

# 遺伝的アルゴリズムを用いた 階層メニューの最適化

松井 正一

博士（情報学）

総合研究大学院大学  
複合科学研究科  
情報学専攻

平成 20 年度  
(2008)

2009 年 3 月

本論文は総合研究大学院大学複合科学研究科情報学専攻に  
博士（情報学）授与の要件として提出した博士論文である。

## 審査委員会

山田 誠二 (主査)	総合研究大学院大学 / 国立情報学研究所
相澤 彰子	国立情報学研究所
相原 健郎	総合研究大学院大学 / 国立情報学研究所
佐藤 健	総合研究大学院大学 / 国立情報学研究所
古山 宣洋	総合研究大学院大学 / 国立情報学研究所 (主査以外は 50 音順)

# Optimization of Hierarchical Menus with Genetic Algorithm

**Shouichi Matsui**

DOCTOR OF  
PHILOSOPHY

Department of Informatics  
School of Multidisciplinary Science  
The Graduate University for Advanced Studies (SOKENDAI)

March, 2009

A dissertation submitted to the Department of Informatics,  
School of Multidisciplinary Sciences,  
The Graduate University for Advanced Studies (SOKENDAI)  
in partial fulfillment of the requirements for  
the degree of Doctor of Philosophy

## Advisory Committee

Seiji Yamada (Chair)	National Institute of Informatics/ The Graduate University for Advanced Studies
Kenro Aihara	National Institute of Informatics/ The Graduate University for Advanced Studies
Akiko Aizawa	National Institute of Informatics
Nobuhiro Furuyama	National Institute of Informatics/ The Graduate University for Advanced Studies
Ken Sato	National Institute of Informatics/ The Graduate University for Advanced Studies

(Alphabetical order of last name except chair)

## 内容梗概

階層メニューは、GUIでコマンドを指定する目的でオフィスアプリケーション、携帯電話、Webアプリケーションなどの様々な分野で広く用いられており、機能が配置されたメニュー項目と下位のメニューを持つメニュー項目が複数配置された構造を持つ。階層メニューの性能としては様々なものが考えられるが、記憶のしやすさなどの使い勝手を考慮しつつ、目的とする機能にできるだけ早く到達できることが最も基本であり、メニューの性能評価で一般的に用いられている。本論文も、この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。この性能は、メニューの構造、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。また、単一階層のメニューの最適化についてはいくつかの研究が行われてきたが、階層メニューの構造を変更することで、性能を向上する最適化問題としてとらえた研究はほとんどない。

単一階層のメニューの最適化に関しては、Liuらは視覚探索のモデルをメニュー設計に適用し、平均到達時間を最小化する設計を試みている。階層メニューに関しては、Amantらは利用頻度の高い項目を、メニュー階層の浅い位置に上げていくという単純な手法でも、30%程度の時間削減が可能なことを示している。これらの研究において最適化問題として扱われている設計問題は、単一階層のメニューのみであり、また階層メニューの設計問題では使い勝手を明示的に考慮していない。これに対して、本論文では、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、かつ使い勝手を表す指標も目的関数に採り入れて、平均到達時間と使い勝手を表す指標の加重和を最小化する問題として、階層メニューの最適化を定式化する。この定式化では、ユーザが理想的な行動をした場合の平均到達時間を考える。すなわち、ユーザは目的の項目のあるサブメニューまで誤りなく辿り着くものとし、サブメニュー内の項目を誤りなく選択するものと仮定する。

メニュー項目への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。メニューの設計問題は、機能に対応したノード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問題となる。ただし、使いやすさの観点からは、効率だけを考えた項目の配置は望ましくなく、項目の意味を考慮する必要がある。また、サブメニュー毎の項目数が大きく異なるメニューでは、メニューの粒度が異なるため、使い勝手を損ねると考えられる。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。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．

また，ポインティング時間は Fitts の法則に従うものとする．探索・意思決定時間は探索時間と意思決定時間からなり，熟練者は項目の配置を記憶しており探索時間を 0 と仮定できることから，Hick-Hyman の法則によって意思決定時間をモデル化する．初心者では，探索時間を無視できないことから，項目数に比例する時間がかかるモデルを採用する．

先行研究によれば，階層メニューでは 1 ノードあたりの項目数が少なく，最大レベルが大きなメニュー（深いメニュー）よりも，1 ノードあたりの項目数が多く，最大レベルが小さなメニュー（浅いメニュー）の方が，平均到達時間が短いと報告されている．そこで，提案方式では浅いメニューが生成されるような解法を用いる．

対象とする問題は，非常に複雑な組み合わせ最適化問題であり，整数計画法などの厳密解法では，解を求めるために項目数の階乗に比例する時間が必要であり，現実的な時間では解を求めることができない．そこで，良い近似解が短時間で求まる解法を検討した．提案する解法は，遺伝的アルゴリズムに基づくものである．提案方式では浅いメニューが生成されるように，ノードに項目を割り当てていく割当順序を求める方式を採用した．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方式を提案した．

単一階層のメニューに対しては既存研究でその妥当性，近似精度が検証されているが，複数階層での検証は行われていないことから，モデル化の妥当性，近似精度を検証するために，PDA を用いた被験者実験を行った．19 個のメニュー項目からなる小規模なポップアップ型のメニューを PDA 上に表示し，10 名の被験者により平均到達時間を測定した．この結果，モデルによる平均到達時間の予測値 47% に対して，10 名の平均は 43% となり，提案するモデル化が十分な精度を持つことが確認できた．

本論文では，ユーザが理想的な行動をした場合を前提とすることから，被験者実験ではなく，コンピュータシミュレーションにより，提案手法の有効性を確認した．携帯電話のメニューを対象として，利用パターン，モデルで用いるパラメータを様々な値に設定した実験を行い，提案手法により平均到達時間を 40% 以上短縮できるメニュー構造を生成できることを確認した．また，Zipf 分布を用いて多様な利用パターンを生成し，多様な利用パターンに対しても，提案手法は平均到達時間を 40% から 60% 程度短縮できることを示し，利用パターンに対する頑強性を確認した．

以上により，本論文では，従来最適化問題として扱われてこなかった階層メニューの設計問題を，平均到達時間と使い勝手を表す指標の加重和を最小化する問題として新たに定式化し，遺伝的アルゴリズムを用いた新しい解法を提案し，モデルの精度を被験者実験で確認し，さらに多様な利用パターンに対して提案手法が有効であることを示した．

# Abstract

Hierarchical menus are one of the primary controls for issuing commands in GUIs. The performance of a menu can be measured using many metrics, but the essential measure needs to reflect the time it takes to reach a desired target. Therefore, we used the average selection time to reach targets as a performance index. The performance of a menu depends on many factors, such as the structure, layout, and colors. There have been many studies on novel hierarchical menus, such as split menu, adaptive/adaptable menu, and so on. But there has been little work on improving the performance of a hierarchical menu by changing its structure.

Liu et al. used a visual search model to a single level menu design. They used an optimization algorithm to minimize the search times. Quiroz et al. used an interactive genetic algorithm (IGA) with the user interfaces defined by XUL, but this was limited to single level menus. Amant et al. presented the concepts to support the analysis of cell phone menu hierarchies. They also tried to improve the menu selection time by using a simple best-first search algorithm, and got over 30% savings in selection time.

First, we develop a formulation to deal with the menu design problem as an optimization problem. The optimization problem of hierarchical menus is considered one that deals with placing menu items on the nodes of a tree. The time to select the target item is the time to traverse from the root to the target node. The problem is to minimize the average selection time with respect to the given search frequencies of different menu items.

We must traverse the tree from the root to the target node to select the desired function. Search/decision is necessary for the nodes, and pointing is necessary to select an item on a node. Therefore, traverse time from the root to a node is expressed, using the search/decision time and the pointing time. The pointing time is modeled using Fitts' law, and the search/decision time is modeled using Hick-Hyman law for an expert, and linear search for a novice.

We introduce two metrics: *functional similarity* and *menu granularity*, to cope with the difficulties of representing and reasoning about the menu item semantics. Functional similarity is a metric that represents the similarity between two menu items in terms of their functions. We use this metric to avoid placing items with little similarity in the same submenu of a node. If items with little similarity are put in the same submenu, it is difficult for a user to remember the menu layout. Menu granularity is a metric that reflects the number of submenus a node has as its descendants. We introduce this metric to avoid placing an item that has many children and an item that has no child as children of the same node.

Due to the usability factors, the optimization as a menu design is so very hard that mathematical programming can not solve it. Thus, we apply GA to automatically optimize the

menu design problem. Since the previous studies showed that breadth was preferable to depth, we used a kind of breadth-first search algorithm as the core of our proposed GA. An algorithm that places menu items one by one on a usable node can find a good solution. If we have a sufficient number of intermediate nodes, we can search for enough space to find an optimal solution.

The accuracy of the models for hierarchical menus with plural levels has not yet been verified. Therefore, we conducted experiments using a PDA to verify the model in hierarchical menus with plural levels. We measured the average selection time using small hierarchical menus that were generated for inexperienced users. There were some participants whose reduction ratios were lower than expected, but there also were some participants whose reduction ratios were higher than expected. The reduction ratios agreed well with the computed one, and we concluded that the model and the algorithm can be used to optimize hierarchical menus with plural levels.

We experimented on the static hierarchical menu of a cellular phone where a small screen and limited input device are assumed by using a wide variety of usage patterns to confirm the effectiveness of the algorithm. We gathered usage frequency data by recording the daily usage of each function for two months, and we generated the usage frequency data from that record. The record contained the functions that were built into the phone, and the E-mail folders and Web URLs that were added by the user. There were 129 terminal nodes, including the E-mail folders and Web URLs, in the data. We also generated the additional datasets that model different usage patterns. We also tested the robustness of the proposed algorithm by generating usage patterns using Zipf's law.

By the experiments, we concluded that the proposed algorithm can generate better menus than a forward movement. The difference in reduction ratio between the proposed algorithm and a forward movement is at least 20 points and even 40 points in many cases. The variance in the reduction ratio of the proposed algorithm is small; the standard deviation is less than 3.5 points.

The results showed that the algorithms can generate a better menu structure. They were able to reduce the average selection time by over 40% for a wide variety of usage patterns.



# 目次

<b>第1章</b>	<b>序論</b>	<b>1</b>
1.1	メニュー，階層メニュー	1
1.2	メニュー設計で考慮すべき点	3
1.3	メニューの最適化の必要性	4
1.4	本研究の目的	6
1.5	本論文の構成	8
<b>第2章</b>	<b>関連研究</b>	<b>9</b>
2.1	階層メニュー	9
2.2	インタラクションのモデル	13
2.2.1	ポインティング時間	13
2.2.2	意思決定・選択時間	16
2.2.3	ユーザのモデル化	17
2.3	ユーザビリティの評価	20
2.4	メニューの最適化	24
2.4.1	浅いメニューあるいは深いメニュー	24
2.4.2	単一階層メニューの最適化	29
2.4.3	階層メニューの最適化	32
2.4.4	メニューの使い勝手の向上方策	34
2.5	ユーザインタフェースの最適化	37
2.6	遺伝的アルゴリズム	41
2.6.1	遺伝的アルゴリズムの基本構造	41
2.6.2	選択交配	42
2.6.3	交叉	43
2.6.4	突然変異	44
2.6.5	世代モデル	44
<b>第3章</b>	<b>階層メニューの最適化問題</b>	<b>45</b>
3.1	定式化の概要	45
3.2	標記法	46
3.3	到達時間	47

---

3.4	ポインティング時間と探索・意思決定時間 . . . . .	47
3.5	機能の類似度 . . . . .	48
3.6	メニューの粒度 . . . . .	49
3.7	最適化問題 . . . . .	50
3.8	まとめ . . . . .	51
<b>第4章</b>	<b>遺伝的アルゴリズムを用いた解法</b>	<b>53</b>
4.1	染色体表現 . . . . .	53
4.1.1	基本的な方式 . . . . .	53
4.1.2	割当順序表現 . . . . .	54
4.1.3	表現の完全性 . . . . .	55
4.2	局所探索 . . . . .	55
4.3	交叉の方式 . . . . .	56
4.4	他のGAパラメータ . . . . .	58
4.5	他の表現方式によるGAの構成法 . . . . .	58
4.6	まとめ . . . . .	58
<b>第5章</b>	<b>被験者実験によるモデルの検証</b>	<b>59</b>
5.1	実験方法 . . . . .	59
5.1.1	メニューの生成 . . . . .	59
5.1.2	被験者実験の方法 . . . . .	60
5.2	実験結果 . . . . .	61
5.3	まとめ . . . . .	62
<b>第6章</b>	<b>シミュレーションによる最適化の評価</b>	<b>63</b>
6.1	実験に用いたデータ . . . . .	63
6.1.1	ポインティング時間 . . . . .	63
6.1.2	探索・意思決定時間 . . . . .	64
6.1.3	利用頻度と機能の類似度 . . . . .	64
6.1.4	計算条件の設定 . . . . .	67
6.2	性能向上の実験結果 . . . . .	67
6.3	ペナルティ項の影響 . . . . .	77
6.4	推定式の係数の影響 . . . . .	80
6.5	様々な頻度データに対する性能 . . . . .	80
6.6	利用パターンに対する頑健性 . . . . .	84
6.6.1	利用パターンの生成 . . . . .	84
6.6.2	計算結果 . . . . .	85
6.7	計算時間，個体評価回数 . . . . .	92

---

6.8	まとめ	92
<b>第7章</b>	<b>考察</b>	<b>93</b>
7.1	提案手法の有効性と限界	93
7.1.1	提案手法の優位性	93
7.1.2	提案手法が有効でない場合	93
7.2	提案手法の適用範囲	94
7.2.1	静的メニューへの適用	94
7.2.2	適応メニューへの適用	94
7.2.3	適用のためには拡張が必要な事例	95
7.2.4	操作の系列を考慮した最適化	96
7.3	適応干渉	97
7.3.1	提案手法における適応干渉	97
7.3.2	提案手法の拡張の方向性	97
7.4	ユーザのエラーモデル	99
7.4.1	辿るべき階層の誤り	99
7.4.2	辿るべき階層の誤りのモデル化の方向性	99
7.4.3	ポインティングエラー	100
7.5	浅いメニューあるいは深いメニュー	101
7.6	SA に基づく解法との性能比較	102
7.6.1	SA に基づく解法	102
7.6.2	オリジナルデータでの比較結果	102
7.6.3	他のデータセットでの比較結果	103
7.7	解法の性能向上方策	106
<b>第8章</b>	<b>結論</b>	<b>107</b>
	謝辞	110
	参考文献	111
	研究業績	119
<b>付録A</b>	<b>被験者実験で用いたデータ</b>	<b>121</b>
<b>付録B</b>	<b>シミュレーション実験で用いたデータ</b>	<b>123</b>
B.1	想定した携帯電話での困難度の指標 ( <i>ID</i> )	123
B.2	メニュー構造と利用頻度	131



# 目 次

1-1	階層メニューの例 . . . . .	1
1-2	同時メニューの例 . . . . .	2
1-3	設計時と利用時の違い [Fischer 01] . . . . .	4
2-1	階層メニューの例 ( 図 1-1 の再掲 ) . . . . .	9
2-2	Windows のスタートメニューの例 . . . . .	10
2-3	Thunderbird でのメッセージの移動の例 . . . . .	11
2-4	Web での階層メニューの例 . . . . .	12
2-5	Fitts の法則 . . . . .	13
2-6	ポインティング対象が長方形以外の例 [Grossman 07] . . . . .	14
2-7	中心の計算方法 [Grossman 07] . . . . .	15
2-8	Dvorak 配置のソフトキーボードの例 [Mackenzie 99] . . . . .	16
2-9	ACT-R でのチャンク , プロダクションの例 [Anderson 97] . . . . .	18
2-10	ACT-R のメニュー選択問題でのプロダクションの例 [Anderson 97] . . . . .	18
2-11	John らのシステムの画面例 [John 04] . . . . .	19
2-12	Huang らの実験 [Huang 07] . . . . .	20
2-13	Huang らの実験 [Huang 07] . . . . .	21
2-14	SEM-CPU[Lee 06] . . . . .	22
2-15	Huang らの実験 [Ziefle 04] . . . . .	22
2-16	Kiger の実験でのメニューの例 (4 <sup>3</sup> )[Kiger 84] . . . . .	25
2-17	Zaphiris らの実験での画面例 [Zaphiris 03] . . . . .	27
2-18	Parush らの実験での画面例 [Parush 04] . . . . .	27
2-19	Parush らの実験での浅いナビゲーションの例 [Parush 04] . . . . .	27
2-20	Parush らの実験での深いナビゲーションの例 [Parush 04] . . . . .	28
2-21	対象とした端末 [Geven 06] . . . . .	28
2-22	提示したメニュー [Geven 06] . . . . .	29
2-23	Liu らの実験で用いられた画面連 [Liu 02] . . . . .	29
2-24	IGA によるメニューの最適化での染色体表現 [Quiroz 07] . . . . .	30
2-25	IGA によるメニューの最適化例 ( 初期個体 ) [Quiroz 07] . . . . .	31
2-26	IGA によるメニューの最適化例 ( 200 世代目 ) [Quiroz 07] . . . . .	31

2-27 Francis らの対象とする画面例 (ATM) [Francis 02]	32
2-28 GOMS による操作のモデル化例 [Amant 07]	33
2-29 Web でのメニュー構造 [Yu 02]	35
2-30 Web 画面例 [Yu 02]	35
2-31 Web 画面例 (プルダウン型) [Yu 02]	36
2-32 KeyMenu の模式図 [Lyons 03]	36
2-33 ユーザの特性に合わせたインタフェースの自動生成の枠組み [Gajos 08b]	37
2-34 ユーザのモデリングのためのインタフェースの例 [Gajos 08b]	38
2-35 SUPPLE++による生成例 [Gajos 08b]	38
2-36 Gajos らの実験の構成 [Gajos 08a]	39
2-37 携帯電話でのショートカットの効果 [Bridle 06]	40
2-38 遺伝的アルゴリズムの処理手順	41
2-39 単純交叉・1点交叉の例	43
2-40 2点交叉の例	43
2-41 一様交叉の例	43
3-1 階層メニューの木構造表現	46
4-1 順列から木構造への変換方式	54
4-2 局所探索	55
4-3 PMX の例	56
4-4 OX の例	57
4-5 CX の例	57
4-6 突然変異の方式	58
5-1 PDA での実験で用いたメニュー	60
5-2 PDA での実験の画面例 (初期画面)	61
5-3 PDA での実験の画面例 (メニュー遷移の例)	61
6-1 対象とした携帯電話のキー配列	66
6-2 計算結果のばらつき (平均到達時間)	68
6-3 計算結果のばらつき (短縮率)	69
6-4 計算結果のばらつき (機能の類似度のペナルティ)	69
6-5 計算結果のばらつき (メニューの粒度のペナルティ)	69
6-6 元のメニュー	71
6-7 前方移動方式で生成されたメニューの例	72
6-8 生成されたメニューの例 ( $W = 9$ )	73
6-9 生成されたメニューの例 ( $W = 6$ )	74

6-10	生成されたメニューの例 ( $W = 12$ )	75
6-11	生成されたメニューの例 ( $W = 16$ )	76
6-12	$(\alpha, \beta)$ 毎の計算結果のばらつき (平均到達時間)	77
6-13	$(\alpha, \beta)$ 毎の計算結果のばらつき (短縮率)	78
6-14	$(\alpha, \beta)$ 毎の計算結果のばらつき (機能の類似度のペナルティ)	78
6-15	$(\alpha, \beta)$ 毎の計算結果のばらつき (メニューの粒度のペナルティ)	78
6-16	ペナルティ項の重みを 0 として生成したメニューの例 ( $W = 9$ )	79
6-17	Mail2 データセットにおける計算結果のばらつき	81
6-18	Mail3 データセットにおける計算結果のばらつき	82
6-19	Web2 データセットにおける計算結果のばらつき	82
6-20	Web3 データセットにおける計算結果のばらつき	83
6-21	計測した頻度分布と Zipf 則での頻度分布の比較	85
6-22	短縮率のばらつき ( $D(1, 32)$ )	87
6-23	短縮率のばらつき ( $D(1, 64)$ )	87
6-24	短縮率のばらつき ( $D(1, 129)$ )	87
6-25	短縮率のばらつき ( $D(1.5, 32)$ )	89
6-26	短縮率のばらつき ( $D(1.5, 64)$ )	89
6-27	短縮率のばらつき ( $D(1.5, 129)$ )	89
6-28	短縮率のばらつき ( $D(2, 32)$ )	91
6-29	短縮率のばらつき ( $D(2, 64)$ )	91
6-30	短縮率のばらつき ( $D(2, 129)$ )	91
7-1	Morphing メニューの例 [Cockburn 07]	95
7-2	操作の系列の模式図	96
7-3	習熟の例 [Cockburn 07]	98
7-4	SA に基づく解法の概要	102
7-5	平均到達時間の比較 (オリジナル)	103
7-6	Mail2 データセットでの平均到達時間の比較	104
7-7	Mail3 データセットでの平均到達時間の比較	104
7-8	Web2 データセットでの平均到達時間の比較	105
7-9	Web3 データセットでの平均到達時間の比較	105
7-10	GA での収束過程の例 (交叉方式: CX, $W = 9$ )	106
B-1	対象とした携帯電話のキー配列 (図 6-1 の再掲)	123





# 表 目 次

1-1 ユーザインタフェースの定量的評価指標 . . . . .	3
1-2 Adaptive と Adaptable の比較 [Fischer 01] . . . . .	5
2-1 Geven らの実験でのメニューの構造 [Geven 06] . . . . .	28
2-2 Liu らの実験で用いられた画面構成 [Liu 02] . . . . .	30
2-3 スケーリング方法 . . . . .	42
3-1 探索空間の大きさの例 . . . . .	51
5-1 PDA での短縮率 . . . . .	62
6-1 28 個のグループ毎の ID の値 . . . . .	66
6-2 平均到達時間の短縮効果 . . . . .	68
6-3 ペナルティ項の影響 . . . . .	77
6-4 推定式の係数の違いによる性能変化 . . . . .	80
6-5 平均到達時間の短縮効果 (Mail2, Mail3, Web2, Web3) . . . . .	81
6-6 Zipf 則に従った頻度分布での平均選択時間の短縮率 (%) ( $D(1, *)$ ) . . . . .	86
6-7 Zipf 則に従った頻度分布での平均選択時間の短縮率 (%) ( $D(1.5, *)$ ) . . . . .	88
6-8 Zipf 則に従った頻度分布での平均選択時間の短縮率 (%) ( $D(2, *)$ ) . . . . .	90
A-1 PDA による被験者実験で用いたメニューの構造と項目の利用頻度 . . . . .	121
B-1 想定した携帯電話での困難度の指標 ( $ID$ ) (その 1) . . . . .	124
B-2 想定した携帯電話での困難度の指標 ( $ID$ ) (その 2) . . . . .	125
B-3 想定した携帯電話での困難度の指標 ( $ID$ ) (その 3) . . . . .	126
B-4 想定した携帯電話での困難度の指標 ( $ID$ ) (その 4) . . . . .	127
B-5 想定した携帯電話での困難度の指標 ( $ID$ ) (その 5) . . . . .	128
B-6 想定した携帯電話での困難度の指標 ( $ID$ ) (その 6) . . . . .	129
B-7 想定した携帯電話での困難度の指標 ( $ID$ ) (その 7) . . . . .	130
B-8 携帯電話のメニューの構造と項目の利用頻度 (その 1) . . . . .	131
B-9 携帯電話のメニューの構造と項目の利用頻度 (その 2) . . . . .	132
B-10 携帯電話のメニューの構造と項目の利用頻度 (その 3) . . . . .	133



