

階層メニューの最適化手法の提案

松井 正一[†] 山田 誠二^{††}

[†] (財) 電力中央研究所 システム技術研究所

201-8511 東京都狛江市岩戸北 2-11-1

^{††} 国立情報学研究所

101-8430 東京都千代田区一ツ橋 2-1-2

E-mail: [†]matsui@criepi.denken.or.jp, ^{††}seiji@nii.ac.jp

あらまし 階層型メニューは様々なアプリケーションのユーザインタフェースとして広く使われている。この設計においては、メニュー項目の選択に要する手間を軽減することが重要である。選択の手間はメニューの構造、レイアウト、色などの様々な要因で決まる。手間を軽減するための新しいメニュー方式についての研究は多くなされてきたが、構造を変更する観点からの研究は少ない。本報告では、遺伝的アルゴリズム (GA: Genetic Algorithm) を用いて選択の手間を軽減するメニューを自動的に設計する手法を提案する。また、携帯電話を例に提案手法の有効性を示す。
キーワード 階層メニュー、最適化、遺伝的アルゴリズム、ユーザインタフェース、携帯電話

Optimization of Hierarchical Menus

Shouichi MATSUI[†] and Seiji YAMADA^{††}

[†] System Engineering Research Laboratory, Central Research Institute of Electric Power Industry

2-11-1 Iwado-kita, Komae, Tokyo 201-8511

^{††} National Institute of Informatics

2-1-2 Hitotsubashi, Chiyoda, Tokyo 101-8430

E-mail: [†]matsui@criepi.denken.or.jp, ^{††}seiji@nii.ac.jp

Abstract Hierarchical menus are widely used as a standard user interface in modern applications that use GUIs. The performance of the menu depends on many factors: structure, layout, colors and so on. This paper proposes algorithm based on the genetic algorithm (GA) for optimizing the performance of menus. The algorithm aims to minimize the average selection time of menu items by considering the user's pointer movement and search/decision time. We will show the results on a static hierarchical menu of a cellular phone as an example where a small screen and limited input device are assumed. We will show the effectiveness of the algorithm by using wide variety of the usage patterns.

Key words hierarchical menu, optimization, genetic algorithm, user interface, cellular phone

1. はじめに

階層メニューは GUI でコマンドを指定する目的で広く用いられている。階層メニューの性能 (本稿では、目的とするメニュー項目に辿り着くまでの平均時間を平均到達時間と呼び、それを階層メニューの性能とする) は構造、レイアウト、色などの多くの要因によって決まる。現在までに、ユーザインタフェース分野で多くの研究が行われ、様々なメニュー方式が提案されているが [1]~[3]、構造を変更することで性能を向上するという観点からの研究は少ない [4]~[6]。

この観点からの研究としては Francis らによる多機能ディスプレイの表示方式の最適化の研究が挙げられ、そこでは SA

(Simulated Annealing) が最適化に用いられている [5]。また、Liu らは視覚探索のモデルをメニュー設計に適用し、Guided Search (GS) によって、予め与えられた使用頻度に基づいて平均到達時間を最小化する最適設計を試みている [6]。Amant らは携帯電話のメニュー構成を分析するための方法を提案している [4]。彼らは、Fitts の法則、GOMS、ACT-R の 3 モデルに基づいてインタラクションをモデル化することを提案しており、GOMS モデルによる予測が最も適しており、このモデルを用いて、単純な最適化手法であっても、最大 30% 程度の時間削減が可能であることを示している。

単純な最適化手法によっても、平均到達時間の短縮が可能となっていることから [4][6]、適切な最適化手法によりさらなる

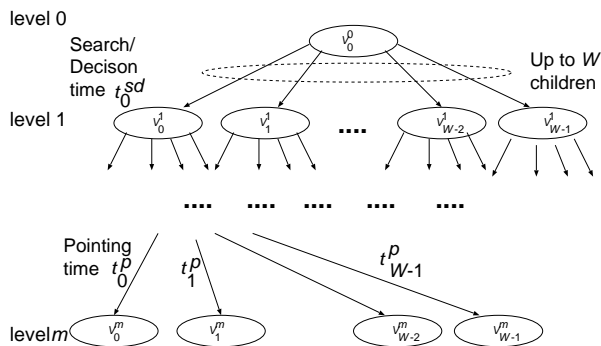


図1 階層メニューの木構造表現

Fig. 1 Tree structure of a hierarchical menu.

