

中指シフト仮名文字配列「花」

富樫雅文

北海道大学理学部

シフトキーを中指ホームポジションに設定した新しい鍵盤仮名文字配列を提案する。

最適化計算はふたつの文字位置の置換によって配列をくりかえし微小変化させて評価値を改善する最急降下法によった。配列の評価には平均打鍵時間と指負荷バランスの良さを同等の重みで使用する。平均打鍵時間はテキストを使用しない自由打鍵実験による2打鍵間時間データに頻度効果の補正を加えたものと、仮名2文字組出現頻度によった。また、指負荷バランスとしては自由打鍵実験における各指の使用頻度との差を用いた。収束判定の際には峠越えを試みて、近在極値の見のがしをふせいだ。

乱数により生成した1万個の初期配列のおのおのに対して最急降下法を適用し、終期配列群から最も性能の良いひとつを提案配列として採用した。最適化にはベクトル型計算機を使用し約5時間を要した。

Middle Finger Shift Kana Keyboard
Layout "HANA"

Masatomo TOGASI

Department of Chemistry,
Faculty of Science,
Hokkaido University,
Kita-ku Kita-10 Nishi-8,
SAPPORO 060, JAPAN

A new layout of kana keyboard is proposed. Shift keys are set on home position of middle fingers. Steepest descent method is applied for optimization of the layout, where step by step descent towards an optimum point is done by swapping two characters on a trial layout.

Evaluation of a layout is based on both average typing speed and finger load balance of which reference is taken from a free typing experiment. To avoid trapping by trivial local minimum, passing technique is used.

The optimization is done for 10000 trial random layouts. Finally, the best one is taken as the proposed layout "HANA".

The calculation took 5 CPU-hours on a super computer HITAC S820/80.

0. はじめに

Nickels【1】は計算機による英語鍵盤配列の最適化のために、指の器用さに関する経験的データから30鍵×30鍵の2打鍵間の時間を求め、これと英語における2文字組の出現頻度表とを用いて乱数により生成した50万種の配列の期待平均打鍵時間を計算した。この過程でDvorak配列はQwerty配列に比べて6.2%速いことが示された。さらに、50万種の中から29種の配列案を初期配列にとり、2文字の位置を交換しながら改善する最急降下法を適用して最適化配列を求めた。その結果、Qwertyに対し25%、Dvorakに対しても20%まで期待平均打鍵時間を短縮する配列を見出した。

しかし、この研究に対して山田【2】は、得られた配列が、(1)使用された各2打鍵の推定速度は上限値であって、実作業では低頻度の打鍵列はこれより遅く打たれるという、「頻度効果」を考慮していない。(2)両手の人差し指だけで合計65%もの打鍵頻度負荷を課していること、および、(3)リズムカルなタイピングによる「レベル効果」を無視している、という問題点を指摘している。

小西ら【3】はQwertyタイピストに対する等確率ランダム文字列の打鍵実験を行ない、2打鍵間の時間を測定し、2文字組の出現頻度表とあわせて、Qwerty、Dvorak各配列を初期配列とする最急降下法による最適化を試みた。しかし、使用した打鍵時間データでは、乱数により生成した約17万種のいかなる配列よりも(またDvorak配列よりも)Qwerty配列の方が優れているという結果を招いた。また得られた最適化配列案もQwerty配列に対してわずかに1.3%の改善に過ぎなかった。彼等はさらに、生の打鍵時間データを(手の左右や指などの)6個の打鍵要素を変数とする1次式に回帰させ、この式からあらためて生成した各打鍵時間をもとにした最適化を試み、Dvorak配列よりわずかに良い配列を見出した。このように、評価対象の配列と評価に用いた配列(被験者の日常使用している配列)が異なることには問題がある。

仮名鍵盤の文字配列としては、カナモジ会によるカナタイプライターの文字配列を基にして作られた旧JIS配列がある。50音順を変形して4段に配置したものであって最適というにはほど遠いが、現在最も多く使用されている仮名鍵盤配列である。最近になって、渡辺ら【4】はJISカナ配列の新しい規格のために3段のみを使用する配列の設計を行なった。そこでは、左右交互打鍵の同手連続打鍵に対する優位性や異指打鍵の同指連続打鍵に対する優位性など、打鍵実験から得られた定性的知見をもとに、左右