# 第1章 序論

## 1.1 メニュー、階層メニュー

一般的には「メニュー」は「献立表」であり、レストランなどで提供可能な料理を一覧にしたものである。情報処理用語大辞典（オーム社；1992年）によれば、「文字表示装置やグラフィック表示装置などを用いて、コンピュータとユーザが会話するとき、ユーザが次に取り得る選択肢を表示したもの」とされている。一画面には表示できないほど選択肢が多い場合、選択肢をグループ化して表示する場合などでは、階層を持ったメニュー、すなわち「階層メニュー」の形が使われる。

階層メニューは、選択肢に直接対応するメニュー項目と下位のサブメニューに対応するメニュー項目が複数配置された構造を持つ（図1-1）。図1-1(a)携帯電話の例では、「1 Browse」から「7 Quit」が第1階層のメニューとして表示され、「3 Rotate」が選ばれることで、その右に「1 Automatic」から「4 Right」の項目を持つ下位のサブメニューが表示されている。

図1-1(b)メールソフトの例は、Windowsで標準的なメニューバーから始まるメニューの例である。「ファイル(F)」「編集(E)」「表示(V)」といったものが表示されたトップメニューから、「ファイル(F)」を選び「新規作成(N)」を選んだ状態である。次に選択可能な項目「メッセージ(M)」「フォルダ(F)」などが右側に表示されている。



(a) 携帯電話の例

(b) メールソフトの例

図 1-1 階層メニューの例

K.L. Norman による “The Psychology of Menu Selection — Designing Cognitive Control at the Human/Computer Interaction” [Norman 91] は 1991 年時点でのメニュー選択に関する研究成果をまとめた書籍である．ここではメニューの構造を大きく以下に分けている．

単一メニュー (single menu)：例えば，ゲームの難度を (a) 容易，(b) 普通，(c) 困難の中から選択するようなメニューである．

逐次線形メニュー (sequential linear menu)：例えば，アンケートの回答者の属性を，(1) 年代，(2) 性別，(3) PC 所有の有無，(4) 使えるプログラミング言語などについて順次聞いていくようなメニューである．

同時メニュー (simultaneous menu)：図 1-2 に示すような，同時に複数のオプションを提示するメニューである．図 1-2 は「端末エミュレーション」設定メニューであり，「様々な端末オプションの設定」，「行規則オプション」，「遠隔操作の印刷」の設定が表示されている．

階層メニュー (hierarchical menu)：階層的に提示されるメニューによって選択肢を順次確定していくものである．

階層メニューは日々の生活で至る所で使われている．例えば，銀行の ATM (Automated Teller Machine; 現金自動預け払い機) にも階層メニューが使われている．「現金引き出し」などを選ぶと，「暗証番号」の画面に変わり，次に「金額」を入力する画面に変わる．あるいは，多数の機能を持つ携帯音楽プレイヤーなどでも階層メニューが使われている．

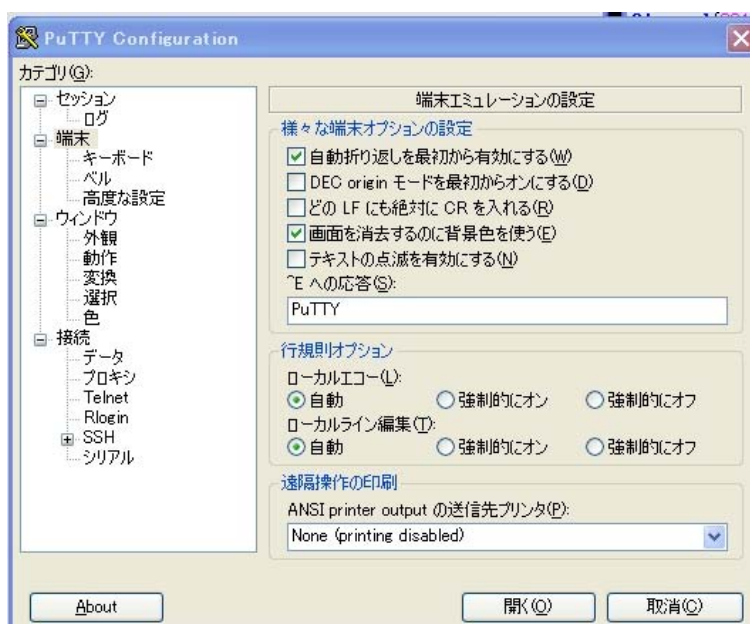


図 1-2 同時メニューの例

## 1.2 メニュー設計で考慮すべき点

ユーザインタフェースの定量的評価では，表 1-1 に示した 4 項目が使われることが一般的である [Lee 06]．この中の「有効さ」「効率」「満足度」は ISO9241-11 でのユーザビリティの定義にもなっている [ユー 07]．この定量的評価指標ができるだけ高いメニューを設計するためには様々な点を考慮する必要がある．

前述の K.L. Norman による書籍は，第 1 部「メニュー選択の理論と実装 (The Theory and Implementation of Menu Selection Systems)」，第 2 部「実験的研究に基づく設計ガイドライン (Design Guidelines from Empirical Research)」，第 3 部「メニューシステムの実装と評価 (Implementation and Evaluation of Menu Systems)」からなる．メニュー選択に関する定評のある書物であり，基本的な理論，実験的な結果が網羅されている．コンピュータの技術進歩によりメニューの表示方式などは変わっているが，内容は古びてはいないことから，設計ガイドラインは現在でも有益な情報を含んでいる．

第 2 部「実験的研究に基づく設計ガイドライン」には，以下のガイドラインが示されている．

- メニュー選択での研究項目と方法 (research issues and methods in menu selection)
- メニューのフォーマットと語法 (formatting and phrasing)
- 性能，獲得，訓練方法 (performance, acquisition, and training methods)
- 階層メニューにおける深さと幅の関係 (depth versus breadth of hierarchical menu trees)
- 階層メニューにおける探索行動 (search behavior in hierarchical menu structures)
- 迅速にアクセス可能なメニュー (rapid access menus)

上記の項目に本論文でどのように対応するかは，第 1.4 節「研究の目的」で述べる．

表 1-1 ユーザインタフェースの定量的評価指標

項目	定義	測定項目例
有効さ (effectiveness)	ユーザが，指定された目標を達成する上での正確さと完全さ	タスク完了率 (%)
効率 (efficiency)	ユーザが，目標を達成する際に正確さと完全さに費やした資源	タスク完了時間 (s)，エラー率 (%)
満足度 (satisfaction)	不快さのないこと，および製品使用に対しての肯定的な態度	満足度の評点
学習容易性	短時間で効率的に学習できること	タスクを繰り返した場合のタスク完了時間 (s) と完了率 (%)

### 1.3 メニューの最適化の必要性

Fischer はヒューマンコンピュータインタラクションで用いることのできるユーザモデルのサーベイを行っている [Fischer 01] . ここで「設計時」と「利用時」のギャップについての問題点を明確にしている (図 1-3) . 図 1-3 に示すように , ユーザインタフェースの設計時には「典型的なユーザ」を想定した設計が行われる . しかし , この「典型的なユーザ」は全てのユーザを含むものではないため , 前述の定量的評価指標はユーザにより異なることになる . 特に , 目的とする機能に辿り着くまでの時間は , 階層を辿るための時間が支配的であり , また , ユーザ毎に機能の利用頻度は大きく異なることから , この時間がユーザ毎に大きく異なるものとなる .

このために , adaptive あるいは adaptable ユーザインタフェースが提供されることが多い . 彼はまた , adaptive と adaptable の違いについて明確にしている (表 1-2) . 本論文では adaptable なインタフェースを前提に , 与えられた階層メニューの構造を変更することで , 目的とする機能に辿り着くまでの時間を平均的に最小化する問題を考える .

前述のユーザインタフェースの定量的評価のための 4 指標の中でも , 「効率性」 , すなわち目的とする機能にできるだけ早く到達できることが最も基本とされており , メニューの性能評価で一般的に用いられている [Amant 07, Findlater 04, Liu 02, Quiroz 07] . 本論文も , この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 .

この性能は , メニューの構造 , レイアウト , 色などの多くの要因によって決まる . これらの要因を変更することで , 性能の高いインタフェースを提供するために , 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 [Findlater 04] . また , 単一階層のメニューの最適化についてはいくつかの研究が行われてきたが [Liu 02, Quiroz 07] , 階層メニューの構造を変更することで , 性能を向上する最適化問題としてとらえた研究はほとんどない .

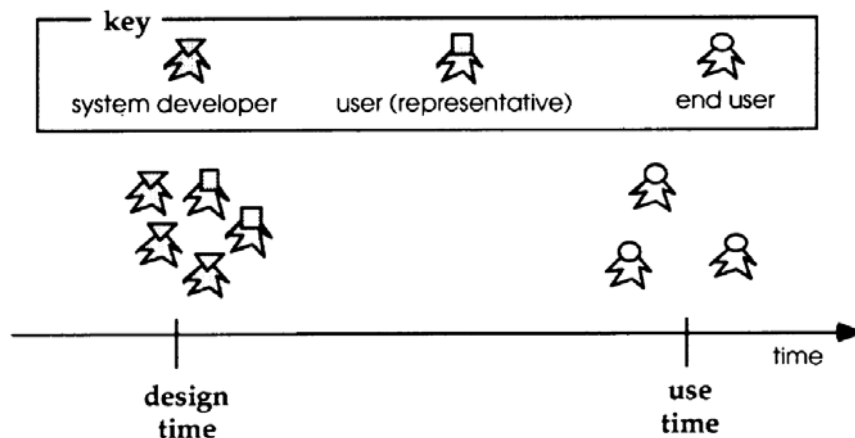


図 1-3 設計時と利用時の違い [Fischer 01]

表 1-2 Adaptive と Adaptable の比較 [Fischer 01]

	Adaptive	Adaptable
定義	現在のタスク，ユーザへシステムが動的に適応する	システムの機能を（システムの支援を受けて）ユーザが変更する
知識	システム内にあるものが多様な形で表出される	拡張可能
強み	ユーザの負荷はほとんどない，ユーザに関する特別な知識は不要	タスクはユーザが一番知っている，ユーザが制御可能，成功例あり
弱み	システムに関してユーザが一貫しモデルを獲得しにくい，制御できない，成功例が少ない	ユーザの負荷は高い，システムの互換性が損なわれる，適合させるために学ぶ必要があり，複雑になる
必要機能	ユーザ・タスク・対話のモデル，ゴールと計画の知識，強力なマッチング機能，逐次的なモデルの更新機能など	レイヤを持ったアーキテクチャ，ドメインモデル，システムからのフィードバックなど
適用対象	能動的ヘルプ，情報検索，ユーザインタフェースのカスタマイゼーションなど	情報検索，エンドユーザによる変更可能性，フィルタリングなど

先行研究については，第2章「関連研究」でさらに詳しく述べるが，概要は以下である．単一階層のメニューの最適化に関しては以下の報告がある．Liu らは視覚探索のモデルをメニュー設計に適用し，Guided Search によって，平均到達時間を最小化する設計を試みている [Liu 02]．Quiroz らは XUL で記述された単一階層のメニューを対話的 GA で最適化する方式を報告している [Quiroz 07]．これらの研究では，本論文では対象としていない，視覚探索を効率化する「色」も最適化の対象としているが，単一階層のメニューの最適化にとどまっている．

階層メニューに関しては，Amant らが携帯電話のメニュー構造を分析するための方法を提案として，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いる．実験の結果として，GOMS モデルによる予測が最も適しており，利用頻度の高い項目を，メニュー階層の浅い位置に上げていくという単純な手法でも，30%程度の時間削減が可能なことを示している [Amant 07]．ただし，この研究では，機能のまとまりを保存する意味から，浅い位置に項目を移動するのみであり，ショートカット機能が利用できる場合でも，そこに移動させることはできない．

また，Francis らは多機能ディスプレイの表示方式（階層メニューの一種とみなすこともできる）の最適化を報告している [Francis 00]．この研究では，階層構造は予め与

えられるものとして、階層内の項目配置の最適化のみを行っている。前述のように、先行研究において最適化問題として扱われている設計問題は、単一階層のメニューのみであり、また階層メニューの設計問題では使い勝手を明示的に考慮していない。

これに対して、本論文では、従来最適化問題として扱われてこなかった階層メニューの設計問題を、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、かつ使い勝手を表す指標も目的関数に採り入れて、平均到達時間と使い勝手を表す指標の加重和を最小化する問題として定式化する。さらに、この問題の解法を提案し、表示画面と入力方式が限定された携帯電話とPDAを対象に提案手法の有効性を示す。

## 1.4 本研究の目的

本研究の目的は、adaptableなインタフェースを前提に、階層メニューの最適化問題を定式化し、解法を提案し、提案手法の有効性を示すことである。このために、従来最適化問題として扱われてこなかった階層メニューの設計問題を、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、かつ使い勝手を表す指標も目的関数に採り入れて、平均到達時間と使い勝手を表す指標の加重和を最小化する問題として定式化する。対象とする問題は、非常に複雑な組み合わせ最適化問題であり、整数計画法などの厳密解法では、解を求めるために項目数の階乗に比例する時間が必要であり、現実的な時間では解を求めることができない。そこで、良い近似解が短時間で求まる遺伝的アルゴリズムに基づく解法を提案する。さらに、定式化で用いたモデルの近似精度を被験者実験により確認し、表示画面と入力方式が限定された携帯電話とPDAを対象に提案手法の有効性を示す。

定式化では、ユーザが理想的な行動をした場合の平均到達時間を考える。すなわち、ユーザは目的とする項目を誤りなく選択することを前提とする。つまり、ユーザは目的の項目のあるサブメニューまで誤りなく辿り着くものとし、サブメニュー内の項目を誤りなく選択するものと仮定する。この仮定の下で、ユーザの利用頻度に従って、平均到達時間を最小化するメニューを設計する問題として定式化する。

ポインティング時間はFittsの法則の法則に従うものとして定式化する。探索・意思決定時間は探索時間と意思決定時間からなり、(1)熟練者は項目の配置を記憶していることから、探索時間を0と仮定できることから、Hick-Hymanの法則によって意思決定時間をモデル化する。(2)初心者では、探索時間を無視できないことから、項目数に比例する時間がかかるモデルを採用する。

使い勝手を表す指標として「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。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。

前述のように、インタフェースの定量的評価指標には「有効さ」「効率」「満足度」「学習容易性」の4指標がある。本研究で提案する最適化手法では平均到達時間を短縮することにより「効率」の向上が得られると考える。また、目的関数に採り入れた「機能の類似度」と「メニューの粒度」により「学習容易性」も向上できると考える。提案手法により生成されるメニュー構造は、利用頻度の高い項目は浅い階層に配置されることから「有効さ」と「満足度」も間接的に向上できると考えられる。

Norman が示している設計ガイドラインの項目で、設計に関わる部分は以下の形で対応していることになる。

- メニューのフォーマットと語法：adaptable を前提とすることから、初期設計で熟慮されているものとして扱う。
- 性能、獲得、訓練方法：本論文の第一の目的は性能の向上にあり、前述のように、「機能の類似度」と「メニューの粒度」により、獲得、訓練も容易になると考えられる。
- 階層メニューにおける深さと幅の関係：幅を変えた複数のメニューを生成し、平均到達時間が最小となる方式を採用することとする。
- 迅速にアクセス可能なメニュー：基本的には固定の階層メニューを対象とし、補助機能による迅速化は直接対象としない。

問題の定式化で採用したモデル化の妥当性、モデルの近似精度を、PDA 上に構築した小規模なメニューを対象とした被験者実験で確認する。

メニューの性能比較は、被験者実験で行われることが一般的である。これは定量的評価指標の4指標を検討するため、またユーザはエラーを起こすことを前提としているためである。これに対して、本論文では、ユーザは誤りなく選択することを前提することから、提案手法の有効性はシミュレーション実験で確認する。このために、実測した利用頻度とそれを基に生成した利用パターン、また Zipf 分布に従って生成した多様な利用パターンを用いて、シミュレーション実験により、提案手法の有効性を確認する。

## 1.5 本論文の構成

本論文は，本章も含めて8章からなる．

第2章「関連研究」では，階層メニューの汎用性，重要性を述べた後で，本研究と関連する既存研究を概観し，本研究の特色を示す．

第3章「階層メニューの最適化」では，階層メニューの最適化問題の定式化について述べる．従来最適化問題として扱われてこなかった階層メニューの設計問題を，平均到達時間を最小化する最適化問題として定式化する．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，かつ使い勝手の制約を考慮して，平均到達時間を最小化する問題としてメニューの設計問題を定式化する．

第4章「遺伝的アルゴリズムを用いた解法」では，階層メニューの最適化手法について述べる．対象とする最適化問題は，非常に複雑な組み合わせ最適化問題となることから，近似解を効率的に求めるための，遺伝的アルゴリズムを用いた解法について述べる．

第5章「被験者実験によるモデルの検証」では，定式化で採用したモデル化の妥当性，モデルの近似精度の検証結果について述べる．単一階層のメニューに対しては既存研究でその妥当性，近似精度が検証されているが，複数階層での検証は行われていないことから，小規模なメニューをPDAに表示させて，平均到達時間を計測した，被験者実験により，妥当性，近似精度を検証した結果を述べる．

第6章「シミュレーションによる最適化の評価」では，提案手法による平均到達時間の短縮効果の確認，すなわち提案手法の有効性の検証結果について述べる．本論文では，ユーザは目的とする項目を誤りなく選択することを前提とすることから，被験者実験ではなく，コンピュータシミュレーションにより，提案手法の有効性を確認した結果を述べる．具体的には，携帯電話端末を対象に，様々な利用パターンを想定したシミュレーション実験により，提案手法の有効性を検証した結果について述べる．

第7章「考察」では，提案手法の有効性と適用可能領域について述べた後，適応干渉との関係，提案手法では明示的に考慮していないユーザのエラーモデルについて議論する．また，浅いメニューと深いメニューの関連について述べる．最後に，GAに基づく提案手法の性能を確認するために行った，シミュレーテッドアニーリングに基づく解法，交叉方式を変えたGAに基づく解法と比較した結果，解法の性能改善の方向性について述べる．

第8章「結論」では，本研究での結論および課題について述べる．

## 第2章 関連研究

本章では、関連研究について概説する。まず、階層メニューについて、多様な利用形態を例示する。次に、インタラクションに関連するモデル化について概説する。その後、ユーザビリティに関連した研究について概説する。ついで、メニューの最適化に関連する研究について概説する。さらに、ユーザインタフェースの最適化に関連した研究について概説する。最後に、提案手法で用いている遺伝的アルゴリズムの概要と関連研究について概説する。

### 2.1 階層メニュー

階層メニューは、GUIでコマンドを指定する目的でオフィスアプリケーション、携帯電話、Webアプリケーションなどの様々な分野で広く用いられている [Norman 91, Helander 97]。階層メニューは機能が配置されたメニュー項目と下位のメニューを持つメニュー項目が複数配置された構造を持つ。

前章で示した携帯電話のメニューとWindowsのアプリケーションであるメールソフトの例(図2-1)以外にも、様々な場面で階層メニューは使われている。例えば、Windows



(a) 携帯電話の例

(b) メールソフトの例

図 2-1 階層メニューの例 (図 1-1 の再掲)

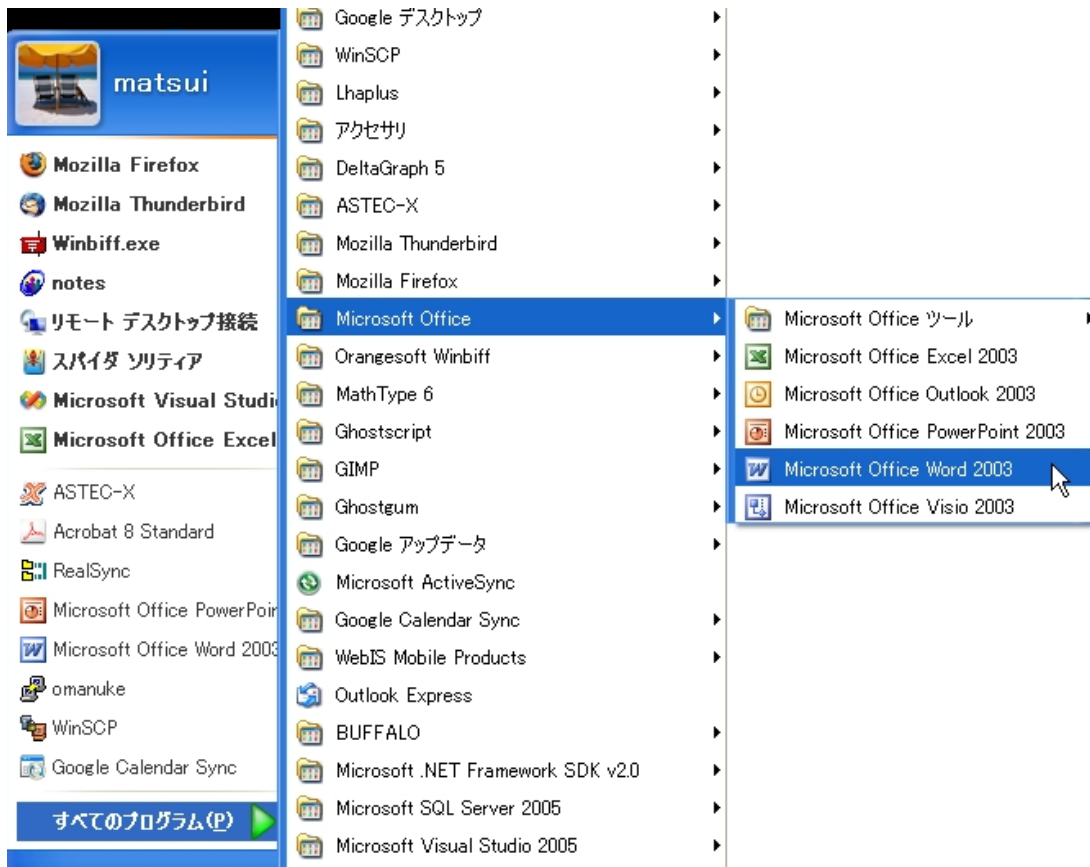


図 2-2 Windows のスタートメニューの例

のスタートメニューも階層メニューとなっている（図 2-2）。図は Windows XP のスタートメニューの例であり、「すべてのプログラム」をクリックし、さらに「Microsoft Office」をクリックした段階の例である。

利用頻度の高いプログラムを、スタートメニューをクリックして表示される部分に配置すれば、起動に必要なクリック回数とマウスの移動距離を少なくできる。逆に「すべてのプログラム」をクリックした場合に、同時に表示される項目数を少なくするためには、フォルダの数を少なくし、階層を深くしたメニューを構成すればよい。

アプリケーションのコンテキストメニューとして現れるものも階層メニューとなっている（図 2-3）。図 2-3 に示したのは、Thunderbird で特定のメッセージ一覧ページにて右クリックで表示されるコンテキストメニューの例である。図は「メッセージの移動」を選択して移動先を選ぼうとしている段階である。この場合でもフォルダを適切に配置することで、マウスの移動距離とクリック数を減少させることができる。