時間短縮が可能と考えられる．そこで我々は遺伝的アルゴリズムによる方式を提案し，携帯電話のメニューの最適化により40%程度の選択時間が可能となることを示した[7]．提案する手法では，ある階層における探索/意思決定時間とポインティング時間の双方を考慮して，平均到達時間を最小化する．本報告では，表示画面と入力方式が限定された携帯電話を例に，既発表の論文よりも多くの実験結果を用いて，有効性を示す．

2. 問題の定式化

2.1 概要

メニュー項目へ平均到達時間を最小化する階層メニューの最適化は，木構造のノードにメニュー項目を適切に配置する問題として定式化できる．最大の深さが D で1ノードが最大 W の子を持つ木構造を考え，根が初期状態に相当し，メニュー項目はノードに配置されるものとする．図1に示すように，中間項目に相当するノードは子としてサブメニューを持つ．目的の項目を選択するために必要な時間は，根から目的のノードまでの到達時間となる．最適化の目的は，各項目毎に与えられる利用頻度分布の下で，平均到達時間を最小化することである．

使いやすさの観点からは，効率だけを考慮して任意の項目を任意の場所に配置することは望ましくなく，項目の意味を尊重する必要がある．例えば，「音量調整」メニューは「設定」の下にあることが自然であり，「メール」の下に配置されると記憶することが困難であり，使い勝手を損ねる．また，サブメニューの項目数が大きく異なるメニューも使い勝手を損ねる．この問題に対応するために，「機能の類似度」と「メニューの粒度」という二つの尺度を導入する．

機能の類似度はメニュー項目間の機能の類似性を表現するものであり，二つのメニュー項目に対して類似度が最大の場合に1，最小の場合に0をとる関数として定義する．「新規メールの作成」と「Webのお気に入り」の類似度は低く，「新規メールの作成」と「受信ボックス」の類似度は高くなるような関数を考える．この尺度を用いて，類似度の低い項目が同一ノードの子ノードとして配置されないようにする．

メニューの粒度は子ノードの持つサブメニューの数から決まる関数として定義する．これは，前述のようにサブメニューの数ができるだけ均一になるように配置するためのものである．

2.2 標記法

木構造の階層番号を l で，子の順番を i で，ノードを v_i^l で表す(図1)． $V = \{v_i^l\}$ がノード集合を， $E = \{e_{ij}^l\}$ (e_{ij}^l はレベル l のノード i とレベル $i+1$ のノード j の間の枝を表す) が枝集合を表すものとし，木を $M = (V, E)$ で表現する．機能に対応するノードを「終端ノード」と呼び，サブメニューを子として持つノードを「中間ノード」と呼ぶ．機能 i の頻度は選択確率 P_i で与えられるものとする．メニュー項目を I_i で表現し，その総数は N であるとする，つまり， $I_i (i = 1, \dots, N)$ であり，機能の数は N 以下である．結果としてメニュー項目 I_i はノード v_i^l に割当られるとし， I_i と v_i^l は同一視する．

2.3 到達時間

レベル l におけるノード，メニュー項目 v_i^l への到達時間 t_i^l は探索・意思決定時間 t_i^{sd} とポインティング時間 t_i^p を用いて $t_i^l = t_i^{sd} + t_i^p$ と表現できる[8]．最終的な項目を選択するためには，根からレベル l まで辿り着く必要があり，レベル l にあるノード v_i^l の到達時間 T_i は根から目的ノードまでのパス上の t_i^l 総和となる．したがって平均到達時間 T_{avg} は $T_{avg} = \sum_{i=1}^N P_i T_i$ で表せる．

2.4 ポインティング時間，探索・意思決定時間

Silfverberg ら[9] と Cockburn [8] が報告しているように，ポインティング時間 t_i^p は Fitts の法則を使って $t_i^p = a + b \log_2(A_i/W_i + 1)$ で表現できる．ここで， $\log_2(A_i/W_i + 1)$ は「困難度の指標」(Index of difficulty) と呼ばれる．係数 a, b は実験データの回帰式で決定する．

レベル l にある n^l 個の項目を持つノードでの探索・意思決定時間 t_i^{sd} は，以下の形になると仮定する[8]．

- 初心者では線形探索を仮定し $t_i^{sd} = b^{sd} n^l + a^{sd}$ とする．
- 熟練者では，Hick-Hyman 法則が成り立つものとし， $H_i = \log_2(1/P_i^l)$ として， $t_i^{sd} = b^{sd} H_i + a^{sd}$ を仮定する．選択確率を一定とすれば，項目数を n として $H_i = \log_2(n)$ である．