各段・各列への文字の振り分けを行ない、最後に数個の候補配列に対するシミュレーションによって最終案を決定した。

この方法の問題点は、一度定量的に得られた打鍵時間データをそのまま使用せず、定性的知見のレベルに縮約した上でこの知見にしたがって配列を作っている点にある。また、シフトキーは同時押下型ではなく前置型に変更してはいるものの、その位置は従来の鍵盤におけるシフトキーの位置とほぼ同じであって依然として左右の小指に対する過重な負担が軽減されていない。

この点、OASYS鍵盤やTRON鍵盤では、シフトキーは左右の親指の位置に移されており、空白キーのみを分担していた親指の活用と同時にシフト時の手の動きを減らすことに成功している。

一方、白鳥【5】は3段30鍵程度という触指打鍵のための条件と63個もの仮名文字との間の矛盾を避け、ローマ字入力(およびその変形)に特化した文字配列を提案している。しかし、それらではシフトキーはもはや必要ではないかわりに、仮名1文字当たりの所要ストローク数が増大する(濁音を1文字とする計算で1.5)という代償を支払わなければならない。ローマ字という規則的な構造をもつマルチストロークコードは学習初期における記憶負荷を軽減するというメリットはあるものの、それが、熟練後における総ストローク数の増大を補償しうるものであるかどうかは疑問である。

1. 問題点

仮名鍵盤配列のこれまでの研究における問題点は次のように整理することができる。

(1) シフトキー

鍵盤のキー数より文字数の方が多ければ、キーへの文字多重配置とシフト機構の導入は自然の成り行きであるが、初期のタイプライターにおいて最小限の改造で済ませたことの結果として、文字キー領域の外縁(左右の下隅)にシフトキーを配置したことが長く尾を曳くこととなった、現在に至るもおシフトキーを配列設計の副次的要素とする傾向が続いている。

親指によるシフトキー(や空白キー)押下は、他の4指の打鍵時の動作のような手の平内部へ向かう「引き付け」ではなく、手首に近い第3関節のみの屈曲により打鍵されている。言い換えれば親指の打鍵は手首近くで制御されている。このため親指打鍵は、しばしば、手首の回転(rolling)によって行なわれことになる。少なくとも標準的な鍵盤形状を前提とする限り、親指による打鍵は他の

4 指の打鍵と様相を異にしているのであって、これに文字打鍵のためのシフト操作の負担を負わせることには依然として問題がある。

また、シフトキーと文字キーの同時押下によるシフト操作は、シフトキーのメイクとブレイクの間に文字キーのメイクが入らなければならない、(Nキーロールオーバー鍵盤における) 通常の文字キー打鍵の連続のように、ブレイクのタイミングに対してルーズな制御がゆるされない。即ち、シフトキーのブレイクは文字キーのメイクと同格の順序制御の対象となる。

(2) 経験的方法と非経験的方法

従来の鍵盤文字配列の多くは、左右手や各指の強さや交互打鍵と同手打鍵、異指打鍵と同指打鍵などの優劣に関する定性的知見をもとに、配列を「組み立て」ていく方法であった。それらの定性的知見は、実際に行なわれる多様かつ大量の打鍵動作を簡潔に把握するための統計的縮約であり、打鍵動作全体のいくつかの断面を示しているに過ぎない。したがって、配列設計における定性的知見の運用の場面では、実証による裏づけのない判断が入り込む余地がある。個別の打鍵間時間のような、大量でより精度の高い情報の利用は、人手による試行錯誤的な配列設計では不可能であっても、高速計算機を使用した設計では、「縮約」ではない全体像により近い打鍵の様子に基づいた最適化ができる。

(3) 記憶負担について

ある配列が容易に受け容れられるための条件のひとつに、「覚えやすい」ということがあげられる。特に Qwerty 配列上のローマ字入力、英語ベースで鍵盤を使用してきた利用者にとっては、初期学習の負担がほとんどゼロであるためにその受け容れに何の抵抗もない。Qwerty 配列自体の学習に要した時間を繰り入れなければ、たとえば JIS カナ配列との、記憶負担の比較は本来不合理であるが、覚えやすさへの心理的傾斜そのものは事実であろう。

しかしながら、長期にわたる使用とその結果の熟練を想定するならば、初期学習に費やした負担に充分見合うだけの性能に達しうる配列を採用することに利があることは、目視打鍵法に対して触指打鍵法が示した優位性を例にとるまでもなく、明らかである。要は適切な学習法を確立することによって、配列自体に記憶負担への配慮を要求することは合理的ではない。