近年は、多くのシステムが Web ベースのインタフェースを採用しており、階層型情報提示方式も階層メニューとして考えることができる（図 2-4）。

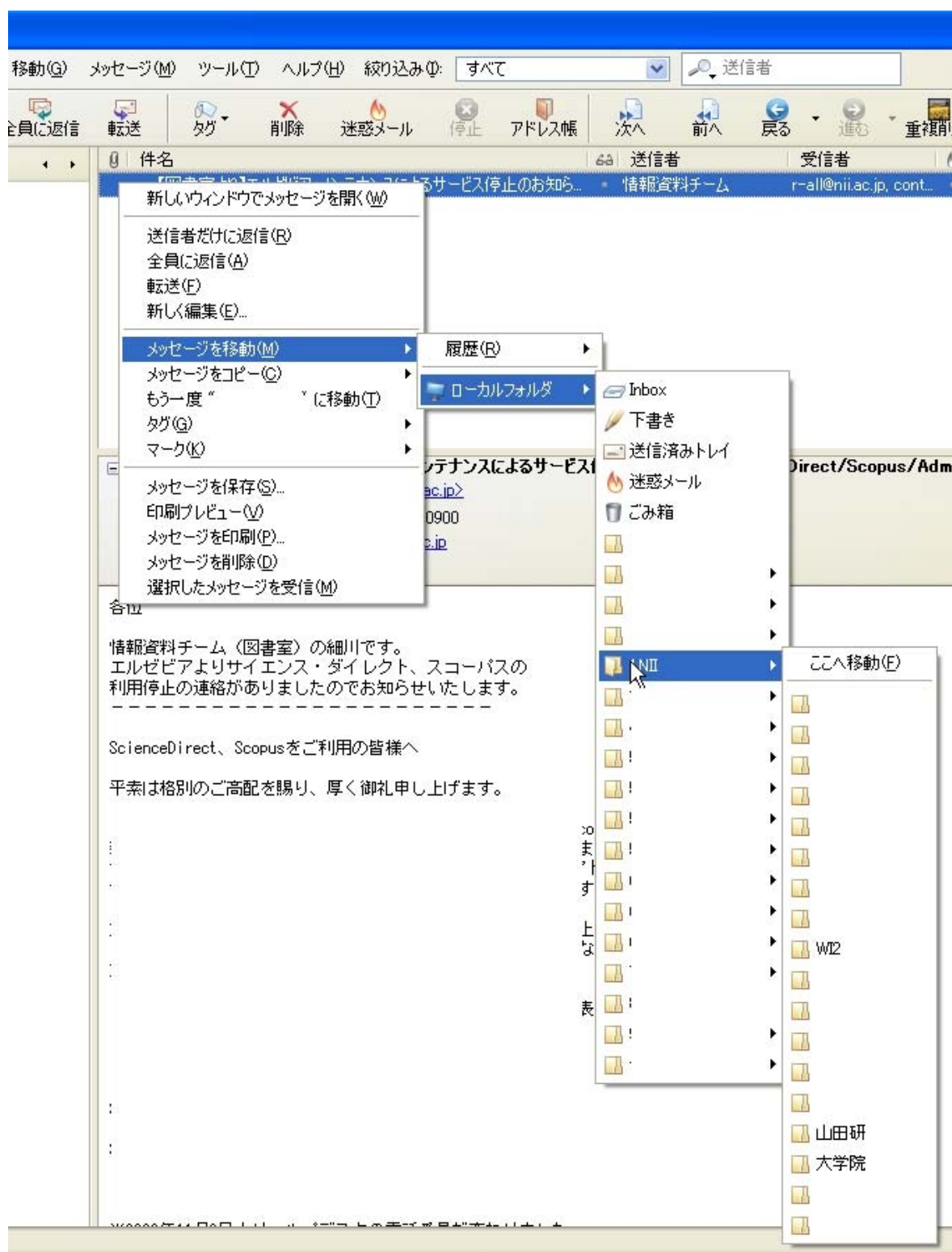


図 2-3 Thunderbird でのメッセージの移動の例



(a) NII のトップページ



(b) NII の「大学院教育」にカーソルを置いた状態



(c) NII の指導教員一覧を選んだ後

図 2-4 Web での階層メニューの例



## 2.2 インタラクションのモデル

ユーザインタフェースあるいはユーザとコンピュータのインタラクションのモデル化に関連した研究について概説する．インタラクションのモデルは、「予測モデル」と「記述モデル」に大別できる [MacKenzie 03]．本論文ではメニュー構造の変更を行うことで最適化する手法を提案することから，主に予測モデルを紹介する．

まず，対象をポインティングするために要する時間のモデル，次に，選択肢からの選択に要する意思決定・選択時間のモデルを紹介する．最後に，ユーザのモデル化に関連した研究について紹介する．

### 2.2.1 ポインティング時間

ディスプレイ上でターゲットの位置までマウスカーソルを移動する，あるいは携帯電話で目的のボタンを押すなどの，物体をポイントするのに要する時間を「ポインティング時間」と呼ぶ．この，ポインティング時間  $t^p$  は Fitts の法則 [Fitts 54] を使って

$$t^p = a^p + b^p \log_2(A_i/W_i + 1) \quad (2.1)$$

で表現できることが知られている ([Norman 91] など)．ここで， $A_i$  はポイントするターゲットまでの距離， $W_i$  はターゲットの幅を意味する (図 2-5(a))．また， $\log_2(A_i/W_i + 1)$  は「困難度の指標」(Index of difficulty; ID) と呼ばれる．ターゲットまでの距離を幅で割ったものの  $\log_2$  に比例する時間がかかることをこの式は意味している．係数  $a^p, b^p$  は実験データの回帰式で決定する．一般的には対象とする個人，入力装置で異なる値となる．

Fitts の法則は図 2-5(b) に示すような実験などを基に導き出された法則であるが，マウスを使う GUI でも適用可能とされている ([MacKenzie 03] など)．

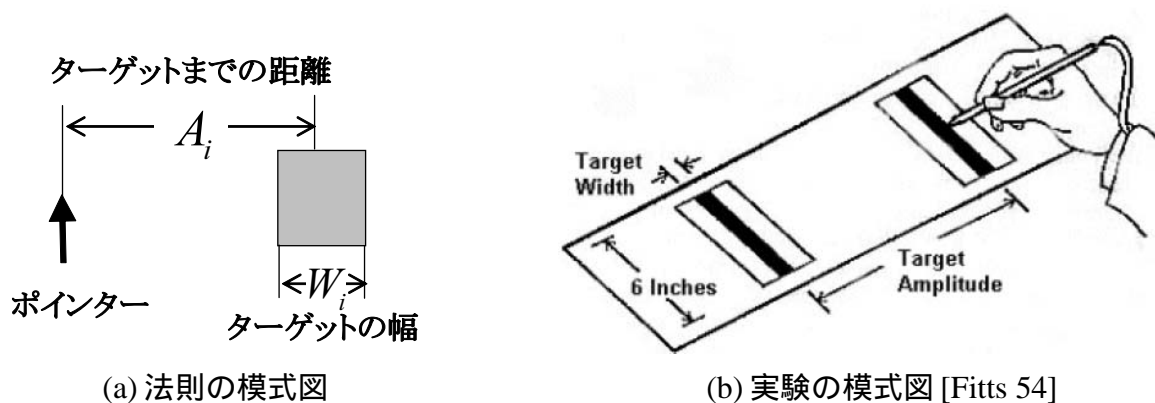
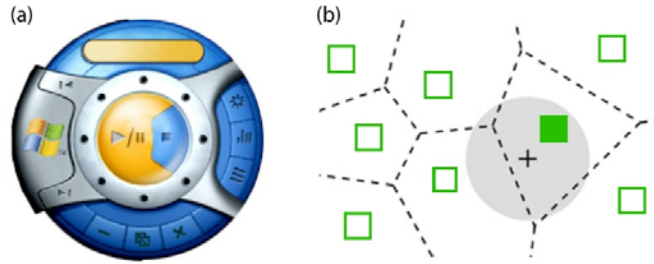


図 2-5 Fitts の法則



(a) Windows Media Player のスキンの例，(b) 緑の四角で示した領域をポイントする場合には，対応するボロノイ領域の中をポイントする．

図 2-6 ポインティング対象が長方形以外の例 [Grossman 07]

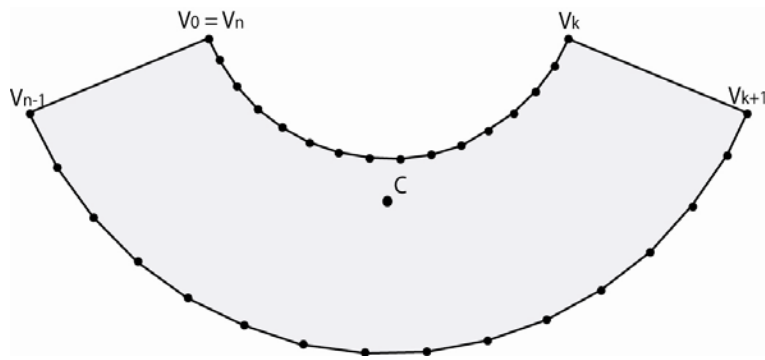
例えば，Mackenzie らはソフトキーボードでの入力速度を Fitts の法則を使ってモデル化し，実験と比較した結果を報告している [Mackenzie 99]．Qwerty, ABC, Dvorak, Fitaly, JustType, 電話機の 6 種類のキー配置にを対象に，入力速度の予測結果と実験結果を対比させ，Qwerty 配列以外は，視覚探索が必要な初心者のモデルと良く合致することを示している．また，Silfverberg らは，携帯電話でのテキスト入力のスピードの予測に Fitts の法則を用いたモデルを作成し，実験結果と良く合致することを報告している [Silfverberg 00]．

最近ではターゲットが長方形以外の形であることも多くなっているが，この場合に Fitts 法則をどのように適用したら良いのかの Grossman らの報告もある [Grossman 07]．Grossman らは，図 2-6 に示す長方形でないターゲットをポイントする場合のモデルを提案している．このモデルは，ポインティングはダーツのようなものとして捉え，投擲のばらつきを 2 次元の正規分布で近似するものである．このモデルでは，ターゲットの中心をどう求めるかが鍵となる．彼らは，(a) ターゲットの形状を多角形で近似して，その頂点の重心を中心とする方法（図 2-7(a)），(b) 被験者実験により中心を求める方法（図 2-7(b)），(c) 人間は，ばらつきを含めて最もヒット率が高くなる点を目指してヒットする，という仮説に基づいて，計算により中心を求める方法（図 2-7(c)），の 3 方式を検討し，方式 (c) で十分に方式 (b) を近似できることを示している．

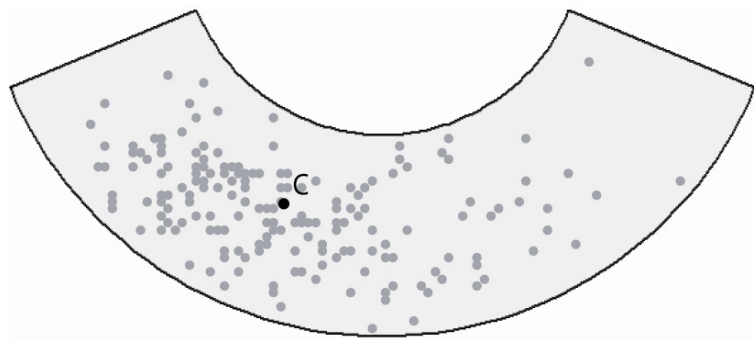
また，最近，Wobbrock らはポインティングエラーを予測するモデルを提案している [Wobbrock 08]．これによれば，ターゲットまでの距離とターゲットの幅，さらにポインティング時間によりエラー率が異なる．Wobbrock らのモデルでは，Fitts の法則を  $t = a + b \log_2(A/W + 1)$ ，実際のポインティング時間を  $t^p$ ， $\text{erf}$  で誤差関数を表すとしたとき，ポインティングエラー  $P(E)$  は次式で表現できる．

$$P(E) = 1 - \text{erf} \left\{ \frac{2.066 \frac{W}{A} \left\{ 2^{\left( \frac{t^p - a}{b} \right)} - 1 \right\}}{\sqrt{2}} \right\} \quad (2.2)$$

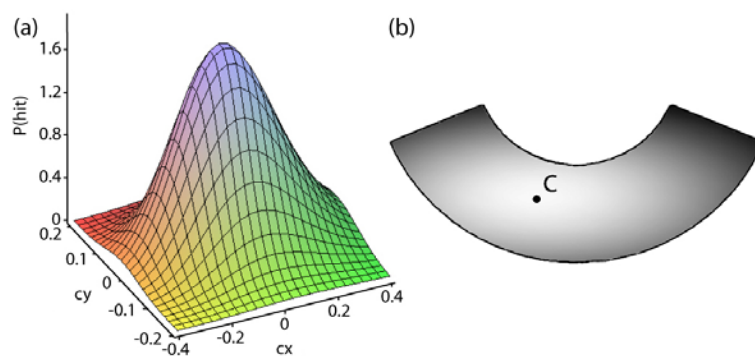




(a) 多角形近似して重心を中心とする方法



(b) 被験者実験で求める方法



(c) ヒット率が最大となる中心を計算で求める方法

図 2-7 中心の計算方法 [Grossman 07]

### 2.2.2 意思決定・選択時間

コンピュータのディスプレイ上に表示された  $n$  個の選択肢から 1 個を選ぶ場合に要する時間  $RT$  は Hick-Hyman の法則を用いて、次式で表現できる [MacKenzie 03] .

$$RT = a^{sd} + b^{sd} \log_2(n). \quad (2.3)$$

文献 [Card 83] では Hick-Hyman 法則は、 $a^{sd}$  の定数項がない、 $b^{sd} \log_2(n+1)$  で定義しているが、最近の文献、例えば [Green 04, Silfverberg 00] など、では上記の定義が一般的である .

選択肢  $i$  の選択確率が  $p_i$  の場合の一般式は、以下となる .

$$RT_i = a^{sd} + b^{sd} \log_2(1/p_i). \quad (2.4)$$

Hick-Hyman の法則もコンピュータとのインタラクションを対象に提案されたのではないが、インタラクションでの意思決定・選択時間のモデルとしても十分な精度を持っている [Cockburn 07, Mackenzie 99] .

Cockburn らは単一階層のメニュー選択のモデル化を Fitts の法則と Hick-Hyman の法則を組み合わせることを提案し、予測と実測が良く合致することを報告している [Cockburn 07] .

また、Mackenzie らは図 2-8 に示した配置などのソフトキーボードでの入力速度のモデル化に適用し、予測と実験結果が良く合致することを報告している [Mackenzie 99] . 彼らは、熟練者ではキーの探索時間は 0 と仮定し、初心者  $RT_i = 200 \log_2(27) = 951$  (ms) となると仮定してモデル化している .

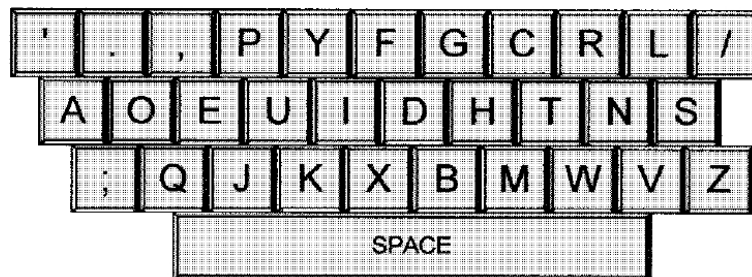


図 2-8 Dvorak 配置のソフトキーボードの例 [Mackenzie 99]

### 2.2.3 ユーザのモデル化

インタラクションにおけるユーザの行動をモデル化する研究は古くから認知科学の分野を中心に行われてきた。これらの研究の中で代表的なものを以下に紹介する。

#### 2.2.3.1 GOMS

GOMS (Goals, Operators, Methods, Selection Rules) モデルは、ゴールを達成する場合の、具体的な行為（オペレータ）、ゴールを達成するためのオペレータの組み合わせ（メソッド）、どの方法を用いるかの判断（選択ルール）でユーザの行為をモデル化する方法である [Kieras 06]。最初の GOMS モデルは Card と Moran と Newell によって 1983 年に提案された [Card 83]。

メソッドはユーザが行うオペレータの一連のステップからなり、サブゴールを呼ぶこともできるため、階層的な構造を持つ。ゴールを達成するために複数のメソッドがある場合には、選択ルールにより状況に合った適切なメソッドが選択される。様々なモデル化に利用されており、後述（第 2.4.3 節）の携帯電話のメニュー構造の分析でもユーザの行動が的確にモデル化できることが示されている。

様々な拡張があり、GOMS-KLM (GOMS-Keystroke Level Model) はキーストロークのレベルまでモデル化するものである。

#### 2.2.3.2 ACT-R

ACT-R は J.R. Anderson が提案する一連の認知モデル ACT (Adaptive Character of Thought) の最新のモデルであり、人の認知に関する膨大な心理学的知見に基づいている [Anderson 93]。高次レベルの認知をモデル化するものであり、ハノイの塔、教室での数学問題の解決、コンピュータ中の迷路の探索、コンピュータプログラミング、人間の記憶、などをうまくモデル化できることが示されている。

このモデルは、宣言記憶 (declarative memory) および手続き記憶 (procedural memory) の 2 種類からなる記憶機構を備えることを特徴としている。宣言記憶は人の長期記憶および短期記憶をモデル化したものであり、チャンクと呼ばれる命題形式の知識が格納されている（図 2-9(a)）。チャンクどうしはネットワーク状に結合されており、また各チャンクは活性度と呼ばれる動的に変化する値をもつ。そしてあるチャンクの活性度がネットワークを介して他のチャンクに伝播する機構をもつことで記憶に関する様々な現象のシミュレーションを可能にしている。手続き記憶にはプロダクションの形式で記述されたルールが格納される（図 2-9(b)）。現在のゴールを記憶する領域が別途あり、現在のゴールに一致するプロダクションが宣言記憶を参照しながら実行され、実行結果に応じて宣言記憶およびゴールが更新される。

fact3+4		
	isa	addition-fact
	addend1	three
	addend2	four
	sum	seven

(a)  $3+4=7$  のチャンク

```
IF the goal is to add n1 and n2 in a column,  
  and n1 + n2 = n3  
THEN set as a subgoal to write n3 in the column
```

(b) プロダクションの例

図 2-9 ACT-R でのチャンク, プロダクションの例 [Anderson 97]

#### Hunt-Feature

```
IF the goal is to find a target that has feature F  
  and there is an unattended object below the current  
    location with feature F  
THEN move attention to closest such object
```

#### Found-Target

```
IF the goal is to find a target  
  and the target is at location L  
THEN move the mouse to L and click
```

図 2-10 ACT-R のメニュー選択問題でのプロダクションの例 [Anderson 97]

ディスプレイでの作業で重要な注視点 (visual attention) の移動などのモデル化にも適用可能であると報告されている [Anderson 97]。この報告では、1 から 9 の数字が縦にランダムに並んだメニューからターゲットを選択する問題に対して ACT-R を適用して (図 2-10)、実験結果を良く説明できることが示されている。

### 2.2.3.3 ユーザモデルの作成支援

ACT-R, GOMS によりモデルを作成するためには, かなりの手間が必要となる. このため, モデル化を支援するためのシステムの研究が行われている. 以下に代表的な例を紹介する.

John らはユーザインタフェースの設計者がユーザモデルを簡単に作成できるシステムについて報告している [John 04]. このシステムは, HTML で記述したモックアップから, キーストロークレベルのモデルを自動的に生成する. ACT-R を使い自動的にインタフェースとインタラクションするプログラムを生成し, これによりインタフェースの性能を計測できる (図 2-11).

Linton らは, アプリケーションの使い方を長期間計測することで, ユーザ, 熟練者のモデルを作成する方法を提案している [Linton 99]. ユーザモデルは計測データから生成し, その合算として熟練者のモデルを作成する.

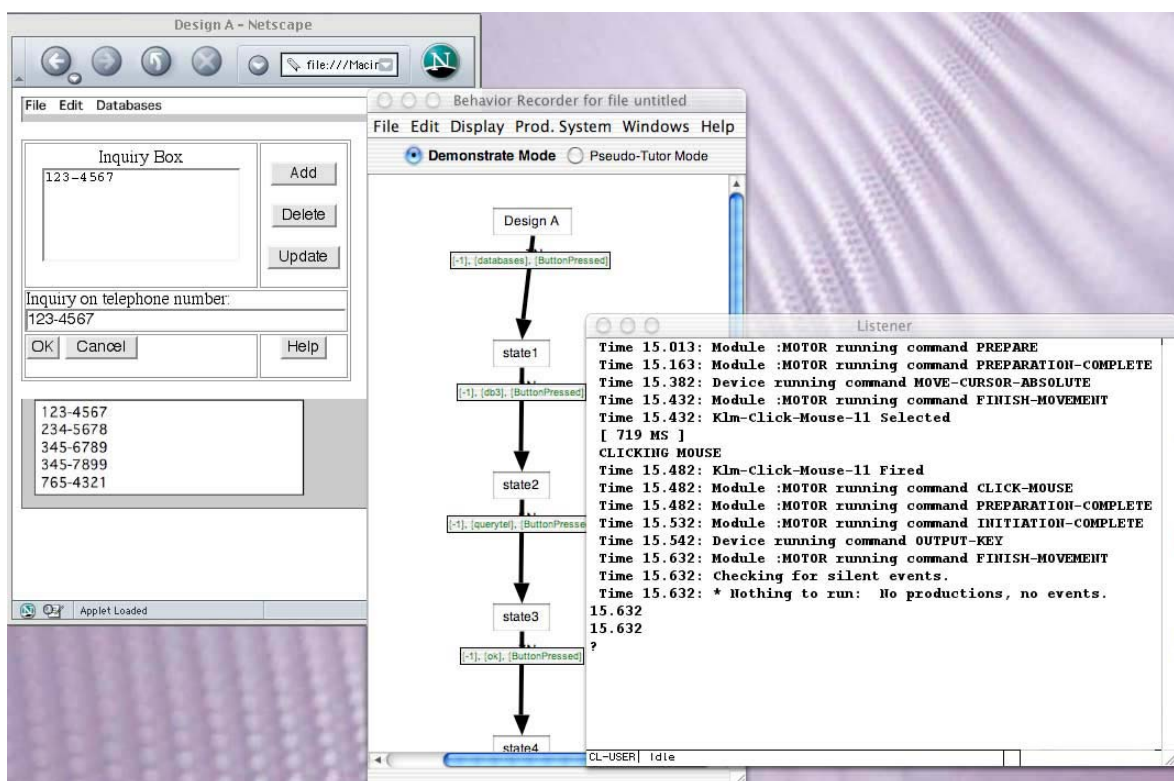


図 2-11 John らのシステムの画面例 [John 04]

## 2.3 ユーザビリティの評価

国際規格のISO9241-11では、ユーザビリティを「ある製品が、指定された利用者によって、指定された利用状況の下で、指定された目的を達成するために用いられる際の、有効さ、効率および利用者の満足度の度合い」と定義している[ユ-07]。これを基に、特定の製品の評価指標、評価方法として、以下のような提案がなされている。

表1-1に示した、一般的なユーザビリティの評価項目に加える項目として、Hanらは、家電製品、携帯電話のユーザビリティを評価するための指標を提案している[Han 01, Han 04]。彼らは「性能」に関する指標として、(1) Perception/cognition (Directness, Explicitness, Modelessness, Observability, Responsiveness, Simplicity からなる)、(2) Learning/memorization (Consistency, Familiarity, Informativeness, Learnability, Memorability, Predictability からなる)、(3) Control/action (Accessibility, Adaptability, Controllability, Effectiveness, Efficiency, Error prevention, Flexibility, Helpfulness, Multithreading, Recoverability, Task conformance からなる)の3指標を提案している。また「印象」の指標として、(1) Basic sense (Shape, Color, Brightness, Texture, Translucency, Balance, Heaviness, Volume からなる)、(2) Description of image (Metaphoric design image, Elegance, Granularity, Harmoniousness, Luxuriousness, Magnificence, Neatness, Rigidity, Salience, Dynamicity からなる)、(3) Evaluative feeling (Acceptability, Comfort, Convenience, Reliability, Attractiveness, Preference, Satisfaction からなる)の3指標を提案している。