それぞれの式の係数は実験データの回帰式で決定する．

2.5 機能の類似度

Toms らは機能の説明文書の情報を用いてクラスター分析を行うことで階層メニューの構造を生成する方式を提案している[10]．しかし，機能を十分に説明した説明文書を用意し，それに基づいた複数人による機能の関連付けが必要があるため，この方法に要する手間は大きい．そこで，以下の方法で類似度を定義する．

項目 I_x と I_y の機能の類似度を0から1の範囲の値をとる関数 $s(I_x, I_y)$ で表現する．項目 I_i はキーワードの集合 $wl_i = \{w_1, w_2, \dots\}$ で特徴付けられるとする．また，このキーワード全体を $WL = \bigcup_i wl_i$ で表現し，キーワードに0からの番号をふるものとする．中間ノードを特徴付けるキーワードは，子として持つメニュー項目のキーワード集合の和集合とする．項目 I_x のキーワードの頻度を表すベクトル I_x を考える．項目 I_y についてのベクトル I_y とする．このとき，機能の類似度 $s(I_x, I_y)$ は $s(I_x, I_y) = \frac{I_x \cdot I_y}{|I_x| |I_y|}$ とする．語の頻度ベクトルを用いる文書ベクトル間の余弦距離である[11]． m 個

の項目を持つノード v_i^l での機能の類似度のペナルティ $P_{v_i^l}^s$ は、 $P_{v_i^l}^s = \sum_{x=0}^{m-1} \sum_{y=0}^{m-1} (1 - s(I_x, I_y))$ と定義する。全体のペナルティ P^s は、 $P^s = \sum_{v_i^l \in V \setminus v_0^0} P_{v_i^l}^s$ とする。

2.6 メニューの粒度

v_i^l でのメニューの粒度 $g_{v_i^l}$ を考える。 v_i^l が終端ノードの場合は $g_{v_i^l} = 0$ とする。中間ノードであり、 v_i^l が m 個の子を持ち ($v_j^{l+1}, j = 0, \dots, m-1$) それぞれのメニューの粒度が $g_{v_j^{l+1}} (j = 0, \dots, m-1)$ である場合には、 $g_{v_i^l}$ は、 $g_{v_i^l} = \sum_{j=0}^{m-1} g_{v_j^{l+1}}$ と定義する。ノード v_i^l でのメニューの粒度のペナルティ $P_{v_i^l}^g$ は、 $P_{v_i^l}^g = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} |g_{v_i^l} - g_{v_j^l}|$ と定義する。全体としてのペナルティ P^g は $P^g = \sum_{v_i^l \in V \setminus v_0^0} P_{v_i^l}^g$ で定義する。

2.7 最適化問題

目的関数は以下となる。

$$f = T_{avg} + \alpha P^s + \beta P^g, \quad (1)$$

ここで α と β は機能の類似度とメニューの粒度をどの程度重視するかを調整するためのパラメータである。

ノード V に配置すべき項目が与えられた場合には、平均到達時間を最小化する項目の順序は簡単に決まる。すなわち、頻度の高いものをポインティング時間の短い位置に配置することで、平均到達時間は極小となることから、式 1 を最小化する、ノードへの割当を求める問題となる。

N 個の項目をいずれかのノードに割り当てる必要があることから、少なくとも $L = \lceil N/W \rceil$ 個のノードから木は構成されることになる。ノードに順番を付けると、1 番のノードには N 個から選んだ W 個の項目が割り当てられる。2 番目のノードには残りの $N - W$ 個から選んだ W 個の項目が割り当てられ、同様にしてすべての項目が割り当てられる。従って、探索空間の大きさは、ほぼ $N C_W \times_{N-W} C_W \times \dots \times_{N-LW} C_W = N! / (W!)^L$ であり、非常に複雑な組合せ最適化問題となる。例えば、 $N = 200$, $W = 10$ なら、探索空間の大きさは、ほぼ $200! / (10!)^{20} \sim 10^{243}$ となる。

3. 遺伝的アルゴリズム

モダンヒューリスティックの 1 手法である遺伝的アルゴリズム (Generic Algorithm: GA) は、生物の進化過程を模擬した計算手法である。最適解が求まる保証はないが、関数の勾配情報を必要とせず、良い近時解が求まることが多いことから、様々な最適化問題に適用されている [12]。前述のように階層メニューの最適化問題は、複雑な組合せ最適化問題であることから、提案手法は GA を使うこととした。