(4) 評価—特に指負担バランス

文字配列の評価において、期待平均打鍵速度とともに、各指にかかる作業負担の配分(指負担バランス)が論じら

れることが多い。強い指にはより多くの負荷がかかり小指のような弱い指には負荷が小さい配列が一般に良い配列とされている。

しかし、一体、何%の負荷がどの指にかかるべきであるのかについての定量的な基準は確立されていない。このために指負担バランスの評価はいきおい主観的判断もしくは定性的判断や、過去の配列との比較に終ることが多い。

(5) 頻度効果

Nickels の研究に対して山田が指摘したように、配列設計に文字使用頻度による打鍵速度の変化(頻度効果)を取り入れることは重要であり、Salthouse【6】による断片的なデータも頻度効果の存在を示唆している。しかし、その定量的なすがたは確認されていない。

(6) 打鍵特性データの配列依存性

打鍵特性を知るために、「何かのテキストを与えて打たせる」という形式を踏む限り、それは実験に使用した文字配列と対象テキストに対して相対的な情報しか与えない。(また、当然にも、それは鍵盤の物理的構造や手指の解剖学的構造に対しても相対的であるが、これらの要素は大きく変化しないものと仮定することができる。)

このため、これまでの文字配列の設計にあたっては、配列依存性が薄れていると思われる程度まで後退した定性的知見レベルの特性値を用いるか、あるいは、特性データに何らかの補正を施したうえで使用する等の方策がとられてきた。しかし、配列に対して中立的でありながら精度の高い打鍵特性情報を得られなかったことは、これまでの配列設計にとってのひとつの弱点であった。

以上のような問題点の認識を踏まえたうえで、本研究では、

- (1) これまでのシフト方式を再考することから出発し、
- (2) 特定の配列に依拠しない打鍵特性データを求め、
- (3) 頻度効果と指負担バランスに対する客観的かつ定量的な取り込みを行ない、
- (4) 大規模な最適化計算を遂行した。

2. 準備

2.1 配列の前提

配列する文字は JIS で定める 63 文字とし、濁点や半濁点も独立した文字として扱う。

鍵盤は標準鍵盤を仮定する。このうち、仮名配列を行なうキーは 3 段 33 キーとする。左右の人差し指と右手の小指はおのおの 2 列を受け持つ。4 段にわたる領域を使用し

ないのは、主として指の移動距離を考慮した結果であって、段数そのものの問題ではない。日本人の手にとって標準鍵盤のサイズはもともと大き過ぎるので最上段のキーの使用は避けることにした。ただし、配列する文字のうち、ふたつのかぎっこ（「」）はあらかじめ第11列（右手小指外側のホーム段と上段）のシフト側に割り当てておく。本研究で使った文字頻度統計からこれらの2文字が除外されていることと、その他の資料によっても頻度が極めて低いことが示されていることから、配列最適化の検討対象外とした。

2.2 シフト方式

シフト方式は中指ホームポジションでの前置型シフトとする。このようにする理由について述べる。

JIS仮名文字の数は63個である。これに対して仮に使用キー数を33鍵に限定するとすれば、30個の文字はシフト側に配置される。単純に高頻度の33文字をノーマル側に、低頻度文字をシフト側にそれぞれ振り分けると、シフト側の文字の打鍵に伴うシフトキーの使用頻度は12.3%にも達する。一方仮名文字のうちで最高頻度の濁点と「イ」の使用頻度はおのおの10.2%と5.7%である。つまり、この場合シフトキーというのは最も頻繁に使用されるキーなのであって、左右2個のシフトキーとしてもなお、「イ」の頻度を上回る。

以上の考察から、前置型シフトキーを最も打鍵しやすい中指ホームポジションに設定する。打鍵データの結果からは、中指ホームポジションの打鍵は必ずしも最も速いというものではないことがわかるが、中指は5指のうちで最も長い指であり、シフトキーとして安定した打鍵動作が期待される。また左右各手の分担するキー群のほぼ中央に位置しており、打鍵時の手指の移動の基準位置ともなることから「基準キー」とも呼ぶべきものである。実際、手のホームポジションを触覚で確認できるように、これらのキーの上面に小さな突起が付いている鍵盤が多い。

