Layout und die Hover-over-Schreibweise im Hybrid-Fall.	123

56	Zeit-optimierte Tastenbelegung für das <i>4x8</i> -Layout und die Hover-over-Schreibweise im Hybrid-Fall.	124
57	Standard- <i>QWERTY</i> -Tastatur (siehe bspw. [Lewis et al., 1999b])	136
58	Alphabetische Belegung des <i>QWERTY-like</i> -Layouts	136
59	[Li et al., 2006]-Tastenbelegung für das <i>QWERTY-like</i> -Layout	137
60	[Leshner u. Moulton, 2000]-Tastenbelegung für das <i>QWERTY-like</i> -Layout	137
61	[Light u. Anderson, 1993]-Tastenbelegung für das <i>QWERTY-like</i> -Layout	137
62	<i>DVORAK</i> -Tastatur (siehe bspw. [Lewis et al., 1999b])	138
63	<i>KALQ</i> -Tastatur aus [Oulasvirta et al., 2013]	138
64	<i>OPTI I</i> -Tastatur (siehe bspw. [MacKenzie u. Soukoreff, 2002b])	139
65	<i>FITALY</i> -Tastatur (siehe [Ichbiah, 1996])	139
66	<i>OPTI II</i> -Tastatur (siehe bspw. [MacKenzie u. Soukoreff, 2002b])	139

Tabellenverzeichnis

1	Klassifikation ausgewählter Literatur aus dem Bereich der Tastaturentwicklung hinsichtlich verwendeter Lösungsmethoden und Anzahl der am Schreibvorgang beteiligten Finger.	17
2	Schematischer Überblick über die Ansätze zur Verbesserung der zeitlichen Lösbarkeit von Problem (P4) und welche Nebenbedingungen hinzugefügt (\oplus) oder entfernt (\ominus) werden.	74
3	Phase 1 - Evaluation der erarbeiteten Ansätze mit unterdrückten Solverstrategien.	75
4	Phase 2 - Evaluation der erarbeiteten Ansätze mit ausgewählten Solverstrategien.	76
5	Phase 3 - Evaluation von Kombinationen ausgewählter Ansätze.	78
6	Vergleich der Methoden zur Reduktion der Einträge der Wahrscheinlichkeitsmatrix.	87
7	Eigenschaften und Ergebnisse des Problems (P12) für abnehmendes \tilde{p} am Beispiel des <i>QWERTY-like</i> -Layouts.	89
8	Ergebnisse der Distanzminimierung für vertikale Anwendungen.	96
9	Ergebnisse der Distanzminimierung für horizontale Anwendungen.	106
10	Zurückzulegende Distanz (in LE) und Mehraufwand für die Texteingabe mit optimierten Tastenbelegungen in "richtiger" und "falscher" Geräteposition.	107
11	Ergebnisse der Distanzminimierung für Hybrid-Anwendungen.	111
12	Zurückzulegende Gesamtdistanz der hybrid-optimierten Tastenbelegungen in verschiedenen Gerätepositionen, sowie zusätzlicher Aufwand gegenüber den Optima der ursprünglichen Positionen.	112
13	Ergebnisse der Zeitminimierung für unterschiedliche Gerätepositionen bei Verwendung der realistischen Schreibweisen.	118
14	Ergebnisse der Zeitminimierung für die Hover-over-Schreibweise im horizontalen Fall.	122
15	Ergebnisse der Zeitminimierung für die Hover-over-Schreibweise im Hybrid-Fall.	124
16	Leistungsfähigkeit der in der Arbeit errechneten <i>distanzopt.</i> (*) oder <i>zeitopt.</i> (**) Tastenbelegungen in vertikaler Geräteposition.	140

