котором значение критерия процесса конверсия) растёт, а ограничения (показатели состава) не выходят за пределы допустимых.

5. Приведены результаты исследований на установке на микрореакторной системе Qmix, когда в качестве исходных компонентов использовались хлорбензогидрол и бромбензогидрол. Результаты исследований показали эффективность разработанных алгоритмов моделирования и оптимизации плана исследований и позволили установить, что конверсия выше в реакции, где участвует бромдифенил метан. И помимо этого, в полученной реакционной массе не остаётся примесей.

1. Энде Д. Дж. Производство лекарственных средств. Химическая технология от R&D производства: пер. с англ. СПб: ЦОП «Профессия», 2015 1280с.
2. Корнюшко В.Ф., Панов А.В., Богунова И.В., Флид А.А., Применение системного подхода для построения информационного пространства вокруг фармацевтической разработки технологии производства готовых лекарственных средств // Прикладная информатика, Том 13. № 3(75). 2018 Май-Июнь, С.83-100.
3. Майофис Л.С. Технология химико-фармацевтических препаратов — Л.: МЕДГИЗ, 1968. — 539 с.
4. Бурляева Е.В., Колыбанов К.Ю., Панова С.А. Информационная поддержка систем принятия решений на производственных предприятиях химического профиля (под редакцией В.Ф.Корнюшко). М.: ИПЦ МИТХТ им. М.В. Ломоносова, 2013.196
5. Cook R., Jeeves T.A., Direct Search Solution of Numerical and Statistical Problems, J. Assoc. Comp. Mach., 8, 212 (1981).
6. Пятигорская Н.В., Ишмухаметов А.А., Беляев В.В. и др. Под редакцией акад. РАН В.В. Береговых. - М.: "ООО Группа РЕМЕДИУМ", 2019-328 С
7. Нив. Р. Генри пространство Деминга. Принципы построения устойчивого бизнеса. - М.: Альпина, Бизнес, 2005.
8. Фармацевтическая разработка: концепция и практические рекомендации. Научно-практическое руководство для фармацевтической отрасли/ Под ред. Быковского С.Н., проф., д.х.н. Василенко И.А., проф., д.фарм.н. Деминой Н.Б., к.фарм.н. Шохина И.Е., к.х.н. Нивожилова О.В., Мешковского А.П., Спицкого О.Р. – М. Изд-во Перо, 2015. – 472 с.: ил.
9. Adamo A., Rachel L., Beingessner, Behnam M., Chen J., Timothy F. Jamison, Klavs F., Jensen, Jean - Christophe M. Monbaliu, Allan S./ Myerson, Eve M. Revalor, David R. Snead, Torsten Stelzer, Nopphon Weeranoppanant, Shin Yee Wong, Ping Zhang. On-demand continuous-flow production of pharmaceuticals in a compact, reconfigurable system: science 352, 61 (2016), DOI: 10.1126/science. aaf 1337.
10. Таптунов В.Н., Интеллектуальная система информационной поддержки выбора технологических схем производства твердых лекарственных препаратов отрасли – Автореферат, канд.тех.наук. М.: 2012.
11. Balogh A., Domokos A., Farkas B., Farkas A., Rapi Z., Kiss D., Nyiri Z., Eke Z., Szarka G., Örkényi R., Mátravölgyi B., Faigl F., Marosi G., Kristóf Nagy Z. Continuous end-to-end production of solid drug dosage forms: Coupling flow synthesis and formulation by electrospinning: Chemical Engineering Journal 350 (2018) 290-299.
12. Plumb K. Continuous processing in the pharmaceutical industry: changing the mindset. Chem. Eng. Res. Des., 2005, 83(A6), p. 730–738.