2.3 最適化の方針

配列の最適化は、任意に決めた初期配列から出発して、ふたつの文字位置の最良の入れ替えを、もはやいかなる入れ替えも配列を改善しないという点に到達するまで繰り返す、いわゆる「最急降下法」によって行なう。これは、Nickelsや小西らがとったものと同じ手法である。この最適化のために、基礎データとして

(1) キーからキーへの打鍵間時間

(2) 仮名文字の2文字組出現頻度

についての統計データが必要である。これらのデータによって、最適化途上の配列を期待平均打鍵時間によって

刻々評価することができる。

2.4 自由打鍵データ

新しい配列を設計するための基礎データが既存配列に依拠して求められることは好ましくない。そこで、テキストを使用せず、従って、文字配列に全く依拠しないで大量の連続打鍵を行なう実験を実施して、キーからキーへの平均打鍵間時間を調べた【7】。

このようにして得た約112万ストロークの打鍵間データをもとに各キーからキーへのすべての組み合わせについてその平均打鍵間時間を求める。本配列で使用するキーは33キーである。仮名文字は63字あるがこのうち、ふたつのかぎっこ（「と」）は最適化による配置対象外としてQwerty配列の@と:の位置にシフトケース文字として固定する。33キーのうち、中指ホームポジションであるQwertyのDとKのキーは前置型シフトキーとして使用する。したがって31個の仮名文字がノーマルケースに、残り30個がシフトケースに回される。シフトケースの文字はクロスシフトで打つものとし、たとえばDキーとこれに続いて打たれた右手用のキーの2打鍵でひとつの仮名文字が入力される。ノーマルケースの場合にはキーそのものをとり、シフトケースの場合にはシフトキーと文字キーの組み合わせたものをもって両者を合わせ、これを論理キーとして定義する。シフトケースの論理キー打鍵時間とは、シフトキーの打鍵時間とその直後の文字打鍵時間を合計したものである。そのうえで、物理的なキーの打鍵データから論理キー間の打鍵間時間を求める。可能な論理キーの組み合わせは $61 \times 61 = 3721$ 通りあるが、120セッション約112万ストロークではそのすべてをカバーできず、2944通りしか出現しなかった。このために意識的にシフトケースを打鍵するようにした20セッションを追加実施した。図1に論理キー間の打鍵間時間分布を示す。図2は論理キー間の平均打鍵時間をその短かい順に並べたものである。傾斜が2か所で変化しているが、左右交互打鍵、同手打鍵および、シフトケースの打鍵がそれぞれの傾斜をもつ3つの直線上にのっている。いくつかの2打鍵列の時間は異常に長く、また、最短時間側にも若干の異常性が認められるがこれについては特に処置をしなかった。

2.5 仮名2文字組出現頻度

仮名2文字組の頻度については、坂村【8】による調査データを使用した。前述したように、これにはふたつのかぎっこが除外されている。このデータはビジネス文書を対象とした約160万文字の母集団から求められている。ちなみに、新JIS仮名配列を設計した際に渡辺ら【4】が