3.1 基本戦略

既往研究によれば、階層メニューでは深さ優先より幅優先の方が使いやすいとされている [13] ~ [15]。そこで、提案方式では幅優先でノードに項目を割り当てる方式を基本とし、割当順序を遺伝的アルゴリズムで探索する。

3.2 染色体表現

前述の考え方に則り、以下の方式で割り当てる割当順序を染

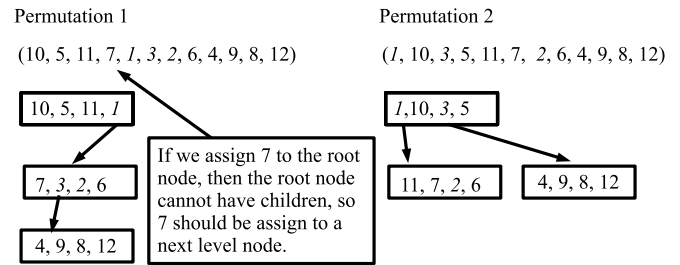


図 2 順列から木構造への変換方式
Fig. 2 Mapping from permutation to a tree structure.

色体に表現する。

(1) メニュー項目 I_i に 1 から順番を付け、染色体はその順番の順列とする。

(2) 染色体の順列にしたがって、割当可能なノードに I_i を一つずつ割当てる。機能に相当する項目が割当てられた場合には、その下にサブメニューを持つことはできない。したがって、その下のノードは全て割当不能となる。逆に、中間ノードの場合には下にサブメニューを持つことができる。この制約を用いて割当可能かどうかを判定する。

$W = 4$ の場合の例を図 2 に示す。この例は、中間ノードをその番号を 1~3 として 3 個用意した場合であり、イタリック体で中間ノードを表す。“Permutation 1” では、“10”、“5”、“11” は根ノードに割当可能である。しかし、“7” を根ノードに割り当てると木はそれ以上大きくできないため、割当不能である。従って、“7” は次のレベルのノードに割り当てる必要があり、根ノードには中間ノード “1” を割り当てる必要がある。

“Permutation 2” の例では最初の “1” は中間ノードであることから、根ノードに割当可能であり、その子供として 4 つの空きが増える。“10” 以降の番号は順次割当可能である。

この順列から木への変換方式を用いる場合、中間ノードが十分な数用意されていれば、最適解が含まれる空間を探索できる。

3.3 他の GA パラメータ

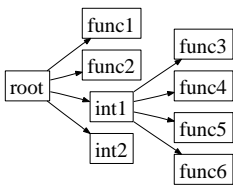
実行不能解を生成しない交叉方式を用いる。このためには巡回セールスマン問題を順列表現で解くために考案されたいくつかの方式が利用できる。具体的には、予備実験の結果に基づいて CX (Cyclic Crossover) [16] を用いることとした。

突然変異のためのオペレータとしては、要素の入替え方式を用いる。具体的にはランダムに選んだ遺伝子座 p と q を入替える方式を用いる。これらの操作によって実行不能解は生成されない、すなわち生成される個体、順列は必ず実行可能解となる。

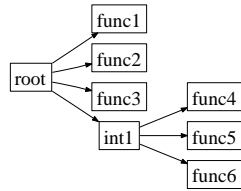
世代交代モデルとしては、総ての個体を入れ替える単純 GA ではなく、組合せ最適化問題で標準的に用いられる定常状態 GA [17] を用い、個体数は 100 とし、突然変異率は 1 回の入替えとする。個体評価回数 (生成されたメニューの評価回数) は 100,000 回とした。

3.4 局所探索

提案手法の探索性能を向上させるために、局所探索も組み入れた。この局所探索では、子供を持たない中間ノード v_i^l を探索し、平均探索時間が短縮される場合には、 v_i^l の兄弟ノードの



left: before local search



right: after local search

図3 局所探索
Fig.3 Local search.



図4 対象とした携帯電話のキー配列
Fig.4 Key layout of the target cellular phone.

