2F2-3

遺伝的アルゴリズムによる階層メニューの最適化

Optimization of Hierarchical Menus by Genetic Algorithm

松井 正一*1　山田 誠二*2
Shouichi Matsui　Seiji Yamada

*1(財) 電力中央研究所
Central Research Institute of Electric Power Industry

*2国立情報学研究所
National Institute of Informatics

Hierarchical menus are now ubiquitous. The performance of the menu depends on many factors: structure, layout, colors and so on. There has been extensive research on novel menus, but there has been little work on improving the performance by optimizing the menu's structure. This paper proposes an 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 movement and decision time. We show results on a static hierarchical menu of a cellular phone where a small screen and limited input device are assumed.

1. はじめに

階層メニューは GUI でコマンドを指定するなどで多用されている．階層メニューの性能は構造，レイアウト，色などの多くの項目によって決まる．現在までに，ユーザインタフェースの観点から多くの研究が行われ，様々なメニューの方式が提案されているが（例えば文献 [Ahlström 05] など），構造を変更することで性能（目的の項目に辿り着くまでの時間が短い）を向上するという観点からの研究は少ない [Amant 04, Francis 00].

Amant らは携帯電話のメニューを対象に，単純な最適化手法により選択時間を削減能している [Amant 04]．．

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 [Matsui 08]．本報告では，既報告よりも多くの実験結果を用いて，携帯電話を例に手法の有効性を示す．

2. 問題の定式化

2.1 概要

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考えることができる．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．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，各項目毎に与えらる利用頻度の下で，平均選択時間を最小化することである．

使いやすさの観点からは，効率だけを考え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．

2.2 定式化

l で木構造の階層番号を，i で子の順番を，v_i^l でノードを表す．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，つまり, $I_i (i = 1, \cdots, N)$ とする．

連絡先: 松井 正一，(財) 電力中央研究所 システム技術研究所，201-8511 狛江市岩戸北 2-11-1，matsui [at] criepi.denken.or.jp

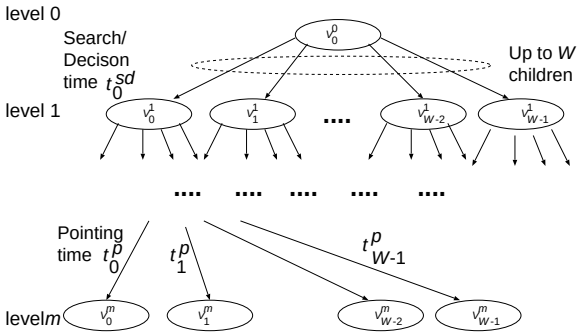

図 1: 階層メニューの木構造表現

2.2.1 選択時間

階層メニュー上のレベル l にあるノード，メニュー項目 v_i^l の選択時間 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 [Cockburn 07]．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 の選択時間 T_i は根から目的ノードまでの総和となる．したがって平均選択時間 T_{avg} は $T_{avg} = \sum_{i=1}^{N} P_i T_i$ で表せる．

2.2.2 ポインティング時間

Silfverberg ら [Silfverberg 00] と Cockburn[Cockburn 07] 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 $t_i^p = a^p + b^p \log_2(A_i/W_i + 1)$ 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^p, b^p は実験データの回帰式で決定する．

2.2.3 探索・意思決定時間

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} は以下と仮定する [Cockburn 07].

- 初心者については $t_i^{sd} = b^{sd} n^l + a^{sd}$ とする．
- 熟練者に対しては，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．選択確率は一定とすれば，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．

2.2.4 機能の類似度

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 はいくつかのキーワード群 $wl_i = \{w_1, w_2, \cdots\}$ で特徴付けられるとする．また，このキーワードの全体を $\mathbf{WL} = \bigcup_i wl_i$ 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の全体とする．項目 x の i 番目のキーワードの頻度を表すベクトル $\mathbf{x}$ を考える．項目 y についてのベクトルを $\mathbf{y}$ とする．このとき，機能の類似度 $s(I_x, I_y)$ は $s(I_x, I_y) = \frac{\mathbf{x} \cdot \mathbf{y}}{|\mathbf{x}||\mathbf{y}|}$ とする．この類似度は情報検索の分野で広く使われているものである．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 $P^s_{v_i^l}$ は，$P^s_{v_i^l} = \sum_{x=0}^{m-1} \sum_{y=0}^{m-1} (1 - s(I_x, I_y))$ と定義する．全体のペナルティ P^s は，$P^s = \sum_{v_i^l \in \{V \setminus v_0^0\}} P^s_{v_i^l}$ とする．