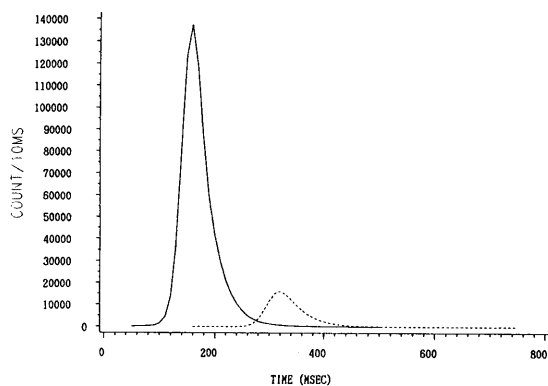


図1 論理キーの2打鍵間時間分布（点線はシフトケース）

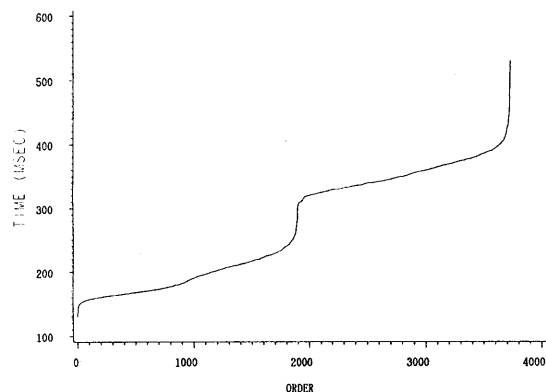


図2 論理キーの2打鍵間時間平均値

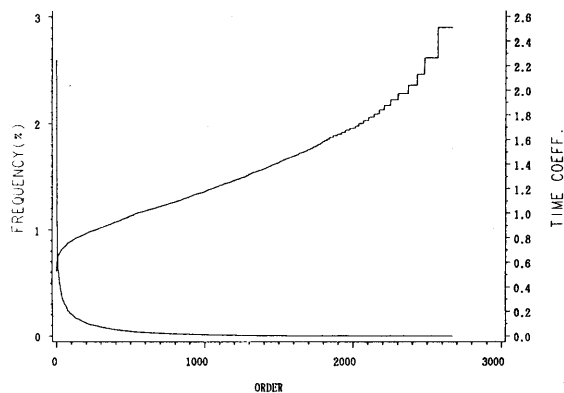


図3 仮名2文字組の出現頻度と頻度効果

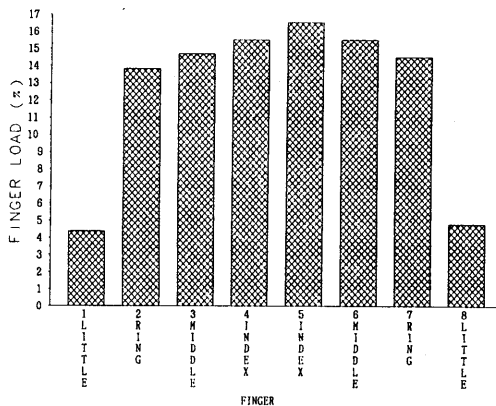


図4 自由打鍵実験から得た目標指負荷配分

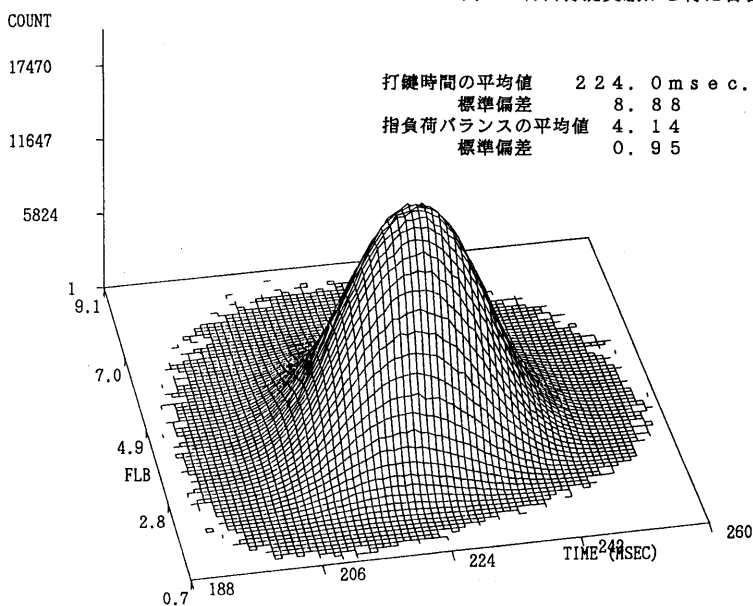


図5 乱数試行配列による平均打鍵時間と指負荷バランスの分布

高校教科書を対象にして求めた頻度表と比較した場合、上位10個の文字までは（順序に若干の違いはあるが）同じである。

2. 6 配列の速度評価

2打鍵間時間および2文字組出現頻度の両者によって試行配列上の1文字あたりの平均打鍵時間を求めることができる。A(i)を配列Aにおける文字iの論理キーとし、 T_{xy} を（論理）キーxからyへの打鍵間時間、 P_{ij} を2文字組<i, j>の出現確率、Pをすべての2文字組にわたる平均出現確率とすると、配列Aにおける1文字あたりの平均打鍵時間Tは次のようになる：

$$T = \sum_{i,j} (T_{A(i)A(j)} P_{ij})$$

しかし、打鍵作業の実務では、同じような運指であっても高頻度文字に対応して頻繁に打たれるものの方が、そうでない場合よりも速くなる傾向があることが知られており、これを「頻度効果」と呼ぶ。そこで、自由打鍵実験によるデータから実務環境下のデータを近似するようにするために、頻度効果による補正を加えた。頻度効果とは、つまるところ反復回数の違いによる能率向上の差であるので、その定量的な評価には練習のべき乗則を適用することができる。この補正では、キーによるポインティング作業時の学習パラメータ【9】を取り入れて次のようにして実効的な平均打鍵間時間を求めた。