Huangらは、携帯電話のメニュー構造を紙の上に木構造として視覚化し、メニューの評価を行う方法について報告している[Huang 07]。彼らの実験では、図2-12のように、紙にメニュー全体を展開し、実際の画面に相当する部分のみが表示されるように

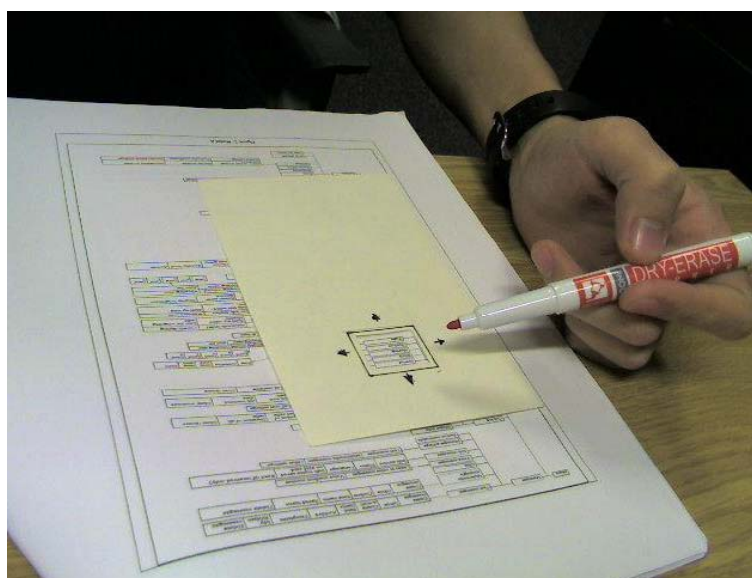


図 2-12 Huang らの実験 [Huang 07]

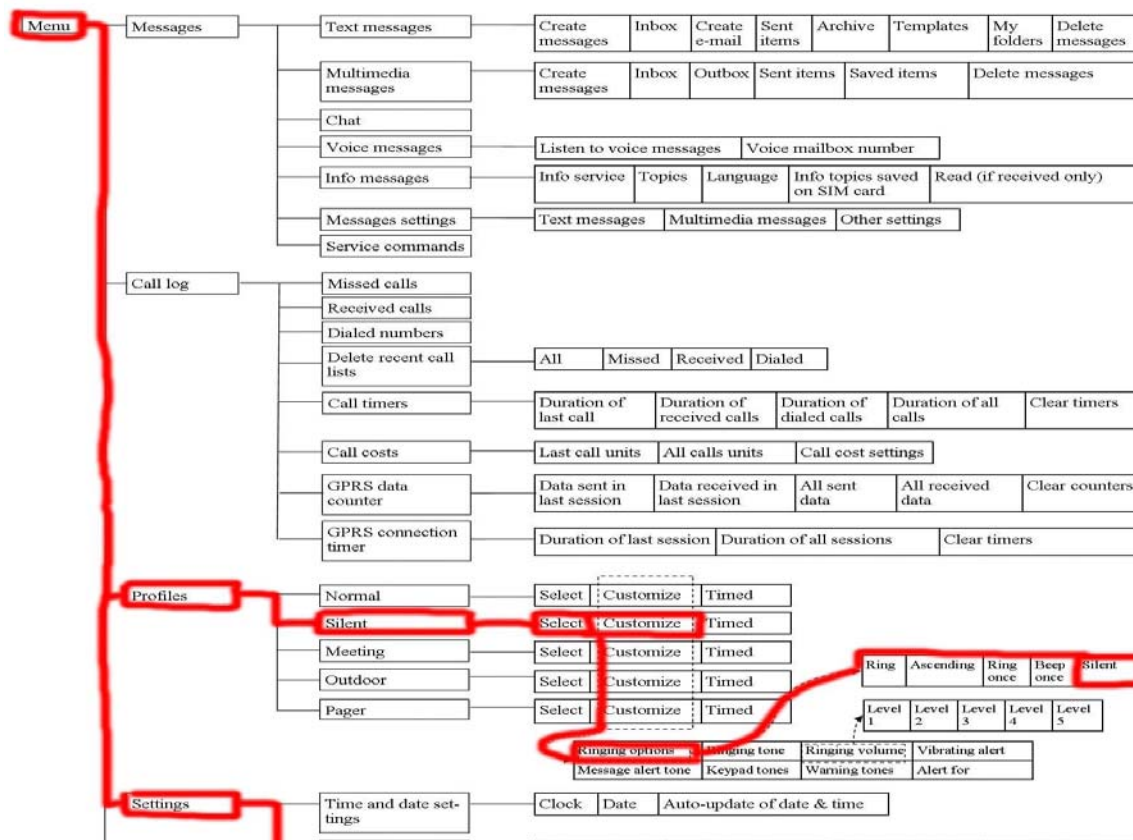


図 2-13 Huang らの実験 [Huang 07]

した窓をずらしながらメニューを辿る．その結果は図 2-13 に示したように，根から目的の項目までのパスとして残る．この方法の利点は，メニューを紙に展開できればよく，モックアップの端末を作成する，あるいは PC 上でエミュレーションするといったことが不要な点である．また，メニューの変更も簡単にできる．実験の結果，小さな画面では広すぎるメニューは好まれないこと，カテゴリ分けの重要性，ラベルの重要性が再確認できたと報告している．

Lee らは，携帯電話のユーザインタフェースを，実験室で系統的に評価するための枠組み (SEM-CPU; systematic evaluation methodology for cell phone user interface) を提案している [Lee 06]．シナリオベースのタスクパフォーマンス，質問票，実験後のインタビュー，ユーザ観察，retrospective think aloud の 5 手法を統合した方式を提案している (図 2-14)．

Ziefle らは，携帯電話のメニューについて，年齢層の違いによるメンタルモデル，使いにくさを解決するための方法について報告している [Ziefle 04, Ziefle 06]．彼らは，図 2-15 に示すような，タッチスクリーン上でシミュレートした上でタスクを実行する，あるいはメニュー項目が記述されたカードを並べることでタスクを実行する実験を行い，メンタルモデルが年齢層でどう異なるかを確認した．その結果，20～32 歳の若年



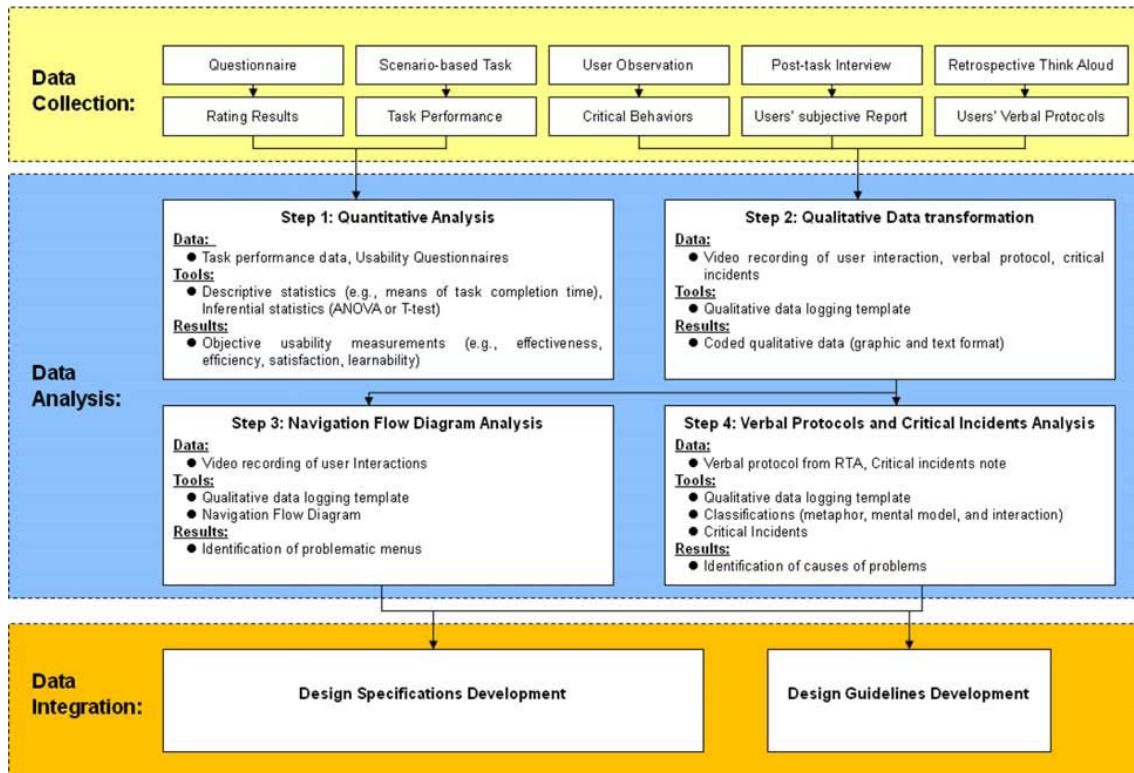


図 2-14 SEM-CPU[Lee 06]



図 2-15 Huang らの実験 [Ziefle 04]

層と50～64歳の中高年層では、メンタルモデルに差があり、中高年層はタスクを達成するためのメンタルモデルを持ちにくいこと、あるいは携帯電話の階層メニュー自体の理解が困難なことを確認した。この問題の解決には、階層構造を理解しやすく提示することが重要であることを提案している。



北島らは、潜在意味解析 (LSA) と Markov 連鎖モデルを用いて、階層メニューの探索過程を評価する方式を提案している [北島 02]。メニューやリストの中から目的とする項目を探し出すなどの項目選択タスクを、ユーザの項目選択過程の認知モデルに基づいて、(1) 項目選択をタスクとの適合度に基づいて行う、(2) 次の選択を行うときにそれまでに行ってきた項目選択の履歴を様々な仕方で考慮する、(3) ある時点で選択可能な項目の中に適当なものがなければ後戻りする、(4) タスクを達成できずに諦める、という4要素によって特徴づけられるタスクとしてとらえ、その過程を Markov 連鎖を用いてモデル化している。タスク目標が達成される確率、および達成に至るまでの平均選択回数を算出し、それに基づいて、階層メニューやウェブサイトのリンク構造を評価する方法を提案している。提案する手法に基づいて、複数のユーザモデルを提案し、モデル毎の特徴を示し、それらを用いての階層メニューでのあるタスクを実行したときの「成功率」「失敗率」「平均クリック数」を算出して、提案手法が有効であることを示している。

Karahoca らは、携帯電話での SMS (Short messaging system) のメニューのユーザビリティを評価した結果について報告している [Karahoca 06]。個人の性格と送信するメッセージの数、間隔などの関係の分析結果、および技術的スキルと所要時間の関係などが報告されている。外交的な個人ほどメッセージ数が多いこと、技術的スキルが高いほど時間が短いことが報告されている。

Keijzers らは、スマートフォンのユーザビリティを向上させるための方法につて報告している [Keijzers 08]。Nokia E6li (Symbian S60, 携帯電話タイプ), HTC S710 (Windows Mobile 6, PC タイプ), Palm Treo680 (Palm OS, PDA タイプ) の3種類を対象に、被験者実験を行い、形式の違いによるユーザビリティの違いを調べている。通話、インターネット、カレンダー、メールの4機能について、目的達成の精度、所要時間、使うメニューの数、誤った選択数、満足度から評価し、機種と機能によりそれらが異なることを示している。デザインルールは守られてはいるが、多機能を持たせようとするために、使い方が複雑になっていることを報告している。

## 2.4 メニューの最適化

本節では、メニューの最適化に関連した研究について概説する。まず、階層メニューでの幅と深さの関係についての研究を概説する。1レベルに表示する項目数が多い「幅が広く、浅いメニュー」と、1レベルに表示する項目数が少ない「幅が狭く、深いメニュー」のどちらが良いのかについての研究を紹介する。次に、単一メニューの最適化についての研究を、その後に階層メニューの最適化についての研究を紹介する。さらに、メニューの使い勝手の向上のための研究について紹介する。

### 2.4.1 浅いメニューあるいは深いメニュー

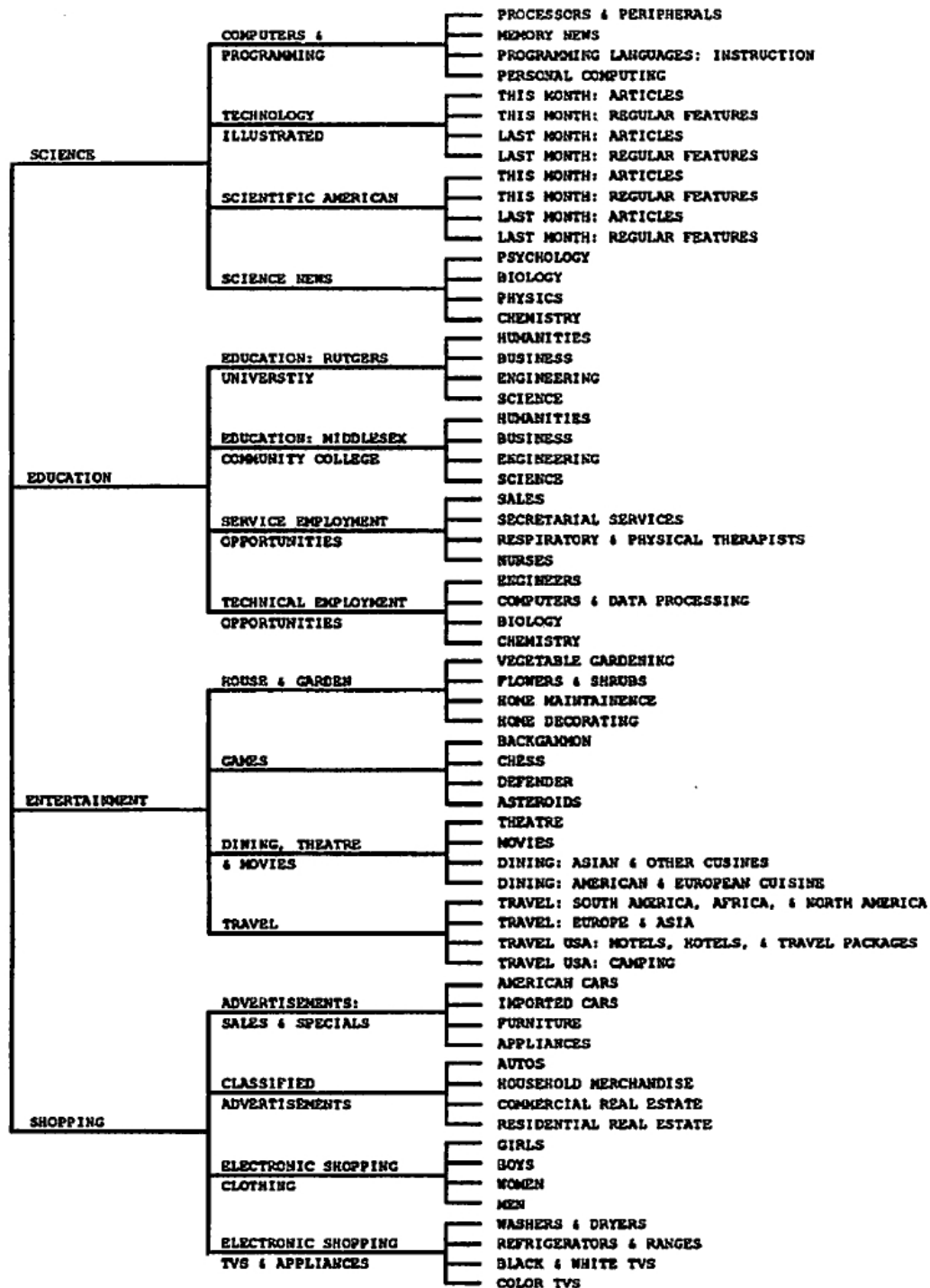
階層メニューでの深さと幅を変えた場合の性能については、古くから多くの研究がなされてきた [Kiger 84, Landauer 85, Larson 98, Parush 04, Schultz 86, Zaphiris 00, Zaphiris 03]。以下に代表的な結果を示す。

Kiger らは Videotex の 64 項目からなるメニューを対象に、1レベル毎の項目数とレベル数の違いによる性能を実験的に調べた [Kiger 84]。実験では、 $2^6$  (2項目ずつの6階層からなるメニュー構造)、 $4^3$  (4項目ずつの3階層)、 $8^2$  (8項目ずつの2階層)、 $4 \times 16$  (第1階層が4項目で、第2階層は16項目ずつ)、 $16 \times 4$  (第1階層が16項目で、第2階層は4項目ずつ) の構造でメニューを構成し、違いを調べた。 $4^3$  の構造のメニューの例を図 2-16 に示す。

22 人の被験者に 5 種類のメニューについて、それぞれ 16 個のターゲットを探索させる実験を行った。5 種類のメニューの提示順はランダムとした。「どれを好ましいとするか」「どれが使いやすかったか」についての質問からは、 $8^2$ 、 $4^3$  の評価が高く、 $16 \times 4$ 、 $2^6$  の評価が低かった。総探索時間の観点からは、 $4 \times 16$ 、 $16 \times 4$ 、 $8^2$  では殆ど差がなく、 $4^3$ 、 $2^6$  より短かった。 $2^6$  は最も時間がかかり、またエラー率も高かった。全体としては、 $8^2$  が最も良いメニューであった。Schultz らも同様な実験を行い、浅いメニューの方が性能が高いことを報告している [Schultz 86]。

Landauer らは、タッチスクリーンを使って同様な実験を行い、浅いメニューの方が良いことを報告している [Landauer 85]。彼らの実験では、 $N = 4096$  項目の数字あるいは単語を選択させる実験を行っている。1画面に表示させる項目数  $b$  を  $b^m = N$  となるように、 $m = 2, \dots, 16$  と変えたメニュー構成で実験を行った。1階層のメニューはスクリーン全体を使う方式を採っており、項目数が増えると、1項目あたりの大きさが減少する構成となっている。

彼らは、項目内からの選択は Hick-Hyman の法則  $t = c_1 + c_2 \log(b)$  に従い、ポインティング時間は Fitts の法則  $mt = c_3 + c_4 \log(b)$  に従う、と仮定した。ここで Fitts の法則で  $\log(b)$  となっているのは、移動距離  $d$ 、ターゲットの幅  $w$  としたとき、Fitts の法則  $mt = c_3 + c_4 \log(d/w)$  であるが、項目数に反比例して幅が狭くなる、すなわち  $w$  は

図 2-16 Kiger の実験でのメニューの例 (4<sup>3</sup>)[Kiger 84]

1/b に比例するためである。実験の結果、 $b = 2$  と  $b = 16$  では探索に要する時間が 2 倍程度異なり、 $b = 16$  が最も速かったことを報告している。

階層メニューではないが、Larson らは Web ページの設計での幅と深さの関係を調べた結果を報告している [Larson 98]。Encarta 百科事典から 512 項目を抜きだし、 $8 \times 8 \times 8$

(8個のトップレベルのカテゴリ, 8個の中間レベル, 8個のコンテンツレベル)  $16 \times 32$  (16個のトップレベルのカテゴリ, 32個のコンテンツレベル),  $32 \times 16$  (32個のトップレベルのカテゴリ, 16個のコンテンツレベル) の3構成で実験を行った。ただし, カテゴリ分けが自然な構成となるように, 512項目から抜き出してメニューを構成したため, 3種類全てに含まれる項目は128項目となった。いずれの構成でも, どの構成のメニューで, どのレベルに在るかが画面上に表示される構成とした。

平均検索時間は,  $16 \times 32$  で36秒 (標準偏差16)  $32 \times 16$  で46秒 (標準偏差26),  $8 \times 8 \times 8$  で58秒 (標準偏差23) であった。「好みの構成か」, 「ナビゲーションが容易であったか」, 「簡単に使えたか」, などの主観的評価では,  $32 \times 16$  の評価が最も高かった。

Zaphiris らも同様に, Web ページでの幅と深さの関係についての実験結果を報告している [Zaphiris 00]。64個の最終ノードを持つ Cyprus 島に関する Web ページを対象に,  $2 \times 6$  (幅2, 深さ6),  $4 \times 3$  (幅4, 深さ3),  $8 \times 2$  (幅8, 深さ8),  $4 \times 1$ ,  $16 \times 1$  (最初のページは幅が4, 次のページは幅が16),  $16 \times 1$ ,  $4 \times 1$  (最初のページは幅が16, 次のページは幅が4) の5構成で実験を行った。

その結果,  $8 \times 2$  では平均到達時間が20.3秒 (標準偏差11.7),  $4 \times 1$ ,  $16 \times 1$  では平均到達時間が21.0秒 (標準偏差11.3) となり,  $2 \times 6$  で36.0(14.9),  $16 \times 1$ ,  $4 \times 1$  で32.4(19.8) などに比べて, 目的とする項目に辿り着くまでに要する時間が短かった。使いやすさの主観的評価では,  $16 \times 1$ ,  $4 \times 1$  の評価が最も高かった。総合的には,  $4 \times 1$ ,  $16 \times 1$ ,  $16 \times 1$ ,  $4 \times 1$  が良いと報告している。

また彼らは, 階層型の情報提示システムを対象に, 幅と深さの関係についての実験結果を報告している [Zaphiris 03]。この実験では, 幅と深さの関係に加えて, Web ページの画面でリンクをクリックして次の画面に移った場合に, 前画面の項目が展開される形式と, 展開されない形式の違いなどと, 利用者の年齢による違いを調べている (図2-17)。実験の結果として以下を報告している。

- 年齢に関係なく, 浅い方を好む。年齢が若い場合には展開される形式を好むが, 年齢が高い場合には展開されない形式を好む。
- 階層のどこに在るか良く分かったかとの問いに対しては, 年齢が高い場合の方が低い評価であった。年齢に関係なく, 浅いメニューの方がどこに在るか良く分かったと評価した。
- 目的までの到達時間には年齢の差がはっきりと現れ, 年齢が高い方が時間を要した。年齢に関係なく, 浅い構造の方が時間が短かった。年齢が若いと, 展開されない深さ2の場合が最も速く, 深さ6の展開されないものが1番遅かった。年齢が高いと展開される深さ2の場合が最も速く, 展開される深さ6の場合が最も遅かった。

Parush らは, 携帯電話に表示される Web 画面での幅と深さの関係について報告している [Parush 04]。彼らは, 携帯電話端末の画面を PC 上でエミュレートし (図2-18),

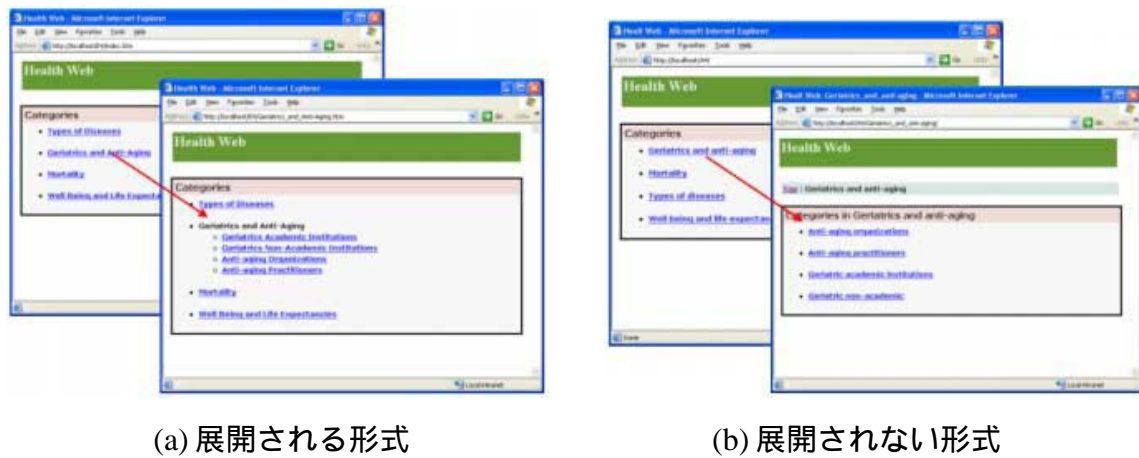


図 2-17 Zaphiris らの実験での画面例 [Zaphiris 03]