Лясковский М.А., Горячкин Б.С., Лещев А.О., Мельников К.И.
Разработка метода оптимизации раскладки клавиатуры «ЙЦУКЕН»

МГТУ им. Н.Э. Баумана
(Россия, Москва)

doi: 10.18411/lj-06-2020-84

idsp: ljjournal-06-2020-84

Аннотация

В статье рассмотрен метод оптимизации стандартной раскладки «ЙЦУКЕН» путём минимизации пройденного пальцами расстояния. Предлагается использовать

итеративный алгоритм создания производных раскладок, для каждой из которых будет вычисляться ряд параметров, по которым алгоритм будет минимизировать пройденное пальцами расстояние и разность относительно стандартной раскладки «ЙЦУКЕН», для упрощения последующего перехода на новую раскладку также введены дополнительные меры и ограничения алгоритма в виде жёсткого закрепления клавиш за правой и левой рукой. Результаты показали, что алгоритм успешно минимизирует пройденное пальцами расстояние, а производная раскладка в несколько раз превосходит по оптимальности изначальную. Проведен и наглядно представлен сравнительный анализ всех параметров производной раскладки с изначальной раскладкой «ЙЦУКЕН» и более оптимальной «Диктор».

Ключевые слова: ЙЦУКЕН, Диктор, сравнение, оптимизация, раскладка, клавиатура, пройденное расстояние.

Abstract

The article discusses the optimization method of the standard layout of "JCUKEN" by minimizing the distance traveled by the fingers. It is proposed to use an iterative algorithm for creating derivative layouts, for each of which a series of parameters will be calculated, according to which the algorithm will minimize the distance traveled by the fingers and the difference with respect to the standard "JCUKEN" layout. To simplify the subsequent transition to the new layout, additional measures and limitations of the algorithm are introduced in the form hard keys for the right and left hand. The results showed that the algorithm successfully minimizes the distance traveled by the fingers, and the derivative layout is several times better than the original one by optimality. A comparative analysis of all the parameters of the derivative layout with the initial layout of "JCUKEN" and the more optimal "Diktor" was carried out and clearly presented.

Keywords.: JCUKEN, Diktor, comparison, optimization, layout, keyboard, distance traveled.

Введение

Автор и точная дата изобретения русской раскладки «ЙЦУКЕН» доподлинно неизвестны. Однако точно известно, что раскладку придумали в США в конце XIX века. Конструкция печатающего узла была достаточно отработанной, и важнейшие буквы расположили под указательными пальцами — в отличие от QWERTY. В то время раскладка официально называлась «стандарт-клавиатура». Буквы Ц и Э размещались среди цифр, а цифры 0, 1 и 3 вообще отсутствовали, так как считалось, что их можно было заменить буквами О, І и З. [1] Раскладка получила название «ЙЦУКЕН» по аналогии с «QWERTY» по шести первым буквам верхнего ряда (рис. 1). Непосредственно в России массовое производство собственных пишущих машинок было налажено только к 30-м годам прошлого века. [1]

Рис. 1. Раскладка «ЙЦУКЕН»

Анализ существующих раскладок клавиатур

Главная проблема раскладки «ЙЦУКЕН» заключается в том, что нагрузка по пальцам не равномерна. Одни пальцы пробегают по клавиатуре больше километров, чем другие, а это в целом не эффективно. Указательные пальцы при наборе трудятся значительно больше других. А мизинцу правой руки существенно не повезло — он работает чаще мизинца левой. Нагрузка должна быть распределена равномерно. Кроме того, желательно, чтобы больше в наборе участвовал основной ряд — тогда не надо было бы так часто гнуть пальцы, что не только сказалось бы при обучении слепому методу, но и на эффективность профессионального набора. [2]

С учётом всех пожеланий была разработана раскладка «Диктор», которая имела ряд преимуществ по сравнению с «ЙЦУКЕН» — самые «популярные» буквы русского алфавита уместились на основном ряду клавиатуры (рис. 2).

Рис. 2. Раскладка «Диктор»

Кроме того, если в этой раскладке напечатать слово «миг» или «грот», то легко заметить, что при наборе уже участвует не один палец.

С одной стороны, нагрузка стала более равномерной, но с другой, данная раскладка почти полностью отличается от популярной на данный момент «ЙЦУКЕН», что делает переход на неё затруднительным, поскольку людям, которые научились методу набора вслепую на «ЙЦУКЕН», будет очень сложно побороть рефлексорную память. Однако, на этом проблемы не заканчиваются.