2.2.5 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（$v_j^{l+1}, j = 0, \cdots, m-1$）それぞれのメニューの粒度が $g_{v_j^{l+1}}, (j = 0, \cdots, m-1)$ 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 $P^g_{v_i^l}$ は，$P^g_{v_i^l} = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} \left| g_{v_i^l} - g_{v_j^l} \right|$ と定義する．全体としてのペナルティ P^g は $P^g = \sum_{v_i^l \in \{V \setminus v_0^0\}} P^g_{v_i^l}$ とする．

2.2.6 目的関数

目的関数は以下となる．

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

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要視するかを調整するためのパラメータである．

2.2.7 最適化問題

ノード V に配置すべき項目が与えられた場合には，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，平均的選択時間は最小となることから，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たとする．その順番で 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 ${}_N C_W \times {}_{N-W} C_W \times \cdots \times {}_{N-LW} C_W = N!/(W!)^L$, であり，非常に複雑な組合せ最適化問題となる．例えば，$N = 200, W = 10$ であるとする，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

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 [Kiger 84, Schultz 86, Zaphiris 03]. そこで，提案方式では幅優先でノードに項目を割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．

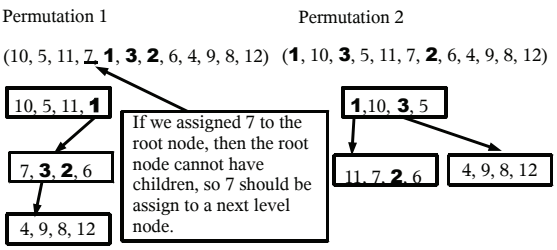

図 2: 順列から木構造への変換方式.

3.2 染色体表現

前述の考え方に則り，以下の方式で割当を行う割当順序を染色体に表現する．

1. メニュー項目 I_i に 1 から始まる番号を付け，染色体はその番号の順列とする．
2. 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ノードはサブメニューを持つことはできない．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．図でボールド体は中間ノードを意味する．“Permutation 1” では，“10”, “5”, “11” は根ノードに割当可能である．しかし “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” の例では順列の順番に沿って割当可能である．

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として予備実験の結果に基づいて CX (Cyclic Crossover) 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．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を生成する．世代交代モデルとしては定常状態 GA 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 の兄弟ノードの子供であるノード v_j^{l+1} と入れ替える．

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 [KDDI 04] は 24 個のキーが図 3 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．先ず，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．

図 3: 対象とした携帯電話のキー配列

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 [Amant 04] 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 [Silfverberg 00][*1]．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

対象とした端末での探索・意思決定時間を求めることは困難であること，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の熟練者を想定した次の式を用いた．$t_i^{sd} = 80\log_2(n^l) + 240$ (ms)[Cockburn 07][*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．

機能の類似度に用いるキーワードは，利用者マニュアル [KDDI 04] 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．また簡単化のために，ノードにおける選択・意思決定時間では，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:「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．

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

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

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

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，$Imp = 100 \times (T_o - T_m)/T_o$ である．

表から分かるように提案手法により平均選択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，平均選択時間が短くなることがあるが，これは，探索・意思決定時間が，ノードに割当られた項目数 n にたいして，$\log_2(n)$ に比例する形で増加するためである．最大項目数を 6 まで小さくすると，平均選択時間は長くなるが，ペナルティ項の値は小さくなる．

4.2.1 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 とすることで平均選択時間は短くなるが，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．

4.2.2 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な平均選択時間の短縮を可能としている．

4.2.3 係数の影響

ポインティング時間，探索・意思決定時間は，実験データを回帰して得られる 4 個の係数 $(a^p, b^p, a^{sd}, b^{sd})$ に依存する．係数の変化で時間短縮効果がどのように変化するか調べた結果が表 4 である．これから分かるように，係数の組合せに依らず 40%以上の時間短縮が可能となっている．