$$T = \sum_{i,j} (T_{A(i)A(j)} P_{ij} (\frac{P_{ij}}{P})^{-0.15})$$

すなわち、実験時に観測された打鍵速度は平均出現頻度をもつ2文字組の時に実現され、図3に示すように、最高頻度の2文字組は実験データの1.9倍速く、また、最低頻度のものでは2.5倍遅く打たれるものと推定される。

2. 7 配列の総合評価

文字配列を1文字あたりの平均打鍵時間だけで評価しようとすると、速く打てる指に対して過度の負荷集中が起こるおそれがある。指の速度能力は必ずしも持久力を表わしてはいないので、長時間の使用が前提となる鍵盤文字配列では、打鍵高速化のために局所的な疲労の犠牲を強いることとなる。このように、「良い」配列を「速い」配列という一面からのみ追求することには問題がある。

そこで、配列の評価基準として平均打鍵速度とともに、各指にかかる負荷の配分が望ましい配分に近いかどうか（指負荷バランス）を用いることにした。ここで、目標とする指負荷は、どの指に何％といった、具体的な数値でなければならない。このため、自由打鍵実験のデータから各指の使用頻度を求め、これを使用することにした。自由打

鍵実験では各指への負荷について何らの制限や拘束もないので、被験者にとっての「自然な」負荷の配分を無意識のうちに行なっているものと推定されるからである。図4はこのようにして求めた目標指負荷配分である。ただし、右手人差し指から0.9％（ポイント）を取り去りこれを左手中指へ与えてある。これは、人差し指－中指－薬指の序列についての経験的知見に合わせるために行なった修正である。この図から、人差し指・中指・薬指の3指と小指の間には与えるべき負荷に約3倍という大きな差のあることがわかる。これには指の長さに関係しているのではないかと思われる。

指負荷バランスは、この目標配分と与えられた配列における配分の差であって、 G_f を目標指負荷配分における指fへの負荷、 L_f を評価対象配列における指fへの負荷としたとき、

$$\sum_f \left(\frac{|L_f - G_f|}{G_f} \right)$$

と定義する。

スカラー量としての配列評価値は、1文字あたり平均打鍵時間とこの指負荷バランスのふたつの成分からなるベクトルと、各成分に対する重み付けをする評価係数ベクトルのスカラー積をとる。このとき、平均打鍵時間と指負荷バランスは、多数のランダムな配列におけるおのおのの平均値からのずれを標準偏差を単位として計ることによって正規化したものを用いる。

可能な仮名文字配列の数は $63! = 1.98 \times 10^8$ の7乗通りあるが（実際には「J」を除いて $61! = 5.08 \times 10^8$ の83乗）、ここでは乱数によって10の7乗個の試行配列を生成して平均打鍵時間および指負荷バランスの分布を調べた。打鍵時間の分布は正規分布に近いが、指負荷バランスはピーク値の回りでやや非対称となっており、値の大きい方に向かって尾を引いている。図5にその結果を2次元的に表示する。

4. 最適化

最適化は、初期配列を与えて、ふたつの文字キー（シフト側文字も含めた論理的キー）のすべての置換の中から最良の評価値を与えるものによって実際の置換をおこない、もはやいかなる置換も評価値の改善をみないところ（極小値）までこの操作を繰り返す、いわゆる最急降下法によった。あるひとつの配列に隣接する（1回の置換で到達する）配列は1830個（ $61 \times 60 / 2$ ）あるので、これは、 2×10^8 の87乗個の点からなる離散空間の中での1830自由度の極値探索の問題であるといえる。また、ひとつの任意の配列から他の任意の配列へは高々60回の置