Сейчас практически у всех клавиатур ряды смещены относительно друг друга и многие считают, что современные стандартные клавиатуры построены по принципам эргономики, и ничего лучше придумать уже нельзя, а если и можно, то уже поздно, существующие клавиатуры уже стали стандартом и никто не собирается переучиваться. Чтобы понять, почему современные клавиатуры имеют плохое расположение клавиш, и что можно с этим сделать, рассмотрим недостатки современных клавиатур.

Исторически так сложилось, что у стандартной клавиатуры ряды смещены определённым образом. Данное смещение появилось не случайно и никоим образом не относится к эргономичности расположения клавиш. Всё дело в том, что раньше в пишущих машинках требовалось выводить между рядами ударный механизм, и поэтому сами ряды сдвинули так, чтобы они не пересекались по вертикали друг с другом (рис. 3) [3].

Рис. 3. Расположение ударных механизмов на печатной машинке

При этом до сих пор делают клавиатуры с абсолютно таким же смещением рядов, хотя технических проблем такого рода уже давно не существует, зато такое расположение порождает проблемы другого рода. Например, при печати слепым десятипальцевым методом это начинает вызывать проблемы с движением пальцев. На рис. 4 показаны траектории движения пальцев левой и правой руки со стартовой позиции рук, для примера взята обычная среднестатистическая клавиатура.

Рис. 4. Обычная клавиатура

Мы видим, что движение пальцев у обеих рук не симметричны, что противоречит логике, потому что у правой руки получается аналогичное движение пальцев. Всё это ведёт к неестественным движениям пальцев. Неясно, почему данный вид расположения клавиш всё ещё является самым распространённым, при том, что эту проблему легко исправить и для этого не придётся учить новую раскладку, достаточно просто выровнять ряды относительно друг друга, что и сделали в ортолинейных клавиатурах (рис. 5).

Рис. 5. Ортолинейная клавиатура

Как видно из рис. 5, движения пальцев на ортолинейной клавиатуре симметричны и любые соседние клавиши находятся на равноудаленном расстоянии друг от друга.

Исходя из вышеописанного, было решено взять за начальную модель ортолинейную клавиатуру с раскладкой «ЙЦУКЕН» и путём итераций улучшить раскладку путём перестановки клавиш местами и изменения их положения.

Алгоритмы создания эргономичной раскладки

На первом этапе была построена модель, где расположение клавиш соответствовало ортолинейной клавиатуре (рис. 6), а их размер соответствовал стандартному размеру клавиатурных клавиш (18 мм) [4].

Рис. 6. Клавиши «ЙЦУКЕН» на координатной плоскости

Перед началом формализации необходимо определиться с набором данных, который будет использоваться, так как, в первую очередь, именно от его анализа следует отталкиваться.

В данном исследовании использовался набор данных (далее датасет) Яндекс Толоки [5] Toloka Persona Chat Rus. Он состоит из 10000 диалогов и предназначен для обучения диалоговых систем.

Сначала датасет прошел предобработку. А именно из данных были очищены побочные значения: цифры, английский текст и различные метаданные. Затем, оставшийся русский текст был разделен на сочетания символов, на основе которых был построен словарь мощностей каждого сочетания, где мощность – это количество употреблений сочетания в соответствующей ему зоне.

На основе этих данных была построена диаграмма пройденного расстояния между клавишами в зонах, изображенная на рис. 7.

Рис. 7. Пройденное расстояние на раскладке «ЙЦУКЕН»

Чем толще и краснее линия на диаграмме – тем больше пройденное расстояние между соответствующими двумя клавишами. Из диаграммы видно, что пальцы проходят неравномерное расстояние и соответственно нагрузка на них не сбалансирована. На диаграмме так же можно увидеть параметры, некоторые из которых будут выбраны в качестве критериев оптимизации для получения производных раскладок. Перечислим эти параметры и опишем их более подробно:

- Max – отрезок с максимальной пройденной дистанцией
- Sum – суммарное пройденное расстояние по всем отрезкам
- Avg – среднее пройденной расстояние по всем отрезкам
- Delta – коэффициент различия раскладки относительно исходной раскладки «ЙЦУКЕН»
- Score – сумма среднего пройденного расстояния и коэффициента различия