子供であるノード v_j^{l+1} と入れ替える (図3)。図3の左側で子供を持たない中間ノード“int2”を“func3”と入れ替えることで、右側の木構造とする。

4. 実験

報告者の1名が日常的に利用している携帯電話のメニューを対象とした実験を行った。この端末[18]は24個のキーが図4に示すように配置されている。

4.1 実験に用いたデータ

4.1.1 困難度の指標

24 × 24 のキーの組合せに対する困難度の指標 (ID) は以下のようにして求めた。まず、24個のキーについて、その中心座標、幅、高さを計測した。これを用いて、小数点以下1桁の精度でIDを求めた結果、28個のグループが得られた。

4.1.2 ポインティング時間

親指1本で操作を行う無報酬の被験者によって、上述の28個の組合せについて、キークリックの音を録音する[4]ことで、ポインティング時間を計測した。得られたデータから回帰直線を求めたところ、ポインティングタイムの予測式として $t_i^p = 192 + 63 \log_2(A_i/W_i + 1)$ (ms) が得られた。この式の係数は Silfvergerg らが報告しているものにかなり近い[9]^(注1)。携帯電話でのメニュー選択はキーを押すことで行われることから、下の階層のメニューへの遷移におけるポインティング時間は、この階層に辿り着くために押したキーと次の階層に遷移するために押すキーのIDによって決まるものと考えられる。対象とした端末にはメニューによっては、数字番号が割当てられており、それをショートカットキーとして利用できる機能があるが、実験ではこの機能は用いないこととした。すなわち、カーソル

キーにより項目間を遷移するものとした。

4.1.3 探索・意思決定時間

メニューの個数を様々に変えて探索・意思決定時間を計測することを対象とした端末を行うことは困難であったこと、ポインティング時間が既往報告とよく合致することから、探索・意思決定時間には既往報告[8]の次の式を用いた。また、対象とする利用者は熟練者を想定した。 $t_i^{sd} = 80 \log_2(n^l) + 240$ (ms)[8]^(注2)

4.1.4 利用頻度と機能の類似度

利用頻度データは報告者の一人が2ヶ月間にわたって毎日の利用履歴を記録したデータから生成した。これを“Original”と呼ぶ。またこのデータから、メールの利用頻度の高いユーザ、Webの利用頻度の高いユーザを想定して、以下の4頻度データを生成した。

Mail2 メール関連の頻度を2倍したもの。

Mail3 メール関連の頻度を3倍したもの。

Web2 Web関連の頻度を2倍したもの。

Web3 Web関連の頻度を3倍したもの。

これは文献[19],[20]で報告されているように、利用者の特性によって携帯の利用実態が異なることを想定したデータである。すなわち、1日あたりの受信メールの数が10~30通と50通以上の頻度が高いことを反映したものがMail2とMail3である。また、若年層は他と比べてSNSやブログの利用頻度が高いことを反映したものがWeb2とWeb3である。

機能の類似度に用いるキーワードは、利用者マニュアル[18]と実際のメニューに表示される単語(英単語)を用いた。

4.2 実験結果

以下に示すケースについて計算を行った。GAは確率的な計算であることから、各々のケースについて50回の試行を行った結果の平均を求めた。ペナルティ項のウェイトは $\alpha = 10.0$, $\beta = 1.0$ とした。また探索・意思決定時間の推定式として既往研究のものを使うことから、ノードにおける選択/意思決定時間の算出では、項目の選択確率は一定とした。

case 1 基本ケース: 提案手法によりどの程度の向上があるかを求めた。1ノードあたりの最大項目数は16とした。

case 2 最大項目数を限定: 既往研究結果[13]~[15]によれば深さよりも幅優先の構成が好まれるが、「下」カーソルを何度も押すことが好まれないことも考えられる。そこで、最大項目数を12, 9, 6と変化させた場合の計算も行った。

表1で“Local Move”は1ノード内の配置は頻度が高いものほどポインティング時間が短い場所に配置した場合を表す。また“Imp(%)”は元のメニュー(Original Menu)からの時間短縮率を示す。ここで、時間短縮率は、元のメニューでの平均到達時間を T_o 、配置を変更した場合の平均到達時間を T_m としたとき、 $(T_o - T_m)/T_o$ である。

表1から、提案手法により平均到達時間を大きく短縮できることが分かる。最大項目数を限定することにより、平均到達時

(注1): $t_i^p = 176 + 64 \log_2(A_i/W_i + 1)$ (ms).