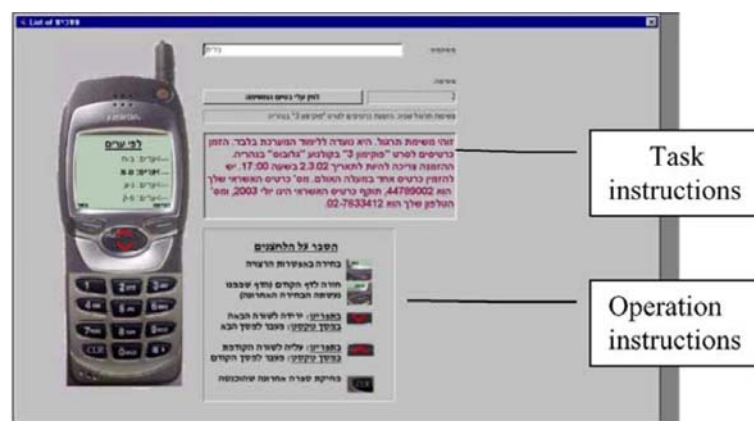


図 2-18 Parush らの実験での画面例 [Parush 04]

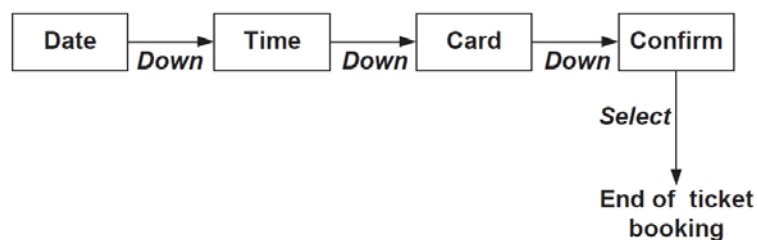


図 2-19 Parush らの実験での浅いナビゲーションの例 [Parush 04]

ナビゲーションの深いもの（図 2-20）ナビゲーションの浅いもの（図 2-19）での差を調べている。

検索のタスクとチケット予約のタスクについて実験を行い，ナビゲーションが浅い方が（図 2-20），所要時間が短く，また成功率も高かったことを報告している。

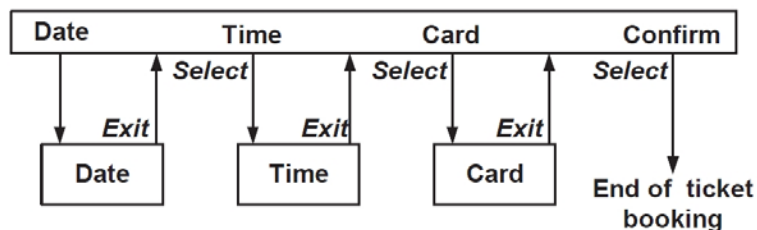


図 2-20 Parush らの実験での深いナビゲーションの例 [Parush 04]

表 2-1 Geven らの実験でのメニューの構造 [Geven 06]

名前	レベル毎の項目数	レベル数
非常に狭い	4 項目	6 レベル
狭い	8 項目	4 レベル
広い	16 項目	3 レベル
非常に広い	64 項目	1 レベル



図 2-21 対象とした端末 [Geven 06]

Geven らは、携帯電話で階層型に情報を提示する場合の、幅と深さの関係について報告している [Geven 06]。図 2-21 に示した 3 種類の端末を対象に実験を行った。3 種類の端末は、左から順に、Siemens C62 (4 行表示可能)、Nokia 6600 (10 行表示可能)、Compaq iPaq pocketPC H3870 (16 行表示可能) である。これらの端末において、表 2-1 の形式の階層メニュー (図 2-22) 表示させて実験を行った。

$4^6, 8^4, 16^3, 64^2$  の 4 種類の形式について被験者実験を行い、 $4^6, 8^4$  の方式が優れていたことを報告している。この結果は、表示画面が広い場合には、浅いメニューが好まれるという結果と異なるが、表示画面が非常に小さいために、スクロールが必要になることが多いため、浅いメニューの性能が悪化しているためと考えられる。



Interactive Genetic Algorithm) で最適化する方式を報告している [Quiroz 07] . 彼らの研究では, インタフェースの設計ガイドラインには相反するものがあり, ユーザによって決定すべき事項もあることから, 対話的 GA を用いて, 最適設計を行うことを提案している. GUI 部品の配置と色使いを染色体に表現し (図 2-24), 個体の良し悪しの評価は, ガイドラインへの合致度の重み付き総和とユーザによる評価を合わせて行い, 個体を進化させている.

IGA では個体の評価をユーザに委ねることから, その回数はできるだけ少ないことが望ましい. そのために全ての個体を評価せずに, 上位 2 個体のみを選択させること, また全ての世代でユーザに評価させずに, ある間隔で評価する方式をとっている.

この方式での初期個体群の例を図 2-25 に, 第 20 世代での個体群の例を図 2-26 に示す. 初期個体群では, スライダーなどの部品の配置, 色使いはバラバラであるが, ユーザの評価が加わって進化した第 200 世代では, 部品の配置は統一され, 色使いも似たものとなっている.

表 2-2 Liu らの実験で用いられた画面構成 [Liu 02]

Model predicted fast				Model predicted slow				Designer predicted fast			
Pred	Freq	Layout	Color	Pred	Freq	Layout	Color	Pred	Freq	Layout	Color
175	4	professor	Red	500	10	air	Black	225	4	professor	Blue
100	9	roadhouse	Red	150	2	circle	Red	250	9	map	Black
200	4	butterfly	Blue	175	3	log	Black	200	10	beauty	Red
125	10	submarine	Blue	200	3	toy	Red	250	10	air	Red
225	3	minute	Blue	125	2	rubber	Red	150	10	submarine	Red
100	9	map	Blue	350	4	insect	Gray	100	10	ground	Red
100	10	air	Gray	400	9	map	Gray	100	9	roadhouse	Blue
375	3	log	Gray	75	2	war	Red	175	4	insect	Black
225	4	insect	Black	325	3	ear	Black	200	4	butterfly	Black
425	2	war	Gray	225	3	minute	Blue	75	4	horseback	Blue
175	3	toy	Black	350	4	professor	Black	325	3	bed	Black
125	3	bed	Gray	500	10	beauty	Black	350	3	toy	Black
75	10	ground	Red	475	10	submarine	Gray	375	3	ear	Black
125	3	heaven	Gray	300	4	butterfly	Blue	400	3	log	Black
100	10	beauty	Black	450	9	roadhouse	Black	400	3	heaven	Black
400	2	rubber	Gray	400	4	horseback	Black	425	3	minute	Black
350	2	circle	Gray	100	2	tip	Red	400	2	tip	Gray
200	3	ear	Blue	250	3	heaven	Black	425	2	war	Gray
425	2	tip	Blue	250	3	bed	Red	450	2	circle	Gray
150	4	horseback	Blue	550	10	ground	Blue	500	2	rubber	Gray

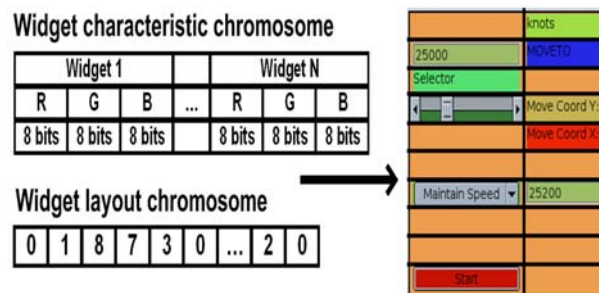


図 2-24 IGA によるメニューの最適化での染色体表現 [Quiroz 07]





図 2-25 IGA によるメニューの最適化例（初期個体）[Quiroz 07]

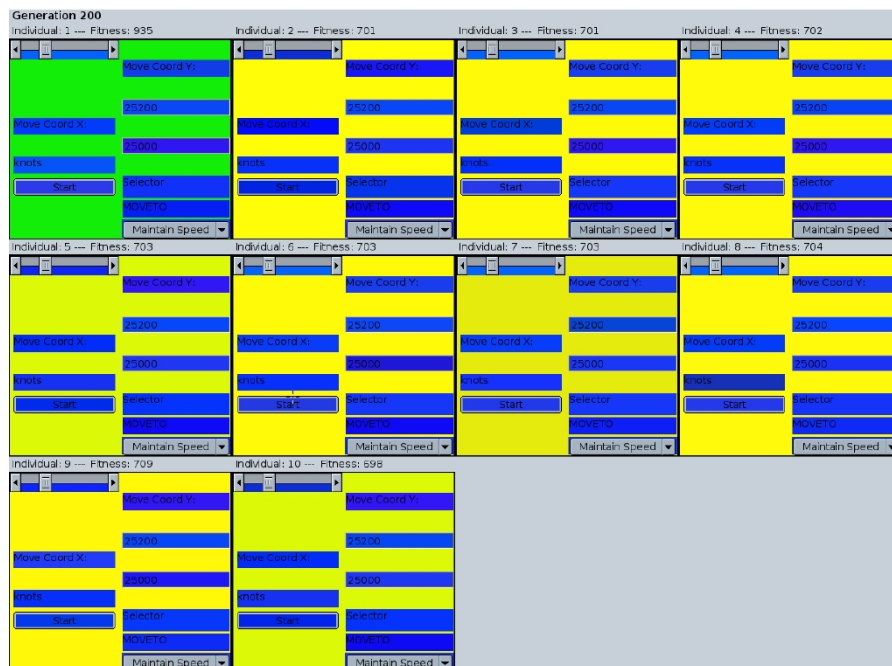


図 2-26 IGA によるメニューの最適化例（200 世代目）[Quiroz 07]

### 2.4.3 階層メニューの最適化

本節では，階層メニューの最適化に関する研究について概説する．まず，Francisらの多機能ディスプレイ（戦闘機のコックピットに搭載する情報表示装置，あるいは銀行のATMのようなもの）の最適化に関する研究について概説する．次に，Amantらの携帯電話のメニューを対象とした研究について概説する．

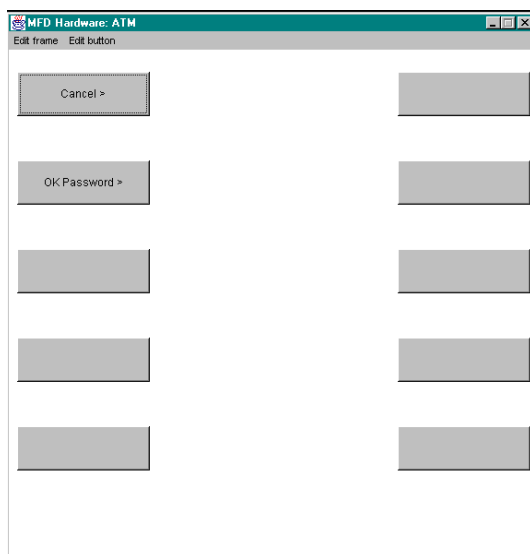
Francisらは，多機能ディスプレイ (multifunction display) の表示方式の最適化を報告している [Francis 00, Francis 02]．彼らの研究は，多機能ディスプレイを対象としたものである（図 2-27）．このため，メニューの構造は予め与えられており，その階層構造を保ったまま，ハードウェア的に決まるディスプレイ上に表示されるボタンに適切にラベルを割り当てることにより最適化する問題となっている．

特定のラベル（ボタン） $i$  に辿り着くまでの時間は，ボタンに到達するためのパスに関連する時間  $B(i)$  とラベルを識別するための時間  $L(i)$  からなるものとする．また， $B(i)$  と  $L(i)$  は独立なものとしている．メニュー全体でのラベルの数を  $m$  とし，ラベル  $i$  の選択確率を  $p_i$  として，平均到達時間  $E(T)$  は以下となる．

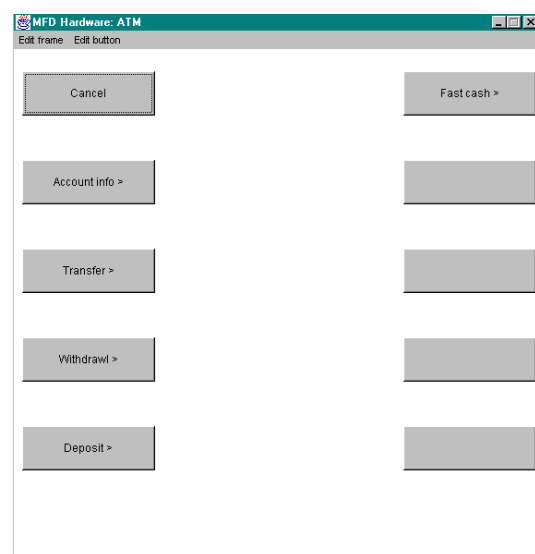
$$E(T) = \sum_{i=1}^m B(i)p_i + \sum_{i=1}^m L(i)p_i. \quad (2.5)$$

最適化の目的は  $E(T)$  の最小化であり， $\sum_{i=1}^m L(i)p_i$  は一定と仮定して， $\sum_{i=1}^m B(i)p_i$  を最小化する問題としている． $B(i)$  のモデル化には Fitts の法則を用いている．

最適化手法としては，シミュレーテッドアニーリング (SA: Simulated Annealing) に基づく方式を使っている．



(a) 初期画面



(b) “OK Password” 選択後の画面

図 2-27 Francis らの対象とする画面例 (ATM) [Francis 02]

Amant らは、携帯電話のメニュー構造を分析するための方法を提案している [Amant 04, Amant 07]。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いる。実験の結果として、GOMS モデルによる予測が最も適しており、利用頻度の高い項目を、メニュー階層の浅い位置に上げていくという単純な手法でも、30% 程度の時間削減が可能であることを示している [Amant 04, Amant 07]。

図 2-28 に GOMS による操作のモデル化の例を示す。例の上部では「音量調節」へ到達するまでのステップを記述している。ラベルが“Menu”のオブジェクトを探し、“Select Menu”を達成し、“Select Settings”を達成し、“Select Sounds”を達成し、“Select Ringer Volume”を達成し、ゴールに到達したので戻る、という一連のステップからなる。

例の下部では、“Select Setting”を達成するステップの詳細化を必要な時間とともに示している。最初のステップでは、“Settings”が画面上になく、スクロールボタンに指がなければ、スクロールボタンに移動することを意味する。次の行は、“Settings”が見つかるまでスクロールすることを意味する。

このようにユーザの動作を詳細に記述する GOMS によるモデルの予測精度が高いことを示しており、このモデルを用いてメニュー構造を変更することによる、平均到達時間の短縮を行っている。

Fitts の法則を使ったモデルの予測精度が低いのは、項目選択における意思決定・選択時間のモデルが含まれないためと考えられる。また、彼らのメニュー構造の変更では、項目を上位に上げていくだけしか行えない。これは、メニューの類似度を考慮する方法が組み込まれていないためである。

#### *Method for TraversalTo RingerVolume*

Step. Look for object whose label is “Menu”.  
 Step. Accomplish goal: Select Menu.  
 Step. Accomplish goal: Select Settings.  
 Step. Accomplish goal: Select Sounds.  
 Step. Accomplish goal: Select RingerVolume.  
 Step. Return with goal accomplished.

#### *Method for Select Settings*

Step Scroll-Test-1.	If ( <i>Settings</i> not found and not on-scroll-button) Move-to scroll-button.	0.050 s or 0.133 s
Step Scroll.	Press scroll-button; Goto Scroll-Test-1.	0.330 s
Step Scroll-Test-2.	If ( <i>Settings</i> not found and on-scroll-button) Press scroll-button; Goto Scroll-Test-1.	0.050 s or 0.330 s
Step OK-Test-1.	If ( <i>Settings</i> found and on-OK-button) Press OK; goto End. ⇩	0.050 s or 0.330 s
Step OK-Test-2.	If ( <i>Settings</i> found and not on-OK-button) Move-to OK-button.	0.050 s or 0.163 s
Step OK.	Press OK; Goto End.	0.330 s
Step End.	Return with goal accomplished.	0.050 s

図 2-28 GOMS による操作のモデル化例 [Amant 07]

#### 2.4.4 メニューの使い勝手の向上方策

Beck らは、表示画面の小さな携帯電話において、ナビゲーションの助けとなるサブウィンドウを表示する方式の提案を行っている [Beck 06]. サブウィンドウには、サブメニューを上位メニューとともに表示する、独立したサブウィンドウが好まれること、項目が選択された直後にサブウィンドウを表示すると、選択時間が短くなることなどを報告している。携帯電話だけでなく、PDA などにも適用可能な方式であるとも報告している。

Findlater らは、静的 (static)、適応可能型 (adaptable)、適応型 (adaptive) の3種のメニューの比較結果を報告している [Findlater 04]。上位の4項目が固定された、スプリットメニューの形式を用いて3種の比較実験を行った結果、静的メニューの性能が高く、また、適応可能型メニューの方が、適応型よりも性能が高かったことを報告している。しかし、ユーザは適応可能型を好むことも報告している。

Ahlström は、カスケード型のプルダウンメニュー（階層メニュー）でメニュー選択に Fitts の法則、ステアリング法則、力場 (force fields) を使ったユーザのモデル化と改善方法を提案している [Ahlström 05]。力場はカーソルをある方向に引きつける方式であり、サブメニューを持つ項目が選択されると、カーソルを右方向に（サブメニューが展開される方向に）引きつける。これにより、18%程度の選択時間の短縮が可能であったことを報告している。

Web 画面でのメニューデザインについては、Yu らが、実験的に設計方式を評価した結果を報告している [Yu 02]。Web 上に、図 2-29 の階層構造を持つ仮想的なショッピングモールを構築し、メニュー設計の違いによる検索とブラウジング速度の違いを実験的に調べている。

ブラウジングには、単純な選択型のメニュー（図 2-30(a)）より、大局的と局所的の情報を同時に提示するメニュー（図 2-30(b)）を提示する方式が良いこと、探索ではプルダウン型のメニュー（図 2-31）が良いことを報告している。

Lyons らは、ウェアブルコンピュータ用に、片手で操作するキーボード入力を前提とした KeyMenu を提案している [Lyons 03]。このメニューではキーとポップアップメニューの項目が1対1になるように配置されている。

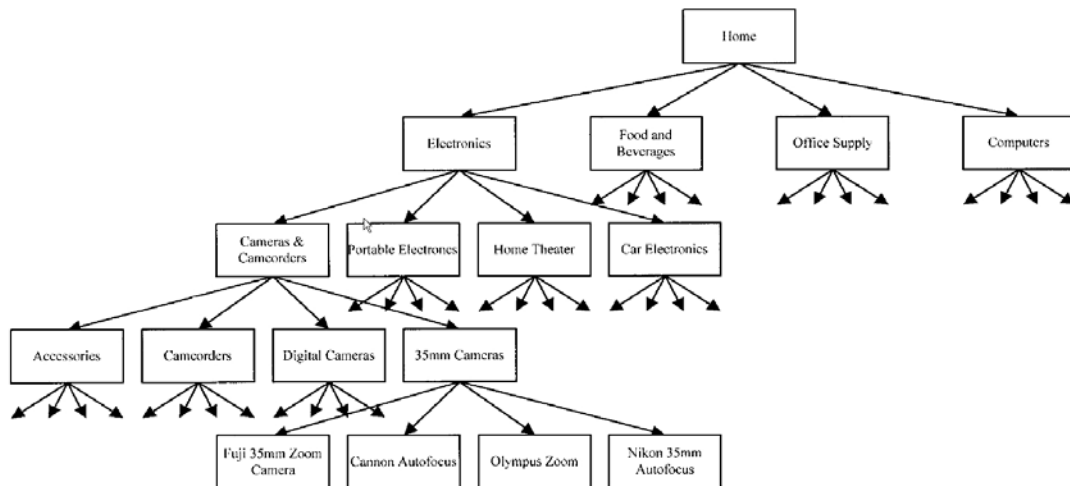


図 2-29 Web でのメニュー構造 [Yu 02]



(a) 単純な選択型

(b) 大局的と局所的の情報の同時提示

図 2-30 Web 画面例 [Yu 02]



図 2-31 Web 画面例 (プルダウン型) [Yu 02]

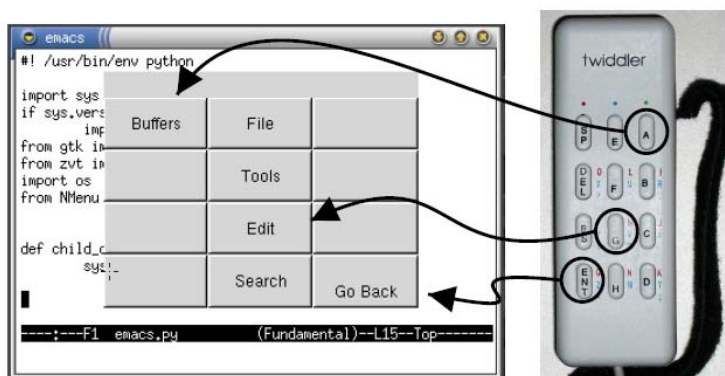


図 2-32 KeyMenu の模式図 [Lyons 03]

## 2.5 ユーザインタフェースの最適化

コンピュータと利用者の間に位置するユーザインタフェースについては、多くの研究がなされてきた。本節では、使い勝手を高める、目的とする機能にできるだけ早く到達できるようにする、といった観点からの研究について概説する。これらの研究では、最適化というキーワードが明示的にふくまれているわけではないが、前述の目的を達成するための研究であるという視点からは、関連性が高い。

Gajos らは、ユーザの特性に合わせたユーザインタフェースを自動的に生成するシステムについて報告している [Weld 03]。SUPPLE, SUPPLE++ [Gajos 04, Gajos 08b] はそのためのソフトウェアであり、ユーザの選好を予測する方式 [Gajos 05]、障害を持つ人用のインタフェースの自動生成 [Gajos 08b] などの一連の成果が報告されている（図 2-33）。

彼らの枠組みでは、図 2-33 に示すように「ユーザの選好」または「ユーザの身体能力」に関連する情報を、図 2-34 に示したインタフェースを用いて獲得する。この情報に基づいて「ユーザの選好のモデル」または「ユーザの身体能力のモデル」を作成し、「機能仕様」と「デバイスの制約」も入力として、SUPPLE あるいは SUPPLE++ を用いて、適切なインタフェースを生成する。

障害を持つ人用のインタフェースを自動生成した例を図 2-35 に示す。図 2-35 の (a) は健常者を想定した「印刷」用のダイアログ画面である。これを障害者の身体能力をモデル化した結果に従って、個人用に最適化した例が (b) である。プリンタ名、印刷部数などのプルダウンメニューは展開されて選択リストの形に変わっている。これにより、制限された身体能力を持つ特定の利用者に適した形となっている。

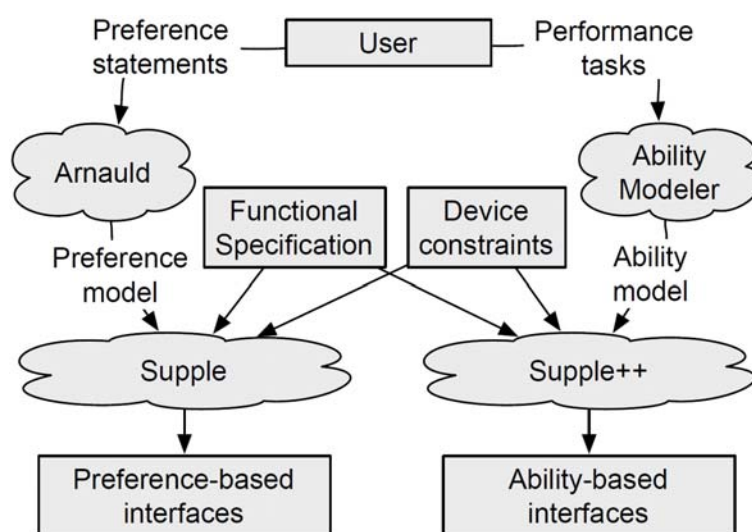
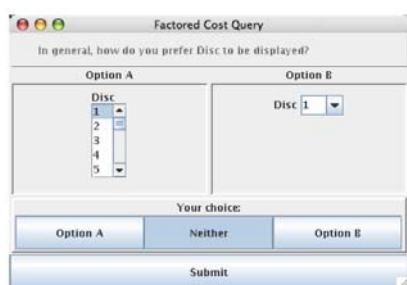


図 2-33 ユーザの特性に合わせたインタフェースの自動生成の枠組み [Gajos 08b]



(a) An example of a query used during the active elicitation part of the preference elicitation.