17	Leistungsfähigkeit der Tastaturen und Tastenbelegungen aus verwandter Literatur bei vertikaler Geräteposition und realistischer Schreibweise.	140
18	Leistungsfähigkeit der in der Arbeit errechneten distanzopt. (*) oder zeitopt. (**) Tastenbelegungen in horizontaler Geräteposition.	142
19	Leistungsfähigkeit der Tastaturen und Tastenbelegungen aus verwandter Literatur in horizontaler Geräteposition.	142
20	Leistungsfähigkeit der in der Arbeit errechneten distanzopt. (*) oder zeitopt. (**) Tastenbelegungen für Hybrid-Anwendungen.	144
21	Leistungsfähigkeit der Tastaturen und Tastenbelegungen aus verwandter Literatur für Hybrid-Anwendungen.	145
22	Betragsmäßig größte, relative Verbesserung der Leistungsfähigkeit der Standard- <i>QWERTY</i> -Tastatur hinsichtlich verschiedener Kriterien durch eine in der vorliegenden Arbeit errechnete (jeweils in Klammern genannte) Tastenbelegung. . . .	146
23	Betragsmäßig größte, relative Verbesserung der Leistungsfähigkeit der Standard- <i>QWERTY</i> -Tastatur hinsichtlich verschiedener Kriterien durch eine in der vorliegenden Arbeit errechnete <i>QWERTY-like</i> -Tastenbelegung.	147
24	Buchstabenwahrscheinlichkeiten p_i	157
25	Bigrammwahrscheinlichkeiten p_{ik}	157
26	Tastenbelegungen für das <i>QWERTY-like</i> -Layout	160
27	Distanzen d_j zum vertikalen Startpunkt (5, 4) für das <i>QWERTY-like</i> -Layout . . .	160
28	Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das <i>QWERTY-like</i> -Layout	160
29	Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das <i>QWERTY-like</i> -Layout	160
30	Distanzen d_{jl} zwischen den Tasten für das vertikale <i>QWERTY-like</i> -Layout . . .	161
31	Distanzen d_j zum vertikalen Startpunkt (5, 4) für das <i>3x9</i> -Layout	164
32	Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das <i>3x9</i> -Layout	164
33	Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das <i>3x9</i> -Layout	164
34	Distanzen d_{jl} zwischen den Tasten für das vertikale <i>3x9</i> -Layout	164
35	Distanzen d_j zum vertikalen Startpunkt (5, 4) für das <i>3x10</i> -Layout	168
36	Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das <i>3x10</i> -Layout	168
37	Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das <i>3x10</i> -Layout	168
38	Distanzen d_{jl} zwischen den Tasten für das vertikale <i>3x10</i> -Layout	168

39	Tastenbelegungen für das 4x8-Layout	173
40	Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 4x8-Layout	173
41	Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 4x8-Layout	173
42	Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 4x8-Layout	173
43	Distanzen d_{jl} zwischen den Tasten für das vertikale 4x8-Layout	173
44	Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 3x3-Layout	179
45	Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 3x3-Layout	179
46	Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 3x3-Layout	179
47	Distanzen d_{jl} zwischen den Tasten für das vertikale 3x3-Layout	179
48	Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 2x4-Layout	180
49	Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 2x4-Layout	180
50	Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 2x4-Layout	180
51	Distanzen d_{jl} zwischen den Tasten für das vertikale 2x4-Layout	180
52	Tastenbelegungen für das 5x6-Layout	181
53	Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 5x6-Layout	181
54	Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 5x6-Layout	181
55	Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 5x6-Layout	181
56	Distanzen d_{jl} zwischen den Tasten für das vertikale 5x6-Layout	182
57	Tastenbelegungen für das 5x7-Layout	186
58	Distanzen d_j zum vertikalen Startpunkt (5, 4) für das 5x7-Layout	186
59	Distanzen \tilde{d}_j zum jeweils zugeordneten Startpunkt ($SP(L) = (0, 1)$ oder $SP(R) = (10, 1)$) für das 5x7-Layout	186
60	Distanzen \hat{d}_j zum zugeordneten Wartepunkt ($WP(L) = (2.5, 2)$ oder $WP(R) = (7.5, 2)$) für das 5x7-Layout	186
61	Distanzen d_{jl} zwischen den Tasten für das vertikale 5x7-Layout	187