(注2): この式は携帯電話ではなくコンピュータディスプレイでの実験結果に基づくものである。

表1 平均選択時間の短縮効果

Table 1 Improvements in average selection time.

| Case | $T_{ave}(ms)$ | Imp(%) | P^s | P^g |
|---------------------|---------------|--------|-------|-------|
| Original Menu | 3331 | 0.0 | 399 | 793 |
| Local Move | 2812 | 15.0 | 399 | 793 |
| Case 1 | 2035 | 38.9 | 675 | 1261 |
| Case 2 ($W = 12$) | 1979 | 40.6 | 485 | 862 |
| Case 2 ($W = 9$) | 1948 | 41.5 | 351 | 301 |
| Case 2 ($W = 6k$) | 2233 | 33.0 | 228 | 177 |

表2 ウェイトの影響

Table 2 Effect of weights.

| α | β | $T_{ave}(ms)$ | (%) | P^s | P^g |
|----------|---------|---------------|------|-------|-------|
| 0 | 0 | 1823 | 45.3 | 523 | 462 |
| 5 | 1 | 1918 | 42.4 | 358 | 283 |
| 10 | 1 | 1948 | 41.5 | 351 | 301 |
| 20 | 1 | 2013 | 39.6 | 347 | 305 |
| 40 | 1 | 2072 | 37.8 | 345 | 320 |
| 20 | 5 | 2020 | 39.4 | 348 | 274 |
| 20 | 10 | 2005 | 39.8 | 354 | 260 |

間が短くなることがあるが、これは、ノードに割当られた項目数を n として、 $\log_2(n)$ に比例する形で探索/意思決定時間が増加するためである。最大項目数を6まで小さくすると、平均到達時間は長くなるが、ペナルティ項の値は小さくなる。

図5と図6に元のメニューと $W = 9$ で最も目的関数の値が小さかったメニュー ($T_{ave} = 1962(msec)$) の例を示す。図では紙面の制約のために頻度の低いノードは省略してある。また、同一レベルであれば、上にあるノードほど頻度が高いように並べてある。両図を比較すると、頻度の高い項目がレベルの低い位置に配置され、また元のメニューでは使われていなかった“Shortcut”キーが有効活用されていることがわかる。

4.3 ペナルティ項の影響

機能の類似度と粒度を制御するために導入した二つのペナルティ項の影響をみるために、ウェイトを変えて計算した結果が表2である。この表から分かるように、ウェイトを0以外に設定した場合のケースは似かよっている。ウェイトを0とすることで平均到達時間は短くなるが、ペナルティ項の値は大きくなる。結果として得られるメニューの構成も異なるグループのものが混在した形となり、記憶することは困難と思われる。

4.4 他の頻度データに対する性能

表3にメール利用頻度の多いユーザを想定した頻度データ (Mail2, Mail3) と Web の利用頻度の多いユーザを想定した頻度データ (Web2, Web3) に対する計算結果を示す。表から分かるように、どのデータセットに対しても提案手法は大幅に平均到達時間が短縮されている。

5. 考察と今後の展開

提案手法により平均選択時間を大きく短縮できる携帯電話のメニュー構造を生成できる。提案手法の対象は携帯電話に限定されないことから、今後は他の階層メニューを対象とした検討

を行う予定である。

本報告では、静的な固定構造のメニューを対象とし、動的メニュー（例えば文献[1]~[3]など）を対象とした最適化は今後の課題である。また、実験に用いたデータは限定的であることから、特に利用頻度のデータは1人のデータである、より多様な頻度データにより提案手法の有効性を確認することも、今後の課題である。

6. おわりに

探索・意思決定時間とポインティング時間を考慮した、階層メニューの項目選択に要する時間の平均値を最小化するための遺伝的アルゴリズムを提案した。携帯電話を対象とした実験結果からは提案手法により時間を40%以上短縮できる構造を生成できた。提案手法は携帯電話以外の階層メニュー全般に対して適用可能であり、幅広い応用が可能である。