(b) Four task types used to measure participants' motor capabilities

図 2-34 ユーザのモデリングのためのインタフェースの例 [Gajos 08b]

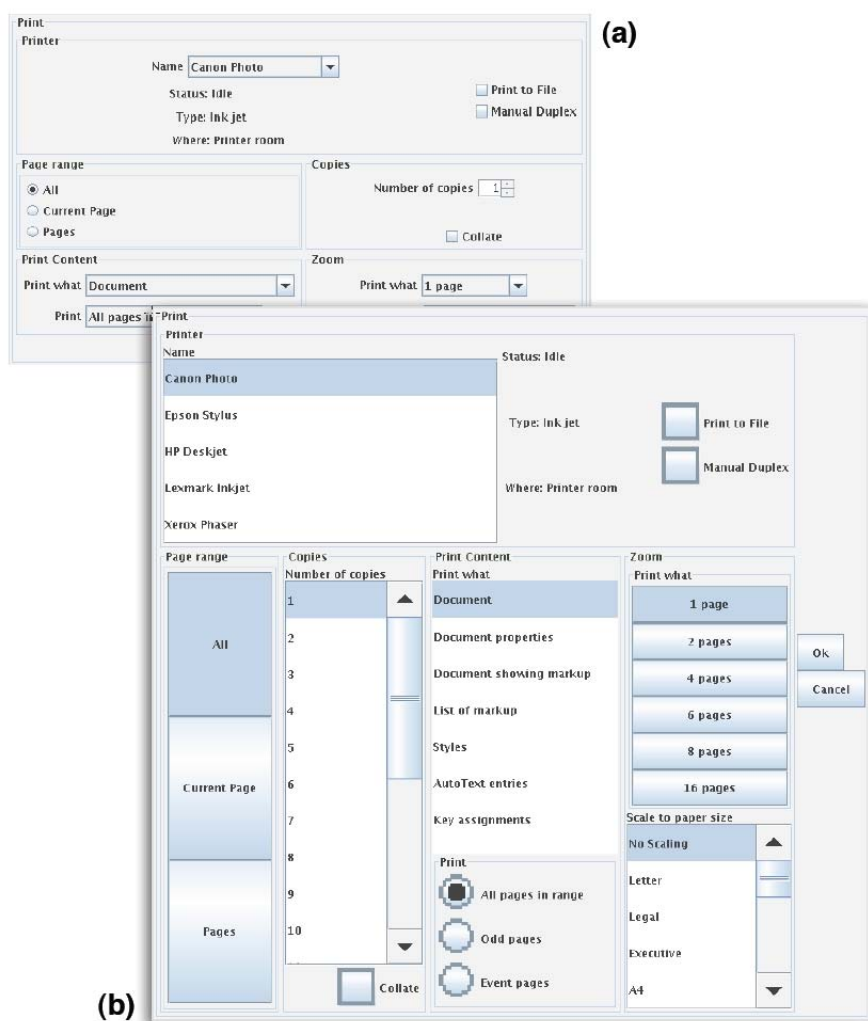


図 2-35 SUPPLE++による生成例 [Gajos 08b]



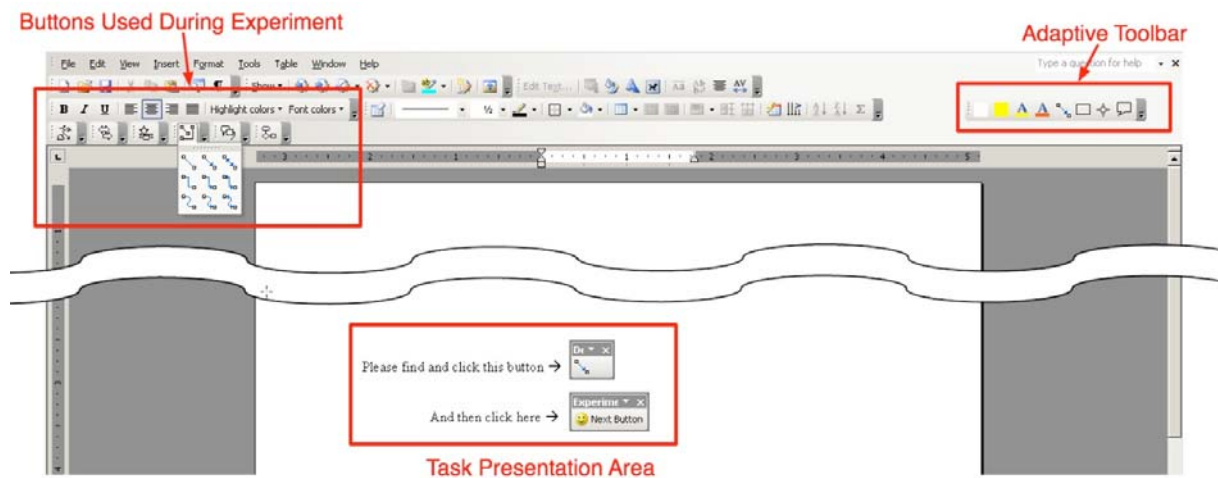


図 2-36 Gajos らの実験の構成 [Gajos 08a]

また、Gajos らは、適応的インタフェースの予測精度が高い場合には、固定的なインタフェースよりも好まれることを報告している [Gajos 08a]。 “Task Presentation Area” に表示するコマンドが “Adaptive Toolbar” に含まれている割合を予測精度として、図 2-36 に示した構成で、“Adaptive Toolbar” に表示するコマンドの予測精度と予測可能性を変化させて被験者実験を行った。予測精度は 50% と 70% の二つの条件を設定した。予測可能性を明示的に制御することは困難であることから、ランダムな状態と直近に使った 8 個のコマンドを使う方式を採った。

一連の作業が完了するまでの時間を計測し、実験後にアンケート調査を実施した。その結果、予測精度が 70% の場合には、50% の場合と統計的に有意な差がでた。予測可能性の違いはなかった。また、予測精度が高いほど “Adaptive Toolbar” の利用割合が高まる結果が得られた。

Bridle らは携帯電話でのショートカットの有効性について報告している [Bridle 06]。Nokia series 60 の携帯電話を対象に、目的とする機能に辿り着くまでに押す必要のある一連の数字列をショートカットとすることによる、時間短縮率をシミュレーションにより確認している。ショートカットは、「最後に行った操作 (Last Performed)」、「最も利用頻度の高い操作 (Most Frequent)」、「C4.5 により生成した意思決定木により、曜日、時間などにより予測する (C4.5 Based)」、「単純ベイズ推定 (Naïve-Bayes Based)」、「コンタクト先応じた選択 (Contingency Based)」、「最も頻度の高い操作とベイズ推定を組み合わせたハイブリッド手法 (Hybrid Approach)」により生成する。シミュレーション結果からは、ハイブリッド手法の性能が高いことを報告している (図 2-37)。

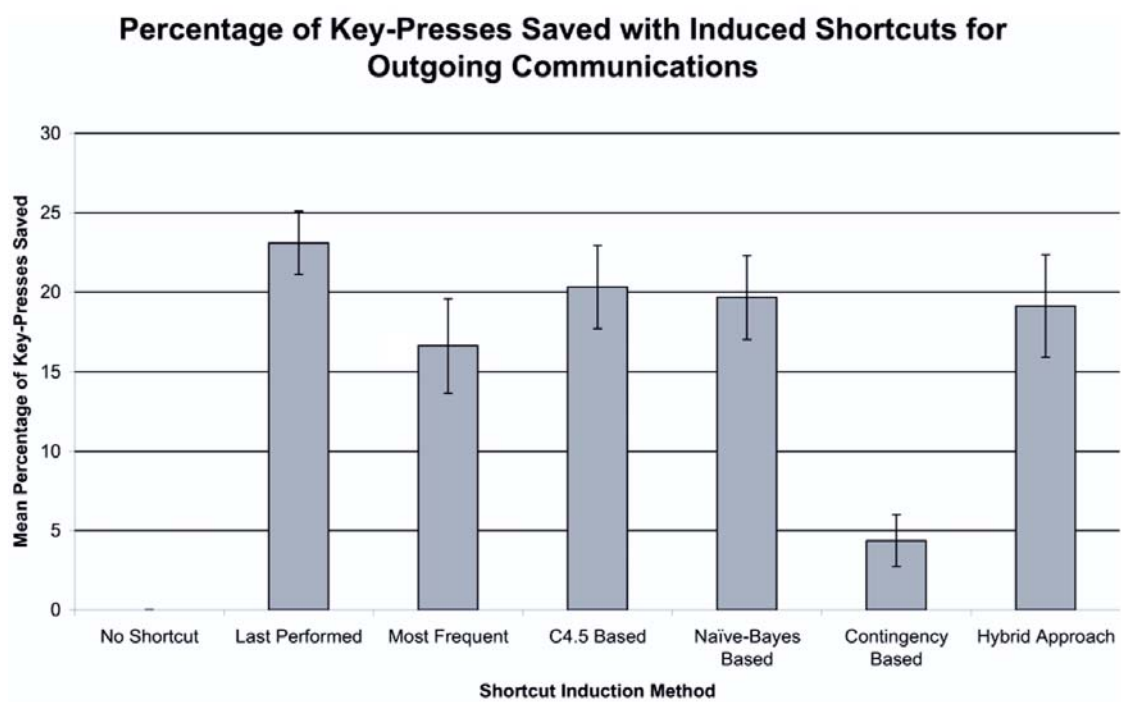


図 2-37 携帯電話でのショートカットの効果 [Bridle 06]

## 2.6 遺伝的アルゴリズム

メタヒューリスティックの一手法である遺伝的アルゴリズム (Genetic Algorithm; GA) は、生物の進化過程を模擬した計算手法である [Goldberg 89, 北野 93, 伊庭 94]。最適解が求まる保証はないが、関数の勾配情報が不要で、良い近似解が求まることから、様々な最適化問題に適用されている [Goldberg 89, 北野 93, 伊庭 94]。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ため、GA を使うこととした。

### 2.6.1 遺伝的アルゴリズムの基本構造

遺伝的アルゴリズムは生成・テスト (Generate-and-Test) 型のアルゴリズムであり、一般に「選択 (selection)」「交叉 (crossover)」「突然変異 (mutation)」の3種類の遺伝的操作を用いて集団を進化させる。問題の解候補を遺伝子型 (genotype) として染色体 (chromosome) に (最も単純には) 1 次元的に表現する。各世代は個体 (individual) の集合である。染色体に対応した問題の解は表現型 (phenotype) と呼ばれる。個体数の決定や染色体の長さ、コーディングの方法などは問題に合わせて決定する必要がある。表現方法が適切でないと性能の低い GA しか得られない [小林 93, Rothlauf 00]。

例えば、組み合わせ最適化問題の一つである周波数割当問題に対する GA で、基地局へ割り当てる周波数の列を染色体としてコーディングすることは直感的である。しかしこの方式では、高い性能は得られない [Ngo 98, 村上 00]。これに対して、割当順序を表現することで、高性能な GA を構成できることが示されている [松井 00, Matsui 00, 松井 01, Matsui 01, 松井 02a, 松井 02b, Matsui 02, 松井 03, Matsui 03]。

遺伝的アルゴリズムの基本的な処理手順を図 2-38 に示す。「初期集団の生成」では、一般的には、決められた個体数の染色体をランダムに生成する。

「適応度の評価」では、個体が表現している解候補の良し悪しの程度を評価する。一般的には、遺伝子型から表現型への変換操作が必要になる。最も単純な場合では目的関数  $f(x)$  を計算することで適応度の評価となる。一般的には良い解が高い適応度とな

1. 初期集団の生成
2. 終了条件が満たされるまで繰り返す
  - (a) 適応度の評価
  - (b) 選択
  - (c) 交叉
  - (d) 突然変異

図 2-38 遺伝的アルゴリズムの処理手順

表 2-3 スケーリング方法

スケーリング方法	関数
線形スケーリング	$f' = af + b$
シグマ切断	$f' = f - (\bar{f} - c \times \sigma)$
べき乗スケーリング	$f' = f^k$

るようにする．個体が表現している解候補から適応度が評価できればよく，関数の勾配情報は不要である．

適応度が求まったら，それを基に選択交配を行う．ここでは，適応度の高い個体により多くの子孫を残すことにより，よい個体を形成する遺伝子が集団中に広がるようにするのが一般的である．

## 2.6.2 選択交配

選択交配のための「選択」方式には様々な方式が提案されている．選択の代表的な手法には以下がある．

適応度比例戦略：ルーレット・モデルまたはモンテカルロ・モデルとも呼ばれ，各個体の適応度に比例した確率で子孫を残せる可能性がある戦略である．

ランク戦略：適応度に応じたランクを各個体に与え，あらかじめ各ランクに対して定めた確率で子孫を残すようにする．

トーナメント選択戦略：集団から定められた数（トーナメントサイズ）の個体は無作為に抽出し，その中で最も適応度の高い個体を次世代に残すことを，残したい個数が選択されるまで続ける．トーナメントサイズは2とすることが一般的である．

エリート保存戦略：集団中で最も適応度の高い個体をそのまま次世代に残す方式である．この方式では，その時点で最も良い解が交叉，突然変異の操作によって破壊されないという利点がある．ただし，エリート個体の遺伝子が集団中に急速に広がるため，局所解に陥る危険性もある．一般的にこの戦略は，他の選択戦略と組み合わせて用いられる．

適応度をそのまま選択時の確率に反映させる必要はなく，何らかの関数を導入して適応度の違いを拡大または縮小させることもある．このような関数を導入することを，スケーリングと呼び，表 2-3 に示すような方法がある．表 2-3 で， $f$  は元の適応度， $f'$  は新たな適応度であり， $\sigma$  は標準偏差である．また， $a, b, c, k$  は定数である．

個体 A :	1	0	0	1	1	1	→	1	0	1	1	0	0
個体 B :	0	0	1	1	0	0	→	0	0	0	1	1	1

図 2-39 単純交叉・1点交叉の例

個体 A :	1	0	0	1	1	1	→	1	0	1	1	1	1
個体 B :	0	0	1	1	0	0	→	0	0	0	1	0	0

図 2-40 2点交叉の例

親 A	1	0	0	1	1	1
親 B	0	0	1	1	0	0
マスク	0	1	1	0	0	0
子 A	1	0	1	1	1	1
子 B	0	0	0	1	0	0

図 2-41 一様交叉の例

### 2.6.3 交叉

交叉 (crossover) は二つの親の染色体を組み合わせることで子の染色体を生成する操作である。代表的な方法には以下がある。

**単純交叉・1点交叉** 最も単純な方法であり、交叉する位置を一つ決めて、その前と後でどちらの親の遺伝子型を受け継ぐかを変える方法である。図 2-39 に例を示す。この例では交叉位置が2番目と3番目の間にあり、個体 A の染色体の先頭から2番目までと個体 B の3番目から最後までが、新しい個体の遺伝子となり、その逆がもう一つの新しい個体の遺伝子となる。

**複数点交叉** 複数点交叉とは、交叉位置が複数ある方法である。図 2-40 に2点交叉の例を示す。この例では、交叉位置は2と4であり、新たな個体の一つは、個体 A の先頭から2番目まで、個体 B の3番目から4番目まで、個体 A の5番目から最後までを組み合わせる。他の新しい個体は逆の組み合わせで生成される。

**一様交叉** 交叉時にマスクを利用してどちらの親の遺伝子を受け継ぐかを決定する方法である。図 2-41 に示すように、二つの親個体とマスクを設定し、マスクの値が0の場合には子 A には親 A の遺伝子をコピーし、1の場合には親 B の遺伝子をコピーする。子 B では逆を行う。

その他の交叉 問題に応じて様々な交叉手法が提案されている。巡回セールスマン

問題など向けには、部分一致交叉 (Partially Matched Crossover; PMX)、順序交叉 (Ordered Crossover; OX)、周期交叉 (Cycle Crossover; CX) などが提案されている。これらは、セールスマンが巡回する都市の番号 (順列) を染色体に表現した場合に用いる交叉方式である。これらについては、第4章の「遺伝的アルゴリズムによる解法」で述べる。

#### 2.6.4 突然変異

突然変異は、遺伝子のある確率で変化させる操作である。変異確率が大きすぎるとランダムサーチに近くなるが、ある程度の変異は必要である。突然変異がない場合には、初期集団の組合せ以外の空間を探索できないため、求まる解の品質にも限界が生ずる。

一般に、固定された確率で各遺伝子が変化するように設定するが、変異率を動的に変化させる方法もある。

#### 2.6.5 世代モデル

一般に、単純な GA では、総ての個体が一斉に子孫を作り、次世代集合を生成するという「離散世代モデル」を採用している。これとは逆に「連続世代モデル」に基づくものもある。「世代間ギャップ」と呼ばれるパラメータは、現世代の何割が入れ替わるかを示すパラメータであり、集団数が100で、世代間ギャップが0.1なら、10の個体が交代する。

定常状態 GA (steady-state GA) では、次世代の生成には、2個体のみを選択し、二つの子孫の個体を作り、適応度の低い個体二つを取り除く。

世代交代モデルには、上記の最も基本的なもの以外にも様々なものが提案されており、澤井らによるパラメータフリー GA は、交叉率、突然変異率、交叉の方式などを選択する必要のないものである [澤井 98a, 澤井 98b, 木津 99, 足立 00, 澤井 00]。

## 第3章 階層メニューの最適化問題

本章では、従来最適化問題として扱われてこなかった階層メニューの設計問題を、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、かつ使い勝手を表す指標も目的関数に採り入れて、平均到達時間と使い勝手を表す指標の加重和を最小化する問題として定式化する [Matsui 07, Matsui 08a, Matsui 08b, Matsui 08c, 松井 08a, 松井 08b, 松井 08c] .

定式化では、ユーザが理想的な行動をした場合の平均到達時間を考える．すなわち、ユーザは目的とする機能を誤りなく選択することを前提とする．つまり、ユーザは目的の機能のあるサブメニューまで誤りなく辿り着くものとし、サブメニュー内の項目を誤りなく選択するものと仮定する．この仮定の下で、与えられたメニューの構造を、ユーザの利用頻度に従って変更する問題として定式化する．

ポインティング時間は、Fitts の法則の法則に従うものとして定式化する．探索・意思決定時間は、探索時間と意思決定時間からなり、(1) 熟練者は項目の配置を記憶していることから、探索時間を 0 と仮定できることから、Hick-Hyman の法則を用いて意思決定時間をモデル化する．(2) 初心者では、探索時間を無視できないことから、項目数に比例する時間がかかるとしてモデル化する．

使い勝手を表す指標として「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．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．

また、最適化問題が非常に複雑な組み合わせ最適化問題であることも示す．

### 3.1 定式化の概要

メニュー項目への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 (図 3-1) . 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る．この木構造において、根が操作開始点に相当し、メニュー項目はノードに配置されるものとする．図 3-1 に示すように、

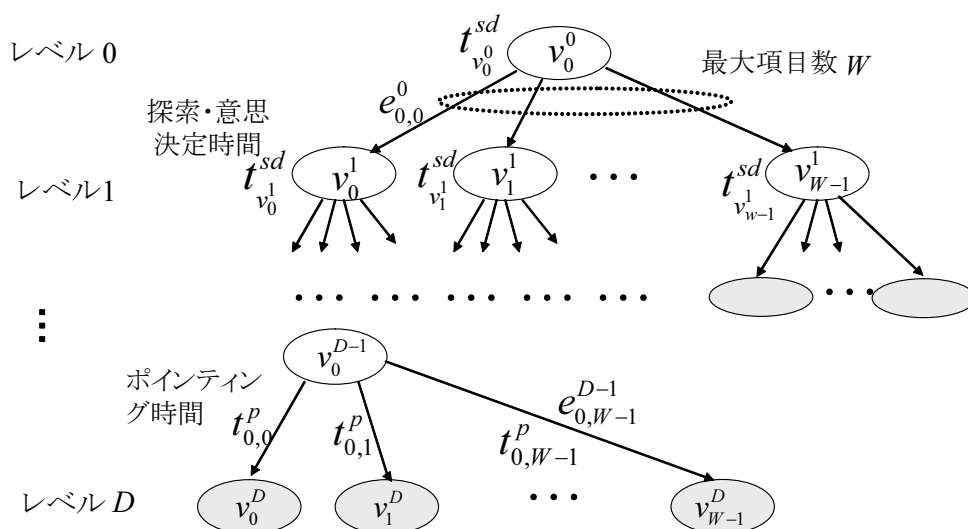


図 3-1 階層メニューの木構造表現

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．よって，メニューの設計問題は，機能に対応したノード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問題となる．

ただし，使いやすさの観点からは，効率だけを考えた項目の配置は望ましくなく，項目の意味を考慮する必要がある [Amant 07]．例えば，「音量調整」メニューは「設定」の下への配置が自然であり，「メール」の下に配置されると記憶が困難となり，使い勝手を損ねる．また，サブメニュー毎の項目数が大きく異なるメニューでは，メニューの粒度が異なるため，使い勝手を損ねると考えられる [Tange 98]．これら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．例えば，「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．

## 3.2 標記法

木構造の階層番号（レベル）を  $l$  で，その階層におけるノードの順番を  $i$  で，ノードを  $v_i^l$  で表す（図 3-1）． $V = \{v_i^l\}$  がノード集合を， $E = \{e_{i,j}^l\}$  ( $e_{i,j}^l$  はノード  $v_i^l$  とノード  $v_j^l$  の間の辺）を， $t_{i,j}^l$  はノード  $v_i^l$  からノード  $v_j^l$  への到達時間（探索・意思決定時間）を， $p_{i,j}^l$  はノード  $v_i^l$  からノード  $v_j^l$  へのポインティング時間を表す．



ド  $v_j^{l+1}$  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. 機能が配置されるノードを「終端ノード」と呼び (図 3-1 でグレーに塗られたノード), 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. 終端ノード  $v_k^T$  の機能が使われる利用頻度は選択確率  $H_k$  で与えられるものとする. メニュー項目を  $I_k$  で表現し, その総数は  $N$  とする, つまり, 機能の数は  $N$  以下である.