Коэффициент Score будет использоваться алгоритмом в функции минимизации при создании производных раскладок. Было решено выбрать этот параметр,

представляющий собой сумму среднего пройденного расстояния и коэффициента отличия от «ЙЦУКЕН» исходя из того, что мы поставили перед собой цель найти раскладку минимально отличающуюся от стандартной раскладки «ЙЦУКЕН» и в то же время сбалансированную, похожую по характеристикам на раскладку «Диктор» – именно этого нам позволяет добиться минимизация среднего пройденного расстояния. Также было введено ограничение при перестановке клавиш для упрощения потенциального перехода на новую раскладку с «ЙЦУКЕН». Для этого клавиши правой и левой руки жёстко закреплены за половиной клавиатуры, на которой они изначально находятся в исходной раскладке «ЙЦУКЕН». Это должно упростить процесс перехода на новую раскладку, поскольку не придётся заново вырабатывать рефлексную память принадлежности клавиш к той или иной руке, достаточно будет просто довольно незначительно изменить движение пальцев и их порядок.

Построим диаграмму пройденного расстояния для раскладки «Диктор» (рис. 8).

Рис. 8. Диаграмма пройденного расстояния на раскладке «ДИКТОР»

В раскладке «Диктор» нагрузка распределена более равномерно по сравнению со стандартной раскладкой и значение параметра Score значительно меньше, чем у раскладки «ЙЦУКЕН», что является хорошим показателем.

На основании вышеприведенных данных был разработан итеративный алгоритм производных раскладок, блок-схема которого изображена на рис. 9.

Рис. 9. Обобщённая схема алгоритма программы оптимизации

Алгоритм работает следующим образом: на вход подается исходная раскладка, которую требуется оптимизировать, затем, от этой раскладки случайным образом генерируются производные раскладки, путём ротации столбца или ряда, затем, все варианты оцениваются алгоритмом оптимизации, который выбирает из них раскладку с наименьшим параметром Score, обновляет константы и при необходимости сбрасывает раскладку или изменяет её, для избегания попадания в локальный минимум, затем эта раскладка снова подаётся на вход алгоритма. Если алгоритм оптимизации считает, что полученную производную раскладку не получится далее оптимизировать, он сохраняет её как оптимальную и завершает работу главного алгоритма.

Для создания производных раскладок используется генератор случайных чисел. Это было предусмотрено для упрощения процедуры перебора вариантов, поскольку число всех вариантов перестановок клавиш на ортолинейной клавиатуре равняется факториалу 36, что является довольно огромным числом и практически непосильной нагрузкой для современного поколения компьютеров. Из более подробной схемы алгоритма производных раскладок на рис. 10 видно, что сначала алгоритм случайным образом решает, какое действие он будет выполнять для создания набора производных раскладок: сдвиг ряда или сдвиг столбца.

Рис. 10. Детализация алгоритма производных раскладок

В отличие от сдвига столбцов, которые сдвигаются полностью – ряды сдвигаются частично, на правой и левой половине для упрощения перехода с раскладки «ЙЦУКЕН» на оптимальную, сгенерированную данным алгоритмом.

Сдвиг столбцов происходит следующим образом. Для примера возьмем 1 столбец из раскладки «ЙЦУКЕН» сверху-вниз и получим последовательность букв «ЙФЯ». Итогом работы алгоритма получится кортеж из следующих наборов букв: «ЙФЯ», «ЯЙФ», «ФЯЙ», затем, на основе этих 3-х наборов создаются 3 раскладки, в которых соответствующий столбец заменяется на сгенерированный набор букв. Аналогичным способом генерируются наборы и для рядов, где единственное отличие от столбцов в том, что наборы ротации символов будут создаваться отдельно для правой и левой половины клавиатуры.

После выполнения одной из вышеописанных операций сгенерированный набор производных раскладок передается в детализированный алгоритм оптимизации (рис. 11).