文献

- [1] D. Ahlström: “Modeling and improving selection in cascading pull-down menus using fitts’ law, the steering law and force fields”, CHI ’05: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 61–70 (2005).
- [2] J. Beck, S.H. Han and J. Park: “Presenting a submenu window for menu search on a cellular phone”, International Journal of Human-Computer Interaction, **20**, 3, pp. 233–245 (2006).
- [3] L. Findlater and J. McGrenere: “A comparison of static, adaptive, and adaptable menus”, CHI ’04: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 89–96 (2004).
- [4] R.S. Amant, T.E. Horton and F.E. Ritter: “Model-based evaluation of cell phone menu interaction”, CHI ’04: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 343–350 (2004).
- [5] G. Francis: “Designing multifunction displays: An optimization approach”, International Journal of Cognitive Ergonomics, **4**, 2, pp. 107–124 (2000).
- [6] B. Liu, G. Francis and G. Salvendy: “Applying models of visual search to menu design”, Int. J. Human-Computer Studies, **56**, pp. 307–330 (2002).
- [7] S. Matsui and S. Yamada: “Genetic algorithm can optimize hierarchical menus”, Proceedings of CHI2008 (2008). to appear.
- [8] A. Cockburn, C. Gutwin and S. Greenberg: “A predictive model of menu performance”, CHI ’07: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 627–636 (2007).
- [9] M. Silfverberg, I.S. MacKenzie and P. Korhonen: “Predicting text entry speed on mobile phones”, CHI ’00: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 9–16 (2000).
- [10] M.L. Toms, M.A. Cummings-Hill, D.G. Curry and S.M. Cone: “Using cluster analysis for deriving menu structures for automotive mobile multimedia applications”, Technical Report 2001-01-0359, SAE (2001).
- [11] R. Baeza-Yates and B. Ribeiro-Neto: “Modern Information Retrieval”, Addison-Wesley (1999).
- [12] D.E. Goldberg: “Genetic algorithms in search, optimization, and machine learning”, Addison-Wesley (1989).
- [13] J.I. Kiger: “The depth/breadth trade-off in the design of menu-driven user interfaces”, J. Int. J. Man-Mach. Stud., **20**, 2, pp. 201–213 (1984).
- [14] E.E. Schultz and P.S. Curran: “Menu structure and ordering of menu selection: independent or interactive effects?”, SIGCHI Bull., **18**, 2, pp. 69–71 (1986).
- [15] P. Zaphiris, S.H. Kurniawan and R.D. Ellis: “Age related difference and the depth vs. breadth tradeoffs in hierarchical online information systems”, Proc. User Interfaces for All, pp. 23–42 (2003). LNCS 2615.

表 3 平均選択時間の短縮効果 (Mail2, Mail3, Web2, Web3)

Table 3 Improvements in average selection time (Mail2, Mail3, Web2, Web3).

| Case | Mail2 | | | | Mail3 | | | | Web2 | | | | Web3 | | | |
|---------------------|-----------|------|-------|-------|-----------|------|-------|-------|-----------|------|-------|-------|-----------|------|-------|-------|
| | T_{ave} | (%) | P^s | P^g | T_{ave} | (%) | P^s | P^g | T_{ave} | (%) | P^s | P^g | T_{ave} | (%) | P^s | P^g |
| Original Menu | 3186 | 0.0 | 399 | 793 | 3123 | 0.0 | 399 | 793 | 3518 | 0.0 | 399 | 793 | 3631 | 0.0 | 399 | 793 |
| Local Move | 2641 | 17.1 | 399 | 793 | 2568 | 17.7 | 399 | 793 | 3030 | 13.9 | 399 | 793 | 3160 | 13.0 | 399 | 793 |
| Case 1 | 1969 | 40.9 | 670 | 1265 | 1918 | 42.4 | 689 | 1262 | 2079 | 37.6 | 679 | 1264 | 2140 | 35.8 | 696 | 1262 |
| Case 2 ($W = 12$) | 1894 | 43.1 | 486 | 865 | 1834 | 44.9 | 495 | 848 | 2046 | 38.6 | 490 | 875 | 2099 | 37.0 | 487 | 875 |
| Case 2 ($W = 9$) | 1888 | 43.3 | 352 | 297 | 1839 | 44.8 | 358 | 302 | 2017 | 39.4 | 354 | 295 | 2046 | 38.6 | 349 | 295 |
| Case 2 ($W = 6k$) | 2154 | 35.3 | 232 | 183 | 2122 | 36.3 | 232 | 175 | 2291 | 31.2 | 229 | 180 | 2310 | 30.7 | 229 | 178 |