### 3.3 到達時間

目的とする終端ノードに辿り着くためには, 根ノードから順に木を辿る. 各ノードで項目を選ぶための探索・意思決定が行われ, 選ばれた項目をポインティングする (枝を辿る) ものとする. したがって, ノード  $v_j^{l-1}$  を経由してノード  $v_i^l$  へ到達する時間  $t_{v_i^l}$  は, ノード  $v_j^{l-1}$  での探索・意思決定時間  $t_{v_j^{l-1}}^{sd}$  と枝  $e_{j,i}^{l-1}$  を辿るためのポインティング時間  $t_{j,i}^p$  を用いて

$$t_{v_i^l} = t_{v_j^{l-1}}^{sd} + t_{j,i}^p \quad (3.1)$$

と表現できる (ただし,  $t_0^0 = 0$ ) [Cockburn 07]. 終端ノードを選択するには, 根からレベル  $l$  まで辿り着く必要があり, 終端ノード  $v_k^T$  への到達時間  $T_k$  は根から目的ノードまでのパス上の  $t_{v_i^l}$  の総和となる. このパスを  $R = (v_0^0, \dots, v_{i_k}^k, k \leq D)$  としたとき,

$$T_k = \sum_{v \in R} t_{v_i^l} \quad (3.2)$$

となる. 平均到達時間  $T_{avg}$  は

$$T_{avg} = \sum_{k=1}^N H_k T_k \quad (3.3)$$

で表せる.

### 3.4 ポインティング時間と探索・意思決定時間

Silfverberg ら [Silfverberg 00] と Cockburn [Cockburn 07] などと同様に, ポインティング時間  $t_{j,i}^p$  は Fitts の法則を使って以下でモデル化する.

$$t_{j,i}^p = a^p + b^p \log_2(A_i/W_i + 1) \quad (3.4)$$

ここで,  $A_i$  はポイントするターゲットまでの距離,  $W_i$  はターゲットの幅を意味する. また,  $\log_2(A_i/W_i + 1)$  は「困難度の指標」(Index of difficulty; ID) 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.

レベル  $l$  にある  $n$  個の項目を持つノードでの探索・意思決定時間  $t_{v_i^l}^{sd}$  は探索時間と意思決定時間からなり, 以下のように表現できる [Cockburn 07].

- 熟練者は項目の配置を記憶していることから，探索時間を 0 と仮定でき，意思決定時間だけを考えればよい．意思決定時間は Hick-Hyman の法則に従うことから，以下でモデル化する．

$$t_{v_i^l}^{sd} = b^{sd} \log_2 n + a^{sd} \quad (3.5)$$

$n$  個の項目数の選択確率が均一でない場合には，項目  $i$  の選択確率を  $p_i$  として，

$$t_{v_i^l}^{sd} = b^{sd} \log_2(1/p_i) + a^{sd} \text{ となる．}$$

- 初心者では，探索時間を無視できないことから，以下でモデル化する．

$$t_{v_i^l}^{sd} = b^{sd} n + a^{sd} \quad (3.6)$$

それぞれのモデルの係数  $a^{sd}, b^{sd}$  は実験データの回帰式で決定する．

項目数が少ない場合には，目的とする機能が配置されているメニューまでの経路は覚えていても，メニュー内の順番まで完全に記憶していることは稀だと考えられる．そこで，本論文では探索・意思決定時間のモデルには初心者のモデルを使うこととする．

### 3.5 機能の類似度

携帯電話のメニュー構造を変更して平均到達時間を短縮した Amant らの論文では，使い勝手を損なわないように，項目を配置する位置を階層の上に移動する方式を採用している [Amant 07]．提案手法では，より大きな変更が可能で，かつ使い勝手を損なわない配置となるように，似た機能が同一のサブメニューに配置されるようにするために機能の類似度を導入する．

Toms らは機能を説明する文書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 [Toms 01]．この方法は，設計段階では有効であるが，複数人による機能の関連付けが必要があるため，提案する手法で類似度を計算するための方法としては適さない．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  $I_x$  にはキーワード群  $wl_x = \{w_1, w_2, \dots\}$  が付加されているとする．具体的なキーワードの付加例は，後述の実験の章で示す．また，キーワード全体の集合を  $WL = \bigcup_x wl_x$  で表現し，キーワードに 0 から始まる  $|WL| - 1$  までの番号を付ける．項目  $I_x$  に付加されたキーワード  $w_i$  の出現頻度を表すベクトル（以下出現頻度ベクトルと略記）を  $I_x$  とする．中間ノードの出現頻度ベクトルは，子として持つメニュー項目の出現頻度ベクトルの総和とする．項目  $I_y$  についてのベクトルを  $I_y$  としたとき，機能の類似度  $s(I_x, I_y)$  の定義は以下とする．

$$s(I_x, I_y) = \frac{I_x \cdot I_y}{|I_x| |I_y|} \quad (3.7)$$

この類似度は情報検索の分野で広く使われている，語の出現頻度ベクトルを用いた文書ベクトル間の余弦距離である [Baeza-Yates 99]． $m$  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  $P_{v_i^l}^s$  を，

$$P_{v_i^l}^s = \sum_{x=0}^{m-1} \sum_{y=0}^{m-1} (1 - s(I_x, I_y)) \quad (3.8)$$

と定義する．

全体のペナルティ  $P^s$  は，

$$P^s = \sum_{v_i^l \in \{V \setminus v_0^0\}} P_{v_i^l}^s \quad (3.9)$$

とする．例えば，図 1-1 の (b) を想定した場合に，「ファイル (F)」と「編集 (E)」といったまとまりの間の類似度を考慮しても意味がないため，根ノードは含めない．

### 3.6 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，

$$g_{v_i^l} = \sum_{j=0}^{m-1} g_{v_j^{l+1}} \quad (3.10)$$

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，

$$P_{v_i^l}^g = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} |g_{v_i^l} - g_{v_j^l}| \quad (3.11)$$

と定義する．

全体としてのペナルティ  $P^g$  は

$$P^g = \sum_{v_i^l \in \{V \setminus v_0^0\}} P_{v_i^l}^g \quad (3.12)$$

とする．機能の類似度と同じ理由により，全体のペナルティの計算には根ノードは含めない．

### 3.7 最適化問題

本論文が扱う最適化問題の目的関数は以下となる．

$$f = T_{avg} + \alpha P^s + \beta P^g, \quad (3.13)$$

ここで  $\alpha$  と  $\beta$  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．

ノード  $v_i^l$  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，利用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，平均到達時間は最小となる．したがって，最適化問題は，式 (3.13) を最小化するようなメニュー項目のノード  $v_i^l$  への割当を求める問題となる．ただし，第2章「関連研究」でも述べたように，階層メニューでは1ノードあたりの項目数が少なく，最大レベルが大きなメニュー（以下「深いメニュー」と略記）よりも，1ノードあたりの項目数が多く，最大レベルが小さなメニュー（以下「浅いメニュー」と略記）の方が，平均到達時間が短いと報告されている [Kiger 84, Landauer 85, Schultz 86, Larson 98, Zaphiris 00, Zaphiris 03, Parush 04]．例えば，機能の数が64の場合には，1ノードに4項目とした3レベルのメニューよりも，1ノードに8項目とした2レベルの，より浅いメニューの方が平均到達時間が短いことが報告されている．そこで，提案方式では浅いメニューが生成されるような解法を用いる．具体的な解法については，後述する．

$N$  個の項目を最大  $W$  個の子を持ついずれかのノードに割り当てて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．したがって，探索空間の大きさは，ほぼ

$${}_N C_W \times {}_{N-W} C_W \times \cdots \times {}_{N-LW} C_W = N! / (W!)^L \quad (3.14)$$

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．

表3-1に探索空間の大きさの例を示す．表から分かるように， $N$  の増大に従って探索空間は急激に増大する．また， $N$  が同じであれば， $W$  が大きい方が探索空間は小さい． $N$  が50以下であれば全数探索可能な探索空間であるが，それ以上であれば，全数探索は困難（実際的には不可能）である．

表 3-1 探索空間の大きさの例

$N$	$W$	探索空間の大きさ
30	9	$\approx 10^{10}$
30	12	$\approx 10^6$
50	9	$\approx 10^{31}$
50	12	$\approx 10^{21}$
100	9	$\approx 10^{96}$
100	12	$\approx 10^{79}$
200	9	$\approx 10^{247}$
200	12	$\approx 10^{227}$

### 3.8 まとめ

従来最適化問題として扱われてこなかった階層メニューの設計問題を，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，かつ使い勝手を表す指標も目的関数に採り入れて，平均到達時間と使い勝手を表す指標の加重和を最小化する問題として定式化した．

定式化では，ユーザが理想的な行動をした場合の平均到達時間を想定した．すなわち，ユーザは目的とする機能を誤りなく選択することを前提とした．つまり，ユーザは目的の機能のあるサブメニューまで誤りなく辿り着くものとし，サブメニュー内の項目を誤りなく選択するものと仮定した．この仮定の下で，ユーザの利用頻度に従ってメニュー設計する問題として定式化した．

ポインティング時間はFittsの法則の法則に従うものとして定式化した．探索・意思決定時間は探索時間と意思決定時間からなり，(1) 熟練者は項目の配置を記憶していることから，探索時間を0と仮定できることから，Hick-Hymanの法則を用いて意思決定時間をモデル化した．(2) 初心者では，探索時間を無視できないことから，項目数に比例する時間がかかるとしてモデル化した．

使い勝手を表す指標として「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した．機能の類似度は，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導入した尺度であり，メニュー項目間の機能の類似性を表現するものである．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，情報検索の分野で広く用いられている語の出現頻度ベクトルを用いた文書ベクトルの余弦距離として定義した．

メニューの粒度は，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に導入した尺度であり，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した．

また，最適化問題が非常に複雑な組み合わせ最適化問題であることも示した．



## 第4章 遺伝的アルゴリズムを用いた解法

前章で述べたように，対象とする問題は，非常に複雑な組み合わせ最適化問題であり，整数計画法などの厳密解法では解を求めるために，非常に長い時間が必要である．そこで，良い近似解が短時間で求まる解法を検討した．提案する解法は，遺伝的アルゴリズム (Genetic Algorithm; GA) に基づくものである．

遺伝的アルゴリズムによる解法では，解を染色体として表現するが，この表現方法の適否により，解法の性能が大きく変化する．提案する手法では，メニュー項目を木に割り当てる順番（順列）を染色体に表現する．以下では，この方式について述べた後で，提案する解法で用いた，交叉の方式などについて述べる．

### 4.1 染色体表現

#### 4.1.1 基本的方式

第2章の関連研究で述べたように，遺伝的アルゴリズムに基づく解法では，染色体への解の表現方法により，解法の性能が大きく異なる．また，遺伝子型から表現型への変換方法も解法の性能に影響する．

遺伝子型と表現型が1対1に対応し，総ての遺伝子型が表現型において実行可能解となることが望ましいが，制約条件付きの組み合わせ最適化問題などでは，このような遺伝子表現が困難な場合が多い．このため，一般的には以下の方式がとられている．

- 実行可能解とならない遺伝子（致死遺伝子）が生じないようにコーディングを用いて，遺伝的操作でも致死遺伝子が生じないようにする．
- 遺伝的操作では致死遺伝子の発生を許し，致死遺伝子が生じた時点で修復操作を施す．

単純な操作で修復が可能であれば，後者の方式が採用されることが多いが，対象とする問題では，単純に修復できないことから，前者の方式を採用することとした．

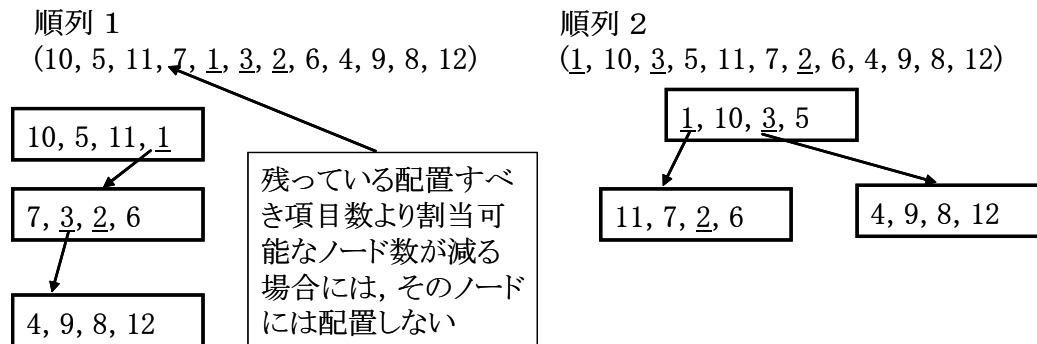


図 4-1 順列から木構造への変換方式

### 4.1.2 割当順序表現

第 2.4.1 節で述べた先行研究によれば，一般的には深いメニューよりも浅いメニューの方が平均到達時間が短いとされている [Kiger 84, Landauer 85, Schultz 86, Larson 98, Zaphiris 00, Zaphiris 03, Parush 04]．ただし，携帯電話などの画面が小さい場合には，あまりに幅が広いメニューは好ましくないとされている [Geven 06]．そこで，提案方式では幅はパラメータとして与え，その範囲内で浅いメニューが生成されるように，ノードに項目を割り当てていく割当順序を求める方式を採用した．具体的には，以下の方式で割り当てていく，割当順序を染色体に表現する．

1. 中間ノードを含むメニュー項目  $I_i (i = 1, \dots, N)$  に 1 から番号を付け，染色体はその番号の順列とする．機能に相当する項目数を  $M$  としたときには，中間ノードの数は最低限  $L = \lceil M/W \rceil$  個必要である．そこで，例えば 1 から  $L$  までは中間ノードとし， $L+1$  から  $N$  までを機能に対応するメニュー項目とする．
2. 染色体の順列に従って，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．
3. 割当が完了した時点で，子供を持たない中間ノードは削除する．

各ノードが持てる子の最大数 ( $W$ ) が 4 の例を図 4-1 に示す．この例は，中間ノードを 3 個 (番号: 1~3) 用意した場合であり，中間ノードには下線を付けてある．

“順列 1”では，メニュー項目“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幅が最大(4)となり，木はそれ以上大きくできないため，割当不能である．したがって，“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．割当が完了した時点で，子供を持たない“2”は削除する．



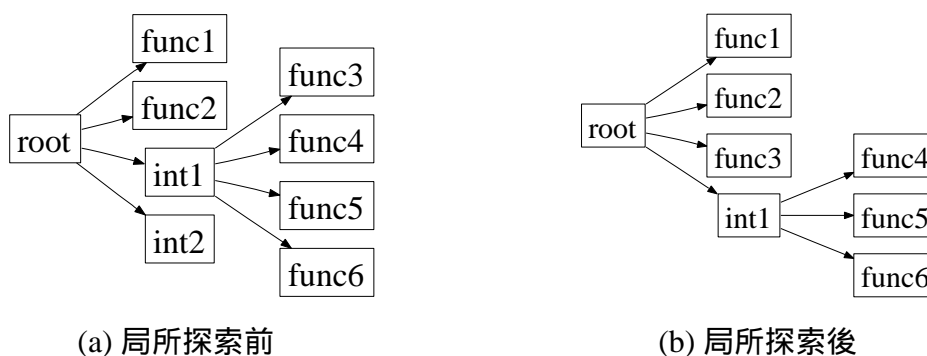


図 4-2 局所探索

“順列 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 個の空きが増える．したがって，“10” 以降のメニュー項目は順次割当可能である．“順列 1” の例と同様に，割当が完了した時点で，子供を持たない “2” は削除する．

### 4.1.3 表現の完全性

総てのメニュー項目を割り当て可能なノード数を持つ木を考える．ノードへ割当てられているメニュー項目数が  $W$  以下なら，不足する分の中間ノードが割当てられているとする．根から順に幅優先で木を辿った結果得られるメニュー項目の番号の列は，一意に決まる順列となる．したがって，上述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．

また，順列用に用いられる交叉方式，突然変異の方式を用いることで，GA での遺伝的操作において，実行不能解，致死遺伝子が生じないようにできる表現方式となっている点が特徴の一つである．

## 4.2 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，平均到達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．図 4-2 に例を示す．この場合，図 4-2(a) の木構造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，図 4-2(b) の木構造とする．局所探索は，初期集団を含めた新たに生成された個体総てに適用する．

1. 親を2点で切断する .
  - $p_1 = (1\ 2\ 3\ |\ 4\ 5\ 6\ 7\ |\ 8\ 9)$
  - $p_2 = (4\ 5\ 2\ |\ 1\ 8\ 7\ 6\ |\ 9\ 3)$
2. 子供に中央部分をコピーする .
  - $c_1 = (x\ x\ x\ |\ 1\ 8\ 7\ 6\ |\ x\ x)$
  - $c_2 = (x\ x\ x\ |\ 4\ 5\ 6\ 7\ |\ x\ x)$
3. 中央部分の対応関係を作成する .  $1 \leftrightarrow 4, 8 \leftrightarrow 5, 7 \leftrightarrow 6, 6 \leftrightarrow 7$  がマッピングの系列となる .
4. 重複しない, 衝突しない部分は親から受け継ぐ .
  - $c_1 = (x\ 2\ 3\ |\ 1\ 8\ 7\ 6\ |\ x\ 9)$
  - $c_2 = (x\ x\ 2\ |\ 4\ 5\ 6\ 7\ |\ 9\ 3)$
5. 残りはマッピング系列を用いて, 対応する要素で埋める .
  - $c_1 = (4\ 2\ 3\ |\ 1\ 8\ 7\ 6\ |\ 5\ 9)$
  - $c_2 = (1\ 8\ 2\ |\ 4\ 5\ 6\ 7\ |\ 9\ 3)$

図 4-3 PMX の例

### 4.3 交叉の方式

実行不能解を生成しないためには, 適切な交叉方式を用いる必要がある. このための交叉方式としては, 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. 巡回セールスマン問題において, セールスマンが巡回する都市の番号 (順列) を染色体に表現した場合のために考案された交叉方式としては, 部分一致交叉 (Partially Matched Crossover; PMX), 順序交叉 (Ordered Crossover; OX), 周期交叉 (Cycle Crossover; CX) などがある [Michalewicz 00].

以下にそれぞれの方式の概要を示す.

**PMX:** PMX は一つの親から部分順列を継承し, 残りの部分は, もう一方の親の順列中の順番と位置をできるだけ継承する方式である (図 4-3).

**OX:** OX は一つの親から部分順列を継承し, 残りの部分は, もう一方の親の順列中の位置を継承する方式である (図 4-4).

**CX:** CX は親の順列中の相対位置を保存する方式である (図 4-5).

1. 親を2点で切断する .
  - $p_1 = (1\ 2\ 3\ |\ 4\ 5\ 6\ 7\ |\ 8\ 9)$
  - $p_2 = (4\ 5\ 2\ |\ 1\ 8\ 7\ 6\ |\ 9\ 3)$
2. 子供に中央部分をコピーする .
  - $c_1 = (x\ x\ x\ |\ 4\ 5\ 6\ 7\ |\ x\ x)$
  - $c_2 = (x\ x\ x\ |\ 1\ 8\ 7\ 6\ |\ x\ x)$
3. 2番目の切断点から開始して、他の親の要素の系列を作成する。最後の要素となったら、先頭から繰り返す .
  - $c_1$  に対しては、9-3-4-5-2-1-8-7-6 が系列となる。これから、既に  $c_1$  に含まれているもの (4, 5, 7, 6) を削除する。
  - $c_2$  に対しては、8-9-1-2-3-4-5-6-7 が系列となる。これから、既に  $c_2$  に含まれているもの (8, 1, 6, 7) を削除する。
4. 上記の系列を2番目の切断点以降にコピーする。この時、染色体の最後に到達したら、先頭に戻ってコピーを行う。これにより以下となる .
  - $c_1 = (2\ 1\ 8\ |\ 4\ 5\ 6\ 7\ |\ 9\ 3)$
  - $c_2 = (3\ 4\ 5\ |\ 1\ 8\ 7\ 6\ |\ 9\ 2)$

図 4-4 OX の例

1.  $p_1=(1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9)$ ,  $p_2=(4\ 1\ 2\ 8\ 7\ 6\ 9\ 3\ 5)$  を考える。親  $p_1$  の最初の要素の 1 より開始、親  $p_2$  の 1 番目の要素は 4 なので 親  $p_1$  の 4 を取り出す。  $p_1$  の 4 の位置に相当する  $p_2$  の要素は 8 なので、  $p_1$  の 8 を取り出す。この結果つぎの子ができる .
  - $c_1 = (1\ x\ x\ 4\ x\ x\ x\ 8\ x)$
2. 上記を繰り返して  $c_1$  を作って行くと以下が得られる .
  - $c_1 = (1\ 2\ 3\ 4\ x\ x\ x\ 8\ x)$
3. 最後に入れられたのは 2 だが、この位置の  $p_2$  の要素は 1 であり、これ以上は進めない。従って残りの要素は反対側の親  $p_2$  のものをそのまま入れることで、次が得られる .
  - $c_1 = (1\ 2\ 3\ 4\ 7\ 6\ 9\ 8\ 5)$
4. 他の親に対しても同様な方法を用いると、次が得られる .
  - $c_2 = (4\ 1\ 2\ 8\ 5\ 6\ 7\ 3\ 9)$

図 4-5 CX の例

予備実験の結果に基づいて CX (Cycle Crossover) を用いることとした。交叉方式による性能の違いについては、第 8 章「考察」でさらに検討する。

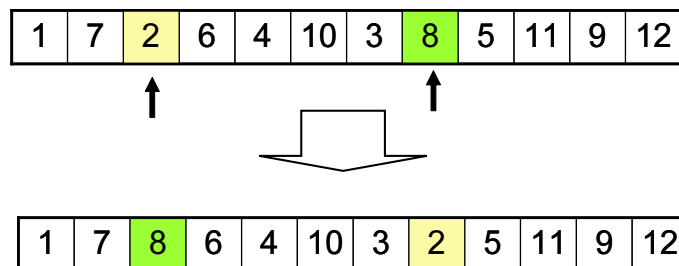


図 4-6 突然変異の方式

## 4.4 他のGAパラメータ

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  $q$  の値を入替える方式を用いる。例を図 4-6 に示す。この例では、3 番目の遺伝子座と 8 番目の遺伝子座の値「2」と「8」を入れ替えている。こ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。

個体の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一般的に用いられている、定常状態 GA [Mitchell 96] を用い、個体の選択にはサイズ 2 のトーナメント方式を用いることとした。

## 4.5 他の表現方式によるGAの構成法

提案する GA では、順列は整数値列として表現しており、致命的遺伝子が生じない交叉方式、突然変異方式を用いている。順列表現は他の方式でも可能である。例えば、Bierwirth らによる、実数値列を用い、それをソートキーとして用いて順列を表現するランダムキー方式 [Bierwirth 96] を用いることも可能である。この方式でも問題によっては、高性能な GA を構成できる [松井 02a]。あるいは、割り当てられていない項目の何番目を次に割り当てるかを表現する順序表現 [Grefenstette 85] による表現も可能である。この方式でも周波数割当問題などでは高性能な GA が構成できる [Matsui 00, 松井 00]

## 4.6 まとめ

本章では、遺伝的アルゴリズムによる解法について述べた。遺伝的アルゴリズムでは解を染色体に表現するが、この表現方法により、解法の性能が大きく変化する。提案する手法では、用いたメニュー項目を木に割り当てる順番（順列）を染色体として表現する方式について述べた。また局所探索による解の改善方式について述べた。さらに、提案する解法で用いた、交叉の方式などについて述べた。