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

	Mail2				Mail3				Web2				Web3			
Case	T_{ave}	(%)	P^s	P^g	T_{ave}	(%)	P^s	P^g	T_{ave}	(%)	P^s	P^g	T_{ave}	(%)	P^s	P^g
Orig. 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
$W = 16$	1969	40.9	670	1265	1918	42.4	689	1262	2079	37.6	679	1264	2140	35.8	696	1262
$W = 12$	1894	43.1	486	865	1834	44.9	495	848	2046	38.6	490	875	2099	37.0	487	875
$W = 9$	1888	43.3	352	297	1839	44.8	358	302	2017	39.4	354	295	2046	38.6	349	295
$W = 6$	2154	35.3	232	183	2122	36.3	232	175	2291	31.2	229	180	2310	30.7	229	178

表 4: 推定式の係数の違いによる性能変化.

coefficients				T_{ave}(ms)		Imp.(%)	
a^p	b^p	a^{sd}	b^{sd}	ave	min	ave	max
50	170	50	180	1535	1501	41.5	42.8
60	90	80	120	1280	1239	39.0	40.9
60	190	60	200	1744	1705	41.5	42.8
66	210	88	264	2120	2081	41.8	42.9
80	220	50	180	1824	1793	41.8	42.8
80	220	100	250	2246	2195	41.5	42.9

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 [Ahlström 05, Beck 06, Findlater 04] 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．

参考文献

[Ahlström 05] Ahlström, D.: Modeling and improving selection in cascading pull-down menus using Fitts' law, the steering law and force fields, in *Proc. of CHI2005*, pp. 61–70 (2005)

[Amant 04] Amant, R. S., Horton, T. E., and Ritter, F. E.: Model-based evaluation of cell phone menu interaction, in *Proc. of CHI2004*, pp. 343–350 (2004)

[Beck 06] Beck, J., Han, S. H., and Park, J.: Presenting a Sub-menu Window for Menu Search on a Cellular Phone, *IJHCI*, Vol. 20, No. 3, pp. 233–245 (2006)

[Cockburn 07] Cockburn, A., Gutwin, C., and Greenberg, S.: A predictive model of menu performance, in *Proc. of CHI2007*, pp. 627–636 (2007)

[Schultz 86] Schultz, E. E. and Curran, P. S.: Menu structure and ordering of menu selection: independent or interactive effects?, *SIGCHI Bull.*, Vol. 18, No. 2, pp. 69–71 (1986)

[Findlater 04] Findlater, L. and McGrenere, J.: A comparison of static, adaptive, and adaptable menus, in *Proc. of CHI2004*, pp. 89–96 (2004)

[Francis 00] Francis, G.: Designing Multifunction Displays: An Optimization Approach, *Int. J. of Cog. Ergo.*, Vol. 4, No. 2, pp. 107–124 (2000)

[KDDI 04] KDDI, : Maunal for CASIO W43CA (2004), http://www.au.kddi.com/torisetsu/pdf/w43ca/w43ca_torisetsu.pdf

[Kiger 84] Kiger, J. I.: The depth/breadth trade-off in the design of menu-driven user interfaces, *J. Int. J. Man-Mach. Stud.*, Vol. 20, No. 2, pp. 201–213 (1984)

[Matsui 08] Matsui, S. and Yamada, S.: Genetic algorithm can optimize hierarchical menus, in *Proc. of CHI2008* (2008), to appear

[Silfverberg 00] Silfverberg, M., MacKenzie, I. S., and Korhonen, P.: Predicting text entry speed on mobile phones, in *Proc. of CHI2000*, pp. 9–16 (2000)

[Toms 01] Toms, M. L., Cummings-Hill, M. A., Curry, D. G., and Cone, S. M.: Using cluster anyalysis for deriving menu structures for automotive mobile multimedia applications, Technical Report 2001-01-0359, SAE (2001)

[Zaphiris 03] Zaphiris, P., Kurniawan, S. H., and Ellis, R. D.: Age related difference and the depth vs. breadth tradeoffs in hierarchical online information systems, in *Proc. User Interfaces for All*, pp. 23–42 (2003), LNCS 2615