

経路依存性を考慮した QWERTY 配列の改善に関する研究

高木遼太^{†1} 藤井慶^{†1}

概要: 本研究では、キーボードのキー配列を組み合わせ最適化問題と捉え、ユーザの QWERTY 配列への強い経路依存性を考慮し、ユーザにいきなり新配列を練習させるのではなく、キー配列最適化の反復と共に、QWERTY 配列からより良いキー配列へ徐々に移行させるアプローチを検討している。本報告では最適化における目的関数の精度向上を試みた、3つの提案法について報告する。

キーワード: QWERTY, 経路依存性, 組み合わせ最適化問題, 貪欲法

An improvement of QWERTY layout keyboard considering path dependence

RYOTA TAKAKI^{†1} KEI FUJII^{†2}

Abstract: Today, many people use QWERTY layout keyboard. Newer and better layouts have been proposed, however they have not become widespread, because of strong path dependence. The authors propose the migration system of keyboard layout by introducing Greedy algorithm. In this reports, we describe 3 new objective functions and their behavior.

Keywords: QWERTY, Path dependence, Combinatorial optimization, Greedy algorithm

1. はじめに

今日、私たちが使用しているキーボードの多くは QWERTY 配列である[1]。しかしながら、このキー配列はタイプライターの時代から用いられているものであり、最善の配列でないことが知られている。QWERTY 配列の代替案として、Dvorak 配列や親指シフト配列、TRON 配列[2]など、より入力効率を改善したキー配列が考案されてきたが、そのどれもが一般には普及していない。その理由として経路依存性やロックイン状態になっていることが挙げられる。経路依存性とは「あらゆる状況下において、人間やその組織が取りうる判断は、その過去や歴史的経緯に制約を受ける」[3]という性質である。つまり、私たちの多くは QWERTY 配列に慣れてしまったが故に、他のキー配列に移行する際の負担が大きくなってしまい、最善ではないキー配列の使用を余儀なくされていると考えられる。

この問題に対するこれまでの取り組みとして、ユーザのキー入力回数に基づいた最適化手法を提案し、一応のキー配列移行システムを完成させた。そこで、本稿では、さらに3つの手法を提案し、既存システムの改良を目指す。

2. キー配列の最適化手法（従来法）

2.1 本研究における最適化の原理

本研究では、最適なキー配列を「タイピング時に手をホームポジションからできるだけ動かさずに済む配列」と定義している[4]。言い換えれば、各指のホームポジションか

らの移動量が最小となるキー配列である。これを踏まえた上で、最適化の手順と最適性の評価方法について説明する。

今 N 個のキーを持つキーボードの i 番目 ($1 \leq i \leq N$) のキーを k_i とし、 k_i が押された回数を c_i 、 k_i に最も近いホームポジション上の指と k_i の距離を d_i とすれば、 k_i の配置に関する評価値 $e_c(i)$ は次式で定義される。

$$e_c(i) = \frac{c_i}{\sum_{j=1}^N c_j} \times d_i \quad (1)$$

これより、キー配列の全体の評価値 E は次式で定義される。

$$E = \sum_{i=1}^N e_c(i) \quad (2)$$

この評価値 E が最小となるのが最適なキー配列である。つまり、最適なキー配列を見つけることは、目的関数 E が最小となるキーの並びを求める組み合わせ最適化問題を解くことと同義である。

そこで著者らはこの最小化問題に対して貪欲法を用いたアプローチを試みている。今、 i 番目と j 番目のキーの入れ替えを考える。 k_i と k_j の位置を入れ替えて(1)式を再計算した場合の、各々のキーの評価値を e'_i および e'_j とすれば、このキーのペア評価値 $e_{i,j}$ は次式で定義される。

$$e_{i,j} = (e'_i + e'_j) - (e_i + e_j) \quad (3)$$

貪欲法では、 $e_{i,j}$ を最小にするペアが目的関数 E を最小値にもっとも近づける一手として選択される。そしてこの基準に基づきキーを繰り返し入れ替えていくことで、 E を最小

^{†1} 熊本高等専門学校
National Institute of Technology, Kumamoto College

に近づけていく。

貪欲法は反復の初期段階において評価値を大きく減少させる逐次的な最適化手法である。キー配列最適化の問題には経路依存性があるため、一気に最適化してしまうのではなく逐次的に最適解に近づくアプローチを採ることで、ユーザが少しずつ新しい配列に慣れていくことが期待される。また、ユーザによっては最適解に至るまでキー入れ替えを繰り返すことを望まず、数個の入れ替えのみを行う可能性がある。この点においても貪欲法は最初の数回で大きく評価値を下げるのが期待される。