## 第5章 被験者実験によるモデルの検証

定式化で採用したポインティング時間と探索・意思決定時間のモデルは，単一階層のメニューでは十分な近似精度であることが確認されている [Cockburn 07]．しかし，2レベル以上の階層メニューでの近似精度は明らかではない．そこで，小規模なメニューを用いて二つのモデルの近似精度を確認するための実験を行った [松井 08c]．

このために，二つの小規模なメニュー（ベースと平均到達時間を短縮した比較対象）を PDA 上に表示させ，被験者実験により，モデルの近似精度を検証した．

その結果，被験者全体としてみれば，予測値と平均値はよく合致している結果が得られた．これにより，定式化で採用したポインティング時間と探索・意思決定時間のモデルは，本研究で対象としている2レベル以上の階層メニューに対しても，十分な近似精度を持っているとの結論を得た．

### 5.1 実験方法

#### 5.1.1 メニューの生成

次章で用いたベースのメニュー（図 6-6）と提案手法で生成したメニュー（図 6-8）から，利用頻度の高い部分だけを抜き出したメニューを作成した．これらのメニューには19個の終端ノードが含まれている．以下，前者に対応するメニューを「ベース」，後者に対応するメニューを「比較対象」と表記する．具体的なメニューの構造を図 5-1 に示す．実験に用いたメニューの作成では，被験者の探索が容易になるように，項目名は，図 6-6 と図 6-8 に示したものと異なる，判別しやすいものとした．

実験で用いた画面（図 5-2）において，“Start” ボタンの大きさは  $50 \times 25$  pixels とし，“Appli”，“SC”，“Mail”，“EZweb” ボタンの大きさは  $48 \times 20$  pixels とした．ポップアップメニューの項目の高さは 16 pixels とした．これらのパラメータから困難度の指標 (ID) を計算し，ポインティング時間の計算に用いた．ポップアップメニューは対応する項目の直上に現れるように配置したため，第1項目への移動距離は0となる．

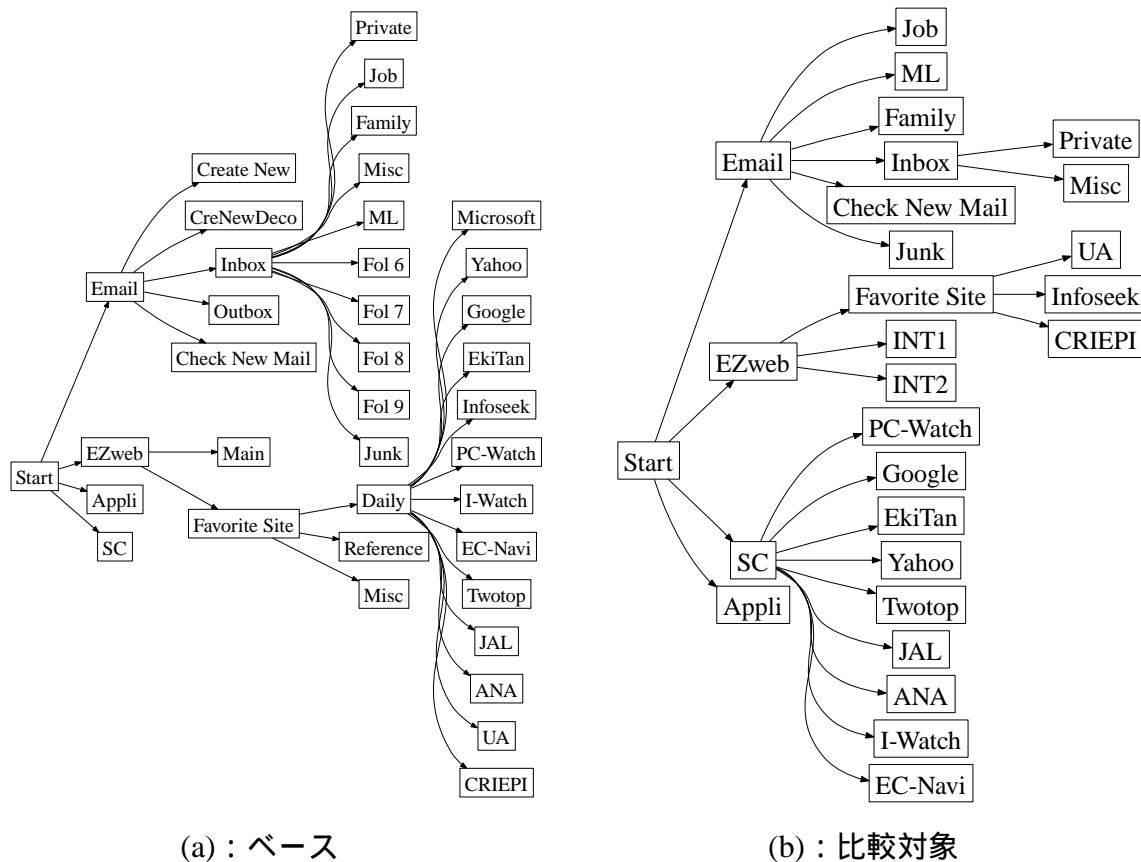


図 5-1 PDA での実験で用いたメニュー

### 5.1.2 被験者実験の方法

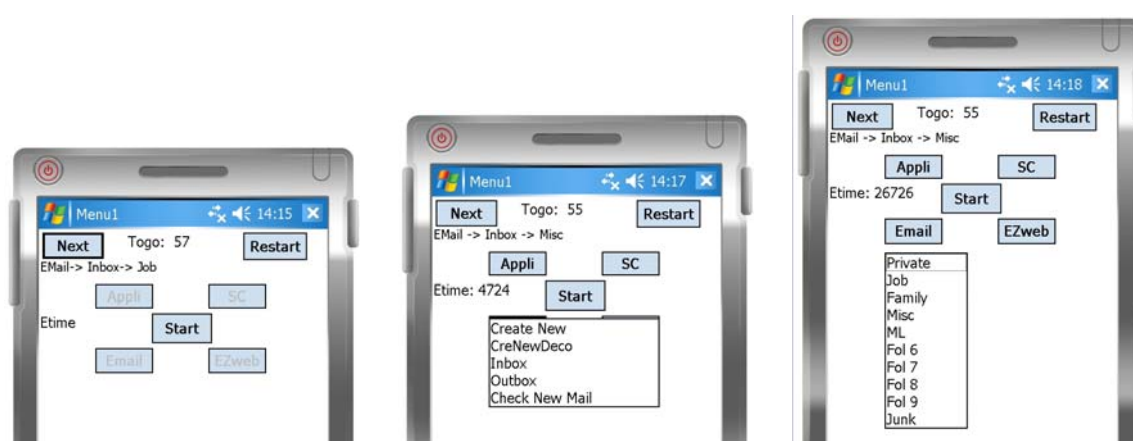
実験は、10代から50代の男女10名に対して、図5-1に示したメニューをHP iPAQ 212上に表示させ、目的とする機能に到達するまでの時間を計測した。誌面の都合上、画面の上部のみを図5-2に示したメニューにおいて、“Start”をタップしてから、目的とする機能に到達するまでの時間をミリ秒単位で計測した。

被験者は画面上部に表示されている指示に従って、メニュー階層を辿る。“Next”をタップすると、その下に、‘EZweb -> Favorite -> Daily -> ANA’のように辿るべきメニュー階層が提示される。この指示に従って、被験者は“Start”をタップしてから、目的とする機能までメニュー構造を辿る（図5-3）。

被験者は、データ計測を開始する前に、メニュー構造の大まかな把握とPDAに慣れるための練習を行った。その後で、ランダムな順番で提示される各々の終端ノードについて3回計測を行い、その平均値を到達時間とした。メニュー全体としての平均到達時間を算出するために必要な各機能の頻度分布としては、次章の実験で用いたものから生成したものをを用いた。具体的な値は付録Aの表A-1に示した。



図 5-2 PDA での実験の画面例（初期画面）



(a) Next をタップした状態 (b) Email をタップした状態 (c) Inbox をタップした状態

図 5-3 PDA での実験の画面例（メニュー遷移の例）

## 5.2 実験結果

実験結果を表 5-1 に示す．表 5-1 で「短縮率」は，ベースでの平均到達時間を  $T_o$ ，比較対象での平均到達時間を  $T_m$  としたとき， $Imp = 100 \times (T_o - T_m) / T_o$  で求めた値である．被験者として「平均」で示した行は，10 名の平均値である．「予測値」として示した行は，第 3 章で示した平均到達時間の計算式から求まる値を示す．個人毎にポインティング時間と探索・意思決定時間は異なる．すなわち，推定式の係数  $a^{sd}$ ,  $b^{sd}$ ,  $a^{pt}$ ,  $b^{pt}$  が異なるため，予測値の行には平均到達時間は示していない．次章で示すように，推定式の係数によって短縮率は大きく変化しないことから，推定式の係数には，次章で用いた値を用いた．

表 5-1 から分かるように，予測値の短縮率よりも低い短縮率しか得られない被験者もいるが，予測値以上を達成している被験者もいる．被験者の 1 名のみが日常的にこの PDA を利用しているのみであり，他の 9 名はこの実験で初めてこの PDA を利用した．この経験の差が短縮率の違いの一要因となっていると考えられる．

被験者全体としてみれば，予測値と平均値はよく合致していると考えられる．この

表 5-1 PDA での短縮率

被験者	性別	年代	平均到達時間		短縮率 (%)
			ベース (ms)	比較対象 (ms)	
1	男	50	1941	1026	47
2	女	40	3792	2294	40
3	女	10	2412	1278	47
4	男	40	2650	1639	38
5	男	30	2226	1342	40
6	男	40	2072	1373	34
7	男	30	2558	1315	49
8	男	40	3203	1625	49
9	男	30	2247	1181	47
10	男	20	1901	1167	39
平均			2562	1468	43
予測値			—	—	45

ことから，定式化で採用したポインティング時間と探索・意思決定時間のモデルは，本研究で対象としている2レベル以上の階層メニューに対しても，十分な近似精度を持っていると考えられる．

### 5.3 まとめ

定式化で採用したモデルは，単一階層のメニューでは十分な近似精度であることが確認されていた．しかし，2レベル以上の階層メニューでの近似精度は明らかではなかった．そこで，小規模なメニューを用いて二つのモデルの近似精度を確認するための実験結果について述べた．

二つの小規模なメニュー（ベースと平均到達時間を短縮した比較対象）をPDA上に表示させ，被験者実験により，モデルの近似精度を検証した．その結果，被験者全体としてみれば，予測値と平均値はよく合致している結果が得られた．これにより，定式化で採用したポインティング時間と探索・意思決定時間のモデルは，本研究で対象としている2レベル以上の階層メニューに対しても，十分な近似精度を持っているとの結論を得た．



## 第6章 シミュレーションによる最適化の評価

前章の被験者実験により，定式化で採用したモデル化の近似精度が確認できた．メニューの性能比較は，被験者実験で行われることが一般的である．これは，ユーザがエラーを起こすことを前提としているためである．これに対して，本論文では，ユーザは誤りなく選択することを前提とすることから，提案手法の有効性はシミュレーション実験で確認する．このために，多様な条件を想定してのコンピュータシミュレーションを行った．著者が日常的に利用していた携帯電話のメニューを対象とした実験を行った [Matsui 07, Matsui 08a, Matsui 08b, Matsui 08c, 松井 08b, 松井 08c, 松井 08a] ．

本章では，まず実験に用いたデータについて述べる．次に，携帯電話の端末の階層メニューを対象に，複数の利用パターンを用いて，提案する解法の性能をシミュレーションにより確認した結果を述べる．また，多様な利用パターンを想定して，Zipf 分布を用いて生成した利用パターンにより，提案手法の頑健性を検証した結果についても述べる．さらに，計算時間についても述べる．

### 6.1 実験に用いたデータ

#### 6.1.1 ポインティング時間

対象とした携帯電話の端末 [KDDI 04] は 24 個のキーが図 6-1 に示すように配置されている．24 個のキーについて，その中心座標，幅，高さを計測し， $24 \times 24$  のキーの組合せに対する困難度の指標 (ID) を求めた（具体的な値は付録 B.1 の表 B-1 ～ 表 B-7 に示す）．小数点以下 1 桁の精度で ID を丸めた結果，28 個のグループが得られた（表 6-1）．表 6-1 では，図 6-1 に示したボタンを，上から順に，左から右の順で “App”，“Up”，“Addr”，“Left”，“Center”，“Right”，“Mail”，“Down”，“Web”，“Call”，“Clear”，“Power”，“1”，“2”，“3”，“4”，“5”，“6”，“7”，“8”，“9”，“\*”，“0”，“#” と表記している．

親指 1 本で操作を行う無報酬の被験者の協力を得て，28 個のキーの組合せに対して，キークリックの音を録音することで，ポインティング時間を計測した [Amant 07] ．得られたデータから回帰直線を求めたところ，ポインティング時間の予測式として

$$t_i^p = 192 + 63 \log_2(A_i/W_i + 1) \text{ (ms)} \quad (6.1)$$

が得られた．この式の係数は先行研究 [Silfverberg 00] の次式の係数に近い．

$$t_i^p = 176 + 64 \log_2(A_i/W_i + 1) \text{ (ms)} \quad (6.2)$$

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から次の階層に遷移するために押すキーまでの ID によって決まるものとする．対象とした端末には，メニュー項目によっては，数字番号が割当てられており，その番号を選ぶことで目的の項目を選択できる機能がある．しかし，頻度データを収集した利用者はこの機能をほとんど利用していなかったことから，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．

### 6.1.2 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は，実験に用いた端末では困難であったこと，ポインティング時間が先行報告とよく合致することから，探索・意思決定時間には利用者としては初心者を想定し，先行研究 [Cockburn 07] の

$$t_i^{sd} = 80n + 300 \text{ (ms)} \quad (6.3)$$

を用いた．

### 6.1.3 利用頻度と機能の類似度

対象とした携帯電話がスマートフォンであれば，Raento らのようにプログラムを追加することで，履歴収集は容易である [Raento 05]．しかしながら，対象とした携帯電話は履歴を収集するためのプログラムを追加できないものであったことから，利用頻度データは著者の 1 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．この履歴には，携帯電話に予め組み込まれている機能に加えて，利用者が追加した受信メールの整理用フォルダ，Web のお気に入りサイトが含まれる．このデータには総計 129 項目（機能，メールのフォルダ，Web のお気に入り）が含まれていた（具体的な利用頻度は付録 B.2 の表 B-8～表 B-10 に示す）．このデータを“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 種類の頻度データも生成した．

**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 [JIPDEC 07][MMD 07] などでも報告されているように、利用者の特性によって携帯電話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、または50通以上と高頻度な場合を想定したデータが Mail2 と Mail3 である。また、他と比べて SNS やブログの利用頻度が高いユーザを想定したデータが Web2 と Web3 である。

機能の類似度計算に用いるキーワードは、マニュアル [KDDI 04] にあるメニューに表示される単語（英単語）を用いて割り当てた。具体的には、上位階層から目的とする機能に辿り着くまでの、メニュー項目として表示される英単語を並べたものを、機能に対応したキーワードとした。例えば、“E-Mail -> Inbox -> Folder0”の場合には、{ “E-Mail”, “Inbox”, “Folder0” } の英単語を “Folder0” のキーワードとした。



図 6-1 対象とした携帯電話のキー配列

表 6-1 28 個のグループ毎の ID の値

group #	ID	example pairs		# of pairs
		from	to	
1	3.7	*	Up	2
2	3.6	0	Up	3
3	3.5	9	Up	6
4	3.4	8	Up	8
5	3.3	8	Right	17
6	3.2	9	Down	22
7	3.1	8	Down	25
8	3.0	6	Right	28
9	2.9	1	Up	29
10	2.8	8	Center	29
11	2.7	1	*	33
12	2.6	2	Right	29
13	2.5	1	9	29
14	2.4	1	0	53
15	2.3	1	3	33
16	2.2	2	Center	20
17	2.1	1	8	25
18	2.0	2	Call	17
19	1.9	1	7	21
20	1.8	Mail	Call	7
21	1.7	1	5	50
22	1.6	1	2	16
23	1.4	2	Clear	9
24	1.3	Right	Up	12
25	1.2	1	4	21
26	1.1	Center	Down	4
27	0.8	Right	Center	4
28	0.0	1	1	24

### 6.1.4 計算条件の設定

予備的な計算実験の結果に基づいて、以下で述べる総てのケースにおいて、次の条件で計算を行った。GA の個体数は 100 とし、突然変異は一对の交換とした。個体評価回数（生成されたメニューの評価回数）が上限の 100,000 回に達した時点で計算終了とした。GA は確率的な計算であることから、50 回の試行を行った結果の平均と標準偏差を求めた。ペナルティ項のウェイトの影響を調べた実験以外では、 $\alpha = 10.0$ 、 $\beta = 1.0$  とした。

## 6.2 性能向上の実験結果

以下に示すケースについて計算を行った。

基本ケース：提案手法によりどの程度の時間短縮が得られるかを求めた。1 ノードあたりの最大項目数は 16 とした。以下これを  $W = 16$  と表記する。ただし、対象とした携帯電話に備わっている「メール」、「Web」、「アプリ」、「QuickAccess」（一種のショートカット）キーの数に合わせて根ノードの項目数は 4 とした。

最大項目数を限定：先行研究結果 [Kiger 84, Schultz 86, Zaphiris 03] によれば、広い画面であれば、深いメニューよりも浅いメニューが好まれることが報告されている。また、携帯電話などの狭い画面では、余り広いメニューは好まれないことが報告されている [Geven 06]。

ある程度の広さを持つ画面であっても、ユーザによっては幅が広くなりすぎて、「下」カーソルを何度も押すことが好まれない場合も考えられる。そこで、最大項目数を 12, 9, 6 と変化させて計算を行った。以下これらのケースを  $W = 12$ 、 $W = 9$ 、 $W = 6$  と表記する。このケースでも、上述と同様に根ノードの項目数は 4 とした。

前方移動：利用頻度データを計測した時点でのメニュー構造（以下これをベースと表記）を元として 1 ノード内の配置を利用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。これは Self-Organizing Heuristics の一種である move-to-front [Atallah 98] であり、広く使われている。以下これを前方移動と表記する。

表 6-2 に各ケースの短縮効果を示す。また図 6-2 ~ 図 6-5 に 50 回の計算結果を箱ヒゲ図の形式で示す。表 6-2 において  $T_{ave}$  は平均到達時間であり、“ $Imp(\%)$ ” はベースからの短縮率を示す。また、 $P^s$  と  $P^g$  はそれぞれ機能の類似度のペナルティー、メニューの粒度のペナルティである。また、括弧内の数値は標準偏差を表す。

表 6-2 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。前方移動と提案手法の短縮率の差は 30 ポイント程度ある。最大項目数を限定することにより、 $W = 9$  までは平均到達時間が短くなる。これは、ノードに割り当てられた項目数を  $n$

表 6-2 平均到達時間の短縮効果

ケース	$T_{ave}(\text{ms})$	$Imp(\%)$	$P^s$	$P^g$
ベース	4941		399	793
前方移動	4423	10.5	399	793
$W = 16$	3030 (115)	38.7 (2.3)	677 (15)	1107 (22)
$W = 12$	2755 (76)	44.3 (1.5)	495 (14)	727 (44)
$W = 9$	2625 (46)	46.9 (0.9)	355 (6.2)	187 (23)
$W = 6$	3007 (85)	39.1 (1.7)	230 (4.3)	121 (18)

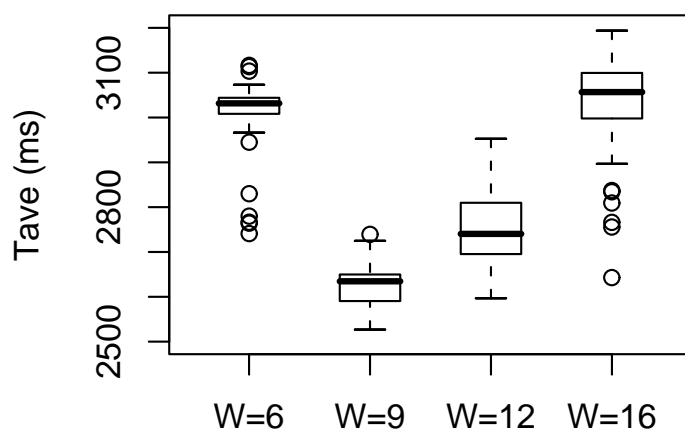


図 6-2 計算結果のばらつき (平均到達時間)

として、 $n$  に比例する形で探索・意思決定時間が増加することが影響している。最大項目数を 6 まで小さくした場合には ( $W = 6$ )、深いメニューとなるため、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 6-2 からは、平均到達時間のばらつきは  $W = 9$  の場合が最も小さく、 $W = 6, W = 16$  の場合のばらつきが大きいことが分かる。特に、平均値より短い時間となるメニュー構成が求まる割合が多いことが分かる。このことは、 $W$  によって適応度の景観が大きく変化することに依っていると考えられる。図 6-3 も、図 6-2 と同様に、短縮率のばらつきも  $W = 9$  の場合が小さいことが分かる。

図 6-4 からは、機能の類似度のばらつきは、 $W = 16$  の場合に大きいことが分かる。図 6-5 からは、メニューの粒度のばらつきは  $W$  に依らずにさほど大きくないことが分かる。

元のメニューを図 6-6 に、前方移動方式で生成したメニューを図 6-7 に、 $W = 9$  で最も目的関数の値が小さかったメニュー ( $T_{ave} = 2605 \text{ ms}$ ) の例を図 6-8 示す。図では紙面の制約のために、低利用頻度のノードのラベルは省略して矢印のみ示してある。また、同一レベルであれば、上にあるノードほど利用頻度が高いように並べてある。

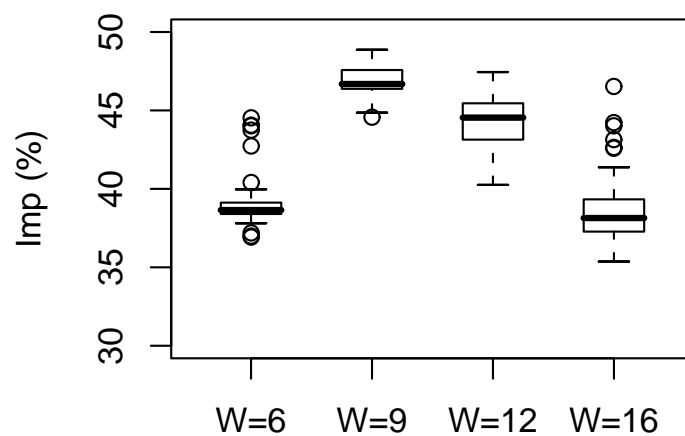


図 6-3 計算結果のばらつき（短縮率）

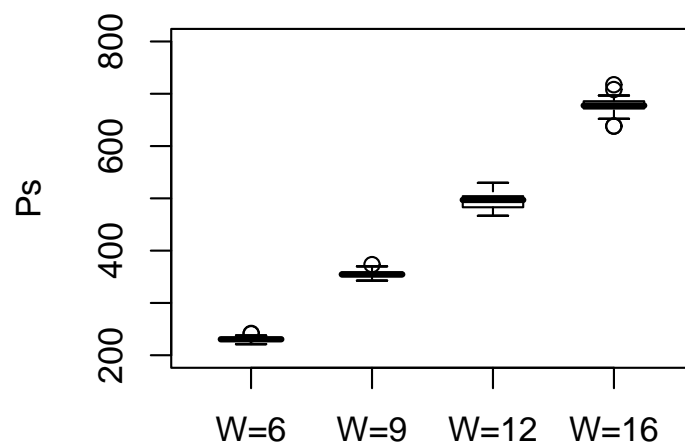


図 6-4 計算結果のばらつき（機能の類似度のペナルティ