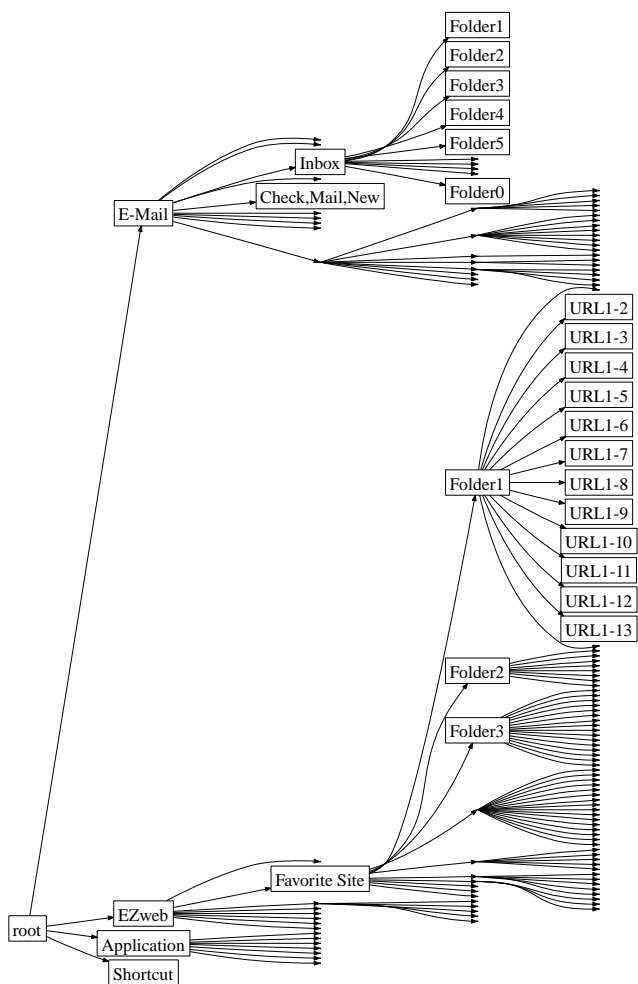


図 5 元のメニュー
Fig. 5 Original menu.

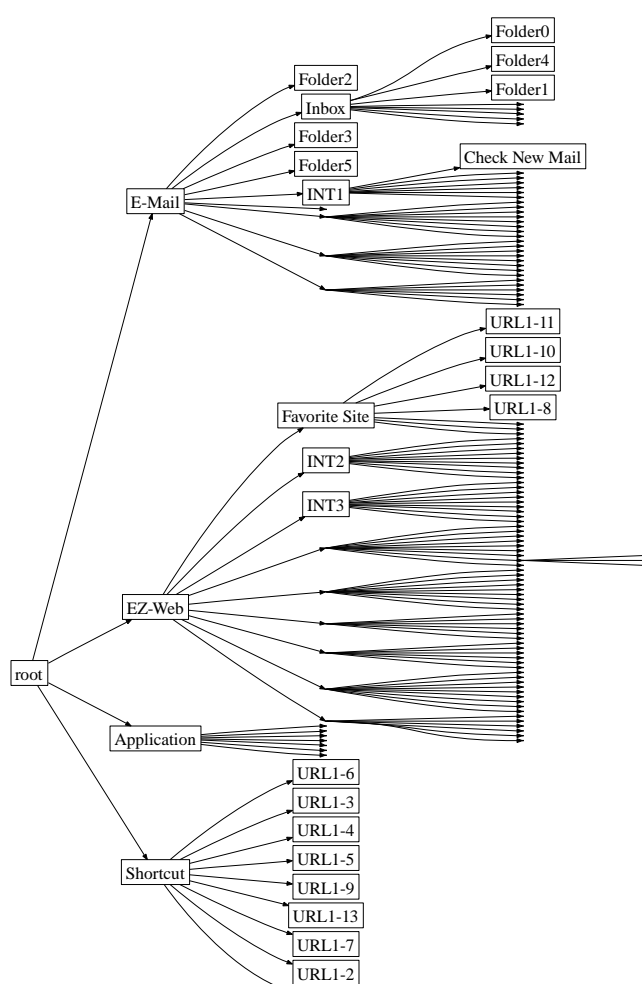


図 6 最適化されたメニューの例 ($W = 9$)
Fig. 6 An example of optimized menu ($W = 9$).

- [16] Z. Michalewicz and D.B. Fogel: "How to Solve It: Modern Heuristics", Springer (2000).
- [17] M. Mitchell: "An Introduction to Genetic Algorithms", MIT Press (1996).
- [18] KDDI: "Maunal for casio w43ca" (2004). http://www.au.kddi.com/torisetsu/pdf/w43ca/w43ca_torisetsu.pdf.
- [19] 日本情報処理開発協会 (編): "情報化白書 2007—ユーザーが牽引するイノベーション—" (2007).
- [20] MMD 研究所モバイルマーケティングデータ研究所: "第 27 回携帯電話の利用に関する実態調査 (2007.7.19)" (2007).