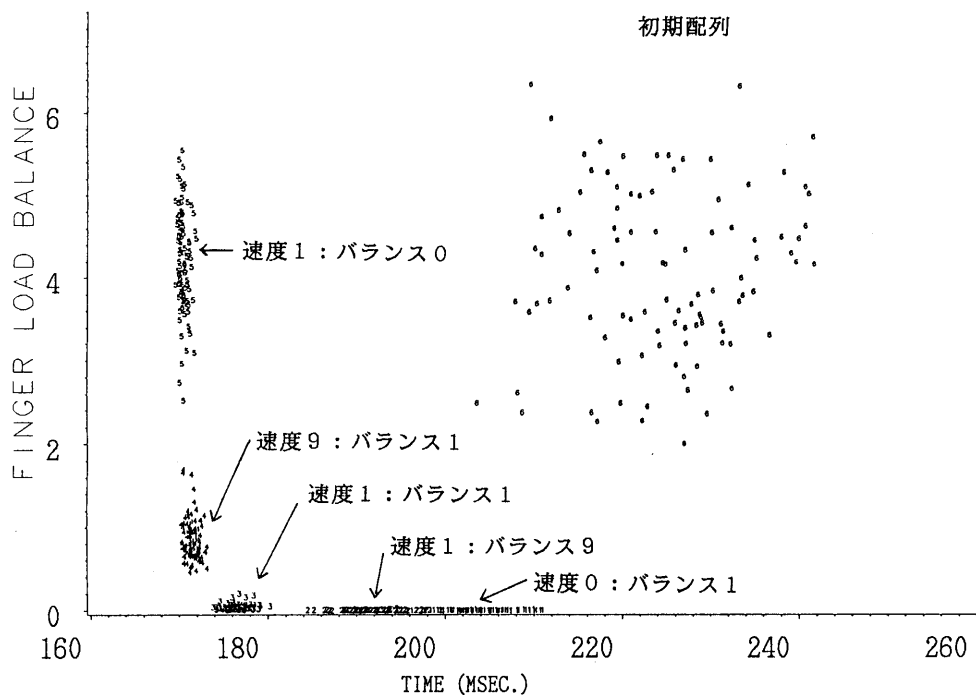


図6 各評価ベクトルによる100回の最適化

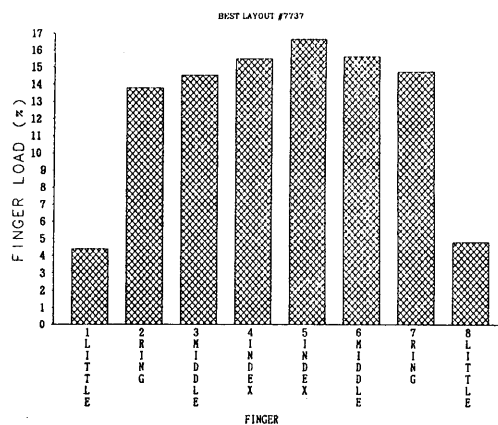


図7 最適化配列における指負荷配分

E	S	C	!	"	#	\$	%	&	'	(	)	0	=	.		B S
T A B	Q ひょ	W けて	E えと	R ほこ	T は	Y やっ	U みく	I そう	O むる	P いー	@「ろ	[				C R
CTRL	CAPS	A あす	S よか	D ゆ☆	F やき	G °た	H ．ん	J ふれ	K ち☆	L も・	; おら	:」り	]			
S H I F T	Z うさ	X しせ	C あな	V わに	B ゆの	N ねい	M まつ	, を。	, おえ	/ ぬめ	- め					S H I F T
	カナ	GRPH	NFER				S P A C E					XFER				

圖 8 最適化配列－「花」－

換で到達できるので、この空間での最大距離は60である。

いったん極小値が見つかったとしても、そのごく近くにより良い配列が存在する可能性がある。そのような見逃しを少しでも防ぐために、極小値からさらに沢づたいに最大10ステップまでの探索をおこなうようにした。その結果、峠を越えて再び沢を下り、前の極小値よりも良い配列が見つければその地点から通常の降下を続行する。そうでなければ探索は終了する。

最適化された配列の特性は評価係数ベクトルによって当然にも異なる。また、同じ評価係数ベクトルを使用したとしても、初期配列によって収束する先の配列は異なる。そこで、評価係数ベクトル〈打鍵時間、指負荷バランス〉として〈1, 0〉、〈9, 1〉、〈1, 1〉、〈1, 9〉、〈0, 1〉をとり（実際にはノルムを1にとって正規化する）、乱数によって生成した100個の異なる初期配列からの最適化を行なってみた（図6）。この図から、速度のみの最適化では指負荷バランスは初期値からみてほとんど改善されないことがわかる。これは従来の速度のみに着目した最急降下法の重大な欠点であった。また〈1, 1〉の場合には、負荷バランスを目標配分に対して数%の範囲で一致させたままで速度がギリギリまで改善されている。終期配列は初期配列にくらべて非常に小さな範囲にまとまっているが、それでもばらつきは残る。これは、初期配列を変えて数多くの最適化を繰り返すことによってより良い配列を見い出せる可能性を示している。