2.2 手法の評価

前節に述べた手法の妥当性を検証するため、以下の手順で評価実験を行った。

2.2.1 キー入力データの収集

実験の前準備として、ユーザのキー入力統計データを収集するキーロガーを作成した。収集するデータは、各キーの「押された回数」「押された時間長（ミリ秒）」「ある別のキーが押された後に、そのキーが押された回数」の3つである。実験に使用するデータを表1にまとめる。

表1 収集したキー入力データの概要

収集期間[日]	34
キー入力の総回数[回]	399395
キー入力の総時間[秒]	941623

2.2.2 キー配列の入れ替え

Enter 実際にキーを20回入れ替える。なお、本研究では、キー再配置のやり易さを考慮して、図1の研究用ソフトウェアキーボードを用いる。また、時間の都合により、一回の入れ替えごとに、そのキー配列に慣れる手順は省略する。



図1 検証用ソフトウェアキーボード

2.2.3 実験結果

横軸にキーの入れ替え回数、縦軸に正規化した評価値Eを取ったグラフを図2に示す。図2より、最初の5回程度の入れ替えで評価値Eが大幅に減少していることが分かる。次に、キーを5回入れ替えた後のホームポジション付近のキー配列を図3に、20回入れ替えた後のキー配列を図4に

示す。各図より、Enter や Back Space など使用頻度の高いキーがホームポジション付近に移動していることが確認できる。

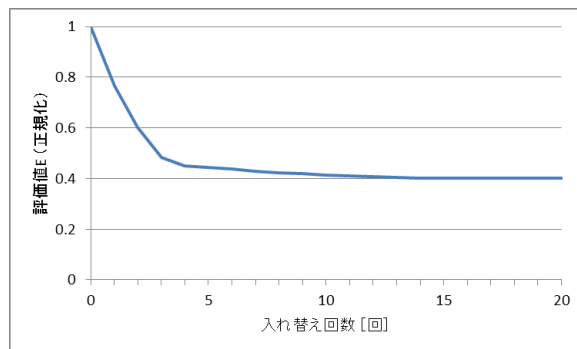


図2 実験結果(従来法)



図3 5回入れ替え後のキー配列(従来法)



図4 20回入れ替え後のキー配列(従来法)

3. 最適化の改善

2.2.3節の実験結果より、キー配置が最適化される傾向は確認できたと言える。しかしながら考慮する要素が不十分である可能性など、まだ改良の余地があると考えられる。そこで本項では評価式やアプローチ自体を見直し、更なる最適化を提案する。

3.1 押された時間長を考慮した最適化(提案法1)

3.1.1 最適化の原理

例えば、コピー&ペーストで多用されるCTRLキーなど、他と比べて「長押し」されやすいキーがある。しかし、前述の手法ではこのようなキーが押された時間長は評価対象外であった。そこで、従来法の(1)式を改造し、キーが押された時間長を考慮したキー配列の最適化手法を提案する。

k_i が押された時間長を t_i とする. このとき, k_i が押された時間長を基にしたキー単体の評価値 $e_t(i)$ は次式で定義される.

$$e_t(i) = \frac{t_i}{\sum_{j=1}^N t_j} \times d_i \quad (4)$$

3.1.2 評価実験

(2)式の $e_c(i)$ を $e_t(i)$ に置き換えた上で, 再び 2.2 節の実験を行った.

横軸にキーの入れ替え回数, 縦軸に正規化した評価値 E を取ったグラフを図 5 に示す. また, キーを 5 回入れ替えた後のホームポジション付近のキー配列を図 6 に, 20 回入れ替えた後のキー配列を図 7 に示す. 各図より, 手法 1 と手法 2 では, 入れ替え対象となったキーに多くの共通点が見られ, キーが押された回数と時間長には高い相関があると言える.

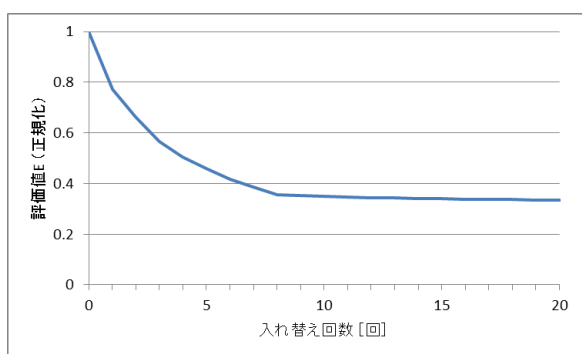


図 5 実験結果(提案法 1)



図 6 5 回入れ替え後のキー配列 (提案法 1)



図 7 20 回入れ替え後のキー配列 (提案法 1)

3.2 指の特性を考慮した最適化(提案法 2)

3.2.1 最適化の原理

手の構造上, 薬指や小指より人差し指や中指が動かしや

すい. しかし, このような指の特性が考慮されていない従来法では, 非常に使用頻度の高いキーにも関わらず小指付近に移動してしまうなど, 不適切な入れ替えが生じていた. そこで, 従来法の(1)式を改良し, 指の特性を考慮したキー配列の最適化手法を提案する.