Рис. 11. Детализация алгоритма оптимизации

Задача алгоритма состоит в извлечении релевантных данных из раскладок и принятия решений на их основе для оптимизации выбранных нами параметров. На вход алгоритма приходит набор раскладок, для каждой из которых считаются параметры, отвечающие за сумму, среднее и максимальное пройденное расстояние, коэффициент отличия от исходной раскладки «ЙЦУКЕН» и целевой параметр Score, который представляет собой сумму среднего пройденного расстояния и коэффициента отличия от исходной раскладки. После этого алгоритмом выбирается раскладка, где параметр Score является наименьшим, затем это значение сохраняется. Сохранённое минимальное значение потом используется для его сравнения с параметром Score из последующих итераций и на основе этого сравнения алгоритм может принять 4 решения, в зависимости от последовательного количества раз, когда целевой параметр оказался больше или равен минимальному значению. Первое решение принимается, если алгоритм успешно минимизирует целевой параметр и ещё не достиг локального минимума и в этом случае просто выбирается оптимальная раскладка, способом, описанным выше, и подается на вход алгоритма производных раскладок. Второе решение принимается, когда алгоритм попадает в локальный минимум, в этом случае выбирается случайная раскладка из предыдущих итераций, таким образом мы «откатываем» алгоритм назад и даём ему возможность пойти другим путём. Третье решение принимается, когда в течение длительного времени алгоритму не получается подобрать раскладку с целевым параметром Score, близким к минимальному, который был получен на ранних итерациях. В таком случае берётся случайная раскладка, где Score был близок к минимальному значению, это позволяет «откатить» алгоритм к моменту, когда параметр Score был близок к минимальному значению и уменьшить отклонение. Последнее четвёртое решение принимается, когда в ходе выполнения алгоритма и описанных выше решений алгоритму не удаётся подобрать раскладку с целевым значением Score, меньше его полученного ранее минимального значения в течение длительного времени. В этом случае полученная раскладка помечается алгоритмом как оптимальная и программа завершает работу.

Алгоритм является очень гибким в настройке и при желании можно иначе настроить параметры и задать иную логику избегания локальных минимумов. Однако для настоящего исследования этого оказалось достаточно, что видно из диаграммы на рис. 12, где суммарное пройденное расстояние гораздо меньше, чем у исходной раскладки «ЙЦУКЕН».

Рис. 12. Оптимизированная производная раскладка от «ЙЦУКЕН»

На рис. 13 представлено изменение целевого параметра Score по мере поиска оптимальной раскладки. Из графика видно, что алгоритм успешно справился с оптимизацией раскладки и достиг минимума уже на 14 итерации.

Рис. 13. Динамика целевого параметра Score до первого минимума

Так как в алгоритме была предусмотрена защита от попадания в локальный минимум, после 24 итерации им предпринимались попытки возврата к предыдущим вариантам раскладки и выполнения действий в другом порядке, чтобы добиться других результатов.

Параметр Score флуктуирует и каждый раз приходит к тому же минимальному значению, спустя несколько итераций (рис. 14). Исполнение алгоритма автоматически завершилось спустя 320 итераций, когда выполнилось последнее условие оптимизации, а целевое значение не уменьшилось после всех предпринятых попыток.

Рис. 14. Динамика целевого параметра Score при поиске глобального минимума

Сравнительный анализ исходных и производной раскладок

Сравним нормированные параметры раскладок «ЙЦУКЕН» и «Диктор» с полученной производной от «ЙЦУКЕН» раскладкой. Данные приведены в табл. 1, где большие значения олицетворяют лучший результат.

Таблица 1.

Нормированные значения параметров раскладок

Параметры	«ЙЦУКЕН»	«Диктор»	Производная от «ЙЦУКЕН»
Max	0,14	1,00	0,80
Sum	0,18	0,92	1,00
Avg	0,18	0,91	1,00
Delta	–	0,23	1,00
Score	0,31	0,41	1,00

Из таблицы выше видно, что алгоритм подобрал раскладку, которая даже превосходит ожидания и опережает раскладку «Диктор» по большинству параметров, из которых особенно выделяются параметры Avg и Delta.

Сравним пройденные расстояния у трёх раскладок:

Таблица 2.