この結果をもとに、最終的な最適化では、評価係数ベクトルとして〈1, 1〉を使用し、乱数によって生成した1万個の異なる初期配列からの最適化を繰り返した。

最適化プログラムはfortranで書かれ、ベクトル型計算機HITACS820/80上で実行された。計算の実行効率を上げるために、順序変更によって最内側ループの繰り返し回数を多くし、配列の評価値はキー置換による差分のみを求める、などの工夫をした。1万回の最適化には5時間15分のCPU時間を要した。1回の最適化につき平均1.88秒で、このときの実行時ベクトル化率（VPU/CPU）は92%であった。最適化過程はいずれも60～120回のキー置換によって収束している。このようにして得られた1万個の最適化配列のうち、最も評価の良かったものについて、その指負荷配分を図7に、文字配列を図8に示す。

この「花」配列において、

仮名1文字あたりの平均打鍵時間は172.7 msec.、

（10の7乗個の試行配列の平均値は224.0 msec.）

指負荷バランスは0.044、

（10の7乗個の試行配列の平均値は4.14）

また、

仮名1文字あたりの平均所要打鍵数は1.18

（濁音を1文字とすれば1.31）

であった。

5. まとめ

仮名鍵盤配列のために、シフトキーを中指ホームポジションに設定する「中指シフト方式」を提案した。既存文字配列によらない自由打鍵実験によって2打鍵間時間データと指負荷配分を求め、頻度効果を組み込んだ速度と指負荷バランスの2元評価による最急降下法を適用し、ベクトル型計算機を使用した大規模計算による最適化を実行し、新しい仮名文字配列を見い出した。

謝辞

打鍵時間測定プログラムを提供して下さった通産省製品科学研究所の森川治研究官と、仮名文字出現頻度表の使用を許諾された東京大学理学部情報科学科の坂村健助教授に感謝します。

参考文献

- [1] Nickels, Robert F. Jr. : The Design of an Optimal Typewriter-like keyboard, M. S. Thesis in Industrial Engineering, Lehigh University, Bethlehem, Pennsylvania, vii+57 pp. (1973)
 - [2] Hisao Yamada: A Historical Study of Typewriters and Typing Methods: from the Position of Planning Japanese Paragraphs, Journal of Information Processing, Vol. 2, No. 4, pp. 175-202 (1980)
- [小笹和彦訳: タイプライタとその入力方法の歴史的考察1～6, bit, Vol. 13, No. 7, pp. 872-880, Vol. 13, No. 8, pp. 974-983, Vol. 13, No. 9, pp. 1112-1121, Vol. 13, No. 10, pp. 1249-1255, Vol. 13,

- No. 11, pp. 1547-1556, Vol. 13, No. 13, pp. 1662-1673 (1981)]
- 【3】 小西和憲, 樽松 明, 田代秀夫: 英語鍵盤配列の評価, 電子通信学会電子計算機研究会資料, EC 81-21, pp. 45-52 (1981)
- 【4】 渡辺定久: 仮名漢字変換形日本文入力装置用けん盤配列の標準化について, 情報処理学会「日本語文書の入力と編集」シンポジウム報告集, pp. 9-16 (1985)
- 【5】 白鳥嘉男, 小橋史彦: 日本語入力用新キー配列とその操作性評価, 情報処理学会論文誌, Vol. 28, No. 6, pp. 658-667 (1987)
- 【6】 Timothy A. Salthouse: The Skill of Typing, Scientific American, Vol. 250, No. 2, pp. 94-99 (1984)
- 【7】 富樫雅文: 自由打鍵実験による打鍵時間分析, 情報処理学会ヒューマンインターフェース研究会報告27-3 (1989)
- 【8】 坂村 健: BTRONにおける入力方式-TRONキーボードの設計, 情報処理学会日本語文書処理研究会報告7-2 (1986)
- 【9】 古川康一, 溝口文雄: インターフェースの科学, 共立出版 (1987)