ホームポジション上で k_i と最も近い指の重みを w_k とする. このとき, 指の特性を考慮したキー単体の評価値 $e_{fc}(i)$ は次式で定義される.

$$e_{fc}(i) = \frac{c_i}{\sum_{j=1}^N c_j} \times d_i \times w_k \quad (5)$$

次に, 各指の重みを表 2 に示す. 指の重み w_k は, 動かしやすく疲れにくい指ほど小さく, 動かしづらく疲れやすい指ほど大きくなる. また, 各指の重みは左右で同じとした.

表 2 指の重み

指	重み
親指	0.6
人差し指	0.7
中指	0.8
薬指	0.9
小指	1.0

3.2.2 評価実験

(2)式の $e_c(i)$ を $e_{fc}(i)$ に置き換えた上で, 再び 2.2 節の実験を行った.

横軸にキーの入れ替え回数, 縦軸に正規化した評価値 E を取ったグラフを図 8 に示す. また, キーを 5 回入れ替えた後のホームポジション付近のキー配列を図 9 に, 20 回入れ替えた後のキー配列を図 10 に示す. 各図より, Enter や Back Space など, 使用頻度の高いキーが優先してホームポジションの中心に集まっていることが分かる.

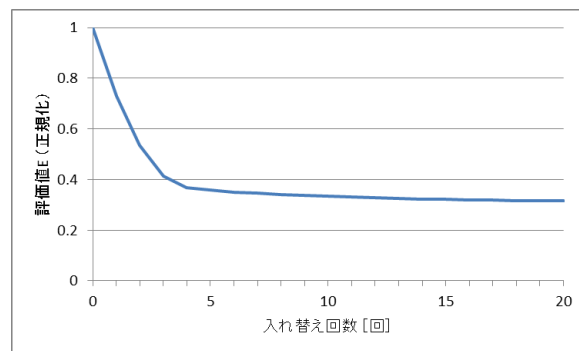


図 8 実験結果(提案法 2)



図 9 5 回入れ替え後のキー配列 (提案法 2)



図 10 20 回入れ替え後のキー配列 (提案法 2)

3.3 一つ前の入力を考慮した最適化 (提案法 3)

3.3.1 最適化の原理

子音と母音など、交互に入力されやすいキーの組は、左右別々の手に割り振られた方が好ましく、例えば Dvorak 配列はこのことを配慮している。しかし本手法ではこのような時系列的な特徴を考慮していない。そこで、従来法の(1)式を改造し、一つ前の入力を考慮したキー配列の最適化手法を提案する。

k_p の次に k_i が押された回数を $c_{p,i}$ とする。また $f(k_p, k_i)$ を「 $k_p \neq k_i$ かつ k_p と k_i の最寄りの手が等しい場合は 1 を、それ以外の場合は 0 を返す関数」とする。A および a を補正定数とする。このとき、一つ前のキー入力を考慮したキー単体の評価値 $e_p(i)$ は次式で定義される。ここで I_i は、統計量の少ないキーの評価値を抑えるための係数である。

$$e_p(i) = e_c(i) + \frac{A \sum_{p=1}^N c_{p,i} \cdot f(p, i)}{S_i} \cdot I_i \quad (6)$$

$$\text{ただし } I_i = \{1 - \exp(-a S_i)\} \quad (7)$$

$$S_i = \sum_{p=1}^N c_{p,i} \quad (8)$$

3.3.1 評価実験

(2)式の $e_c(i)$ を $e_p(i)$ に置き換えた上で、再び 2.2 節の実験を行った。なお、今回は(6), (7)式の補正係数をそれぞれ $A = 50$, $a = 1$ とした。

横軸にキーの入れ替え回数、縦軸に正規化した評価値 E を取ったグラフを図 11 に示す。また、キーを 5 回入れ替えた後のホームポジション付近のキー配列を図 12 に、20 回入れ替えた後のキー配列を図 13 に示す。図 12 ではまばら

であった母音が、図 13 では右側に移動していることが分かる。

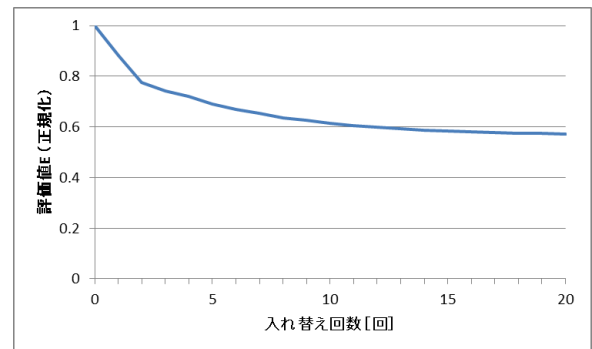


図 11 実験結果 (提案法 3)



図 12 5 回入れ替え後のキー配列 (提案法 3)



図 13 20 回入れ替え後のキー配列 (提案法 3)

4. 手法間の比較

各手法の実験結果に関して、横軸にキーの入れ替え回数、縦軸に正規化した評価値 E を取ったグラフを図 14 に示す。それぞれ評価式が異なるため、評価値自体の差は比較できない (縦軸の値の差での比較ができない)。そこでここでは評価値の下がり方で検証する。これは、本研究の想定するアプローチが経路依存性を考慮しており、ユーザが数回のキー入れ替えまでを行う可能性が高いと考えたためである。

図 14 より、押下時間長を使った提案法 1 に比べ、キーが押された回数に基づく提案法 2 の方が速く減少することが分かる。また、提案法 2 と提案法 3 とを比較すると、提案法 3 の減少速度の方が遅い。これは提案法 3 において「連続しやすいキーの組は左右に割り振る」という条件を設け

たことが原因と考えられる。すなわち、QWERTY 配列は手の交互打鍵を考慮していないため、交互打鍵を考慮する配列へ移行させるまでに多くのキー入れ替えを要してしまうということの表れではないかと考える。

これらの結果から、今回提案した3手法の中では提案法2が評価値減少速度の観点からもっとも良好と考えられる。

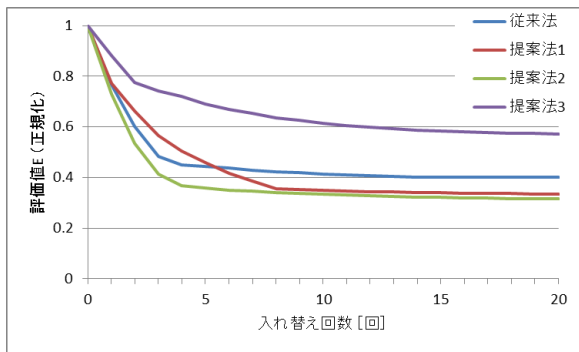


図 14 評価値Eの遷移

5. まとめ

本稿では、キーボードのキー配列の配置問題を組み合わせ最適化問題の一つと捉え、経路依存性を考慮しつつユーザが QWERTY 配列からより良いキー配列へ移行するシステムの実現を目指し、最適化の目的関数について3つの手法を提案した。そして各手法の評価値の減少速度の観点から、指の重みを距離計算に導入した手法2がもっとも良好との見解を示した。

今後の展望として、提案法2をさらに改良することが考えられる。例えば、本報告では指の重み w_k を直感的に決定したが、この値を計測実験などにより設定することで、より指の特性に伴ったキーの入れ替えを提案できるのではないかと期待する。また、一連の提案法は英字キーや修飾キーなどのキーの種類を考慮せずキー入れ替えを行うため、ユーザの心理的な抵抗感のあるキー入れ替えを提案することがあると考える。そこでキー入れ替えの際に、キーの種類に基づく制約を入れると抵抗感を抑えられるのではないかと期待する。

参考文献

- [1] 名和小太郎. QWERTY 配列. 情報処理, 2005, vol. 47, no. 12, pp. 839-841.
- [2] 坂村健, BTRON における入力方式 —TRON キーボードの設計—, 情報処理学会研究報告(HCI), 1986, Vol. 1986, No. 41 (1986-HI-007), pp.1-8.
- [3] “経路依存性とは ～ exBuzzwords 用語解説”. http://www.exbuzzwords.com/static/keyword_5369.html, (参照 2018-02-02).
- [4] 高木遼太, 藤井慶. 組み合わせ最適化手法の応用に関する研究 ～ 経路依存性を考慮したキーボード配列の最適化. 情報処理学会インタラクション 2017, 2017, 2-408-79, pp. 649-653.