Сравнение пройденной дистанции

Пальцы	Пройденное расстояние (км)		
	«ЙЦУКЕН»	«Диктор»	Производная от «ЙЦУКЕН»
Левый мизинец	0,12 (0,38%)	0,05 (0,74%)	0,59 (10,22%)
Левый безымянный	0,10 (0,31%)	0,20 (3,12%)	0,55 (9,37%)
Левый средний	1,86 (5,73%)	0,62 (9,74%)	1,66 (28,53%)
Левый указательный	13,95 (43,04%)	1,15 (18,23%)	0,63 (10,88%)
Правый указательный	15,64 (48,26%)	2,84 (44,81%)	1,31 (22,43%)
Правый средний	0,56 (1,71%)	0,68 (10,69%)	0,53 (9,11%)
Правый безымянный	0,11 (0,35%)	0,48 (7,50%)	0,12 (2,00%)
Правый мизинец	0,07 (0,22%)	0,33 (5,18%)	0,43 (7,74%)
СУММА	32,41	6,33	5,82

Не трудно заметить, что на полученной алгоритмом раскладке пройденное расстояние меньше на 26,59 км, что является довольно существенным результатом и уменьшением пройденного пути. Также можно увидеть, что на полученной производной раскладке пройденное расстояние более равномерно распределено по всем пальцам, в отличие от раскладки «ЙЦУКЕН». Более того у полученной нами раскладки пройденное расстояние даже меньше, чем у раскладки «Диктор» на 0,51 км, что довольно неплохо, учитывая тот факт, что раскладка «Диктор» является одной из самых оптимальных на текущий день раскладок для печати.

Представим результаты в виде гистограммы и сравним полученную раскладку с двумя другими (рис. 15).

Рис. 15. Сравнение равномерности распределения пройденного расстояния у «ЙЦУКЕН» и производной раскладки

Пройденное расстояние производной раскладки более равномерно распределено по всем пальцам - на левый и правый указательные пальцы уже не приходится более 91% нагрузки, если судить по полученным данным. Однако, следует отметить, что мизинцы практически никак не задействованы при печати, на них и безымянные пальцы приходится менее 1% от всей нагрузки. У производной раскладки самая минимальная нагрузка составляет 2%, в среднем около 10% на остальных пальцах и не превышает 28%.

Анализируя аналогичную сравнительную гистограмму для раскладки «Диктор» (рис. 16), можно отметить, что уже не присутствует столь разительной разницы и по нагрузке эти две раскладки уже практически сопоставимы.

Рис. 16. Сравнение равномерности распределения пройденного расстояния у раскладки «Диктор» и производной от «ЙЦУКЕН» раскладки

Заключение

Разработанный метод оптимизации и реализующие его алгоритмы дали достаточно хорошие результаты по рассмотренным параметрам, а именно по пройденному пути. Производную раскладку смело можно назвать оптимизированной и использовать на практике при наборе текста. Однако следует отметить, что пройденное расстояние хоть и распределено более равномерно, чем у раскладки «ЙЦУКЕН», всё же не имеет видимые перепады от пальца к пальцу, а идеальным вариантом было бы подобрать раскладку, у которой на выбранном тексте пройденные расстояния были бы равны на всех пальцах, хотя не факт, что такая раскладка существует и будет работать одинаково хорошо для всех текстов.

1. Менделеев И.П. Новое в области пишущих машин // Природа и Люди, No. 49, Май 2000.
2. ЙЦУКЕН [Электронный ресурс] // Википедия. URL: <https://ru.wikipedia.org/wiki/ЙЦУКЕН>.
3. Карпов П. Альтернативные раскладки [Электронный ресурс] // ErgoCоло. 2007. URL: https://ergosolo.ru/reviews/history/alternative_layouts/
4. Лебедев А. Трагедия запятой [Электронный источник] // artlebedev. 2004. URL: <https://www.artlebedev.ru/kovodstvo/sections/105>.
5. Клавиатура ввода данных. Общие технические требования // ГОСТ 29124-91, Октябрь 1992.
6. Открытые датасеты Яндекс Толоки [Электронный ресурс] // toloka. URL: <https://toloka.yandex.ru/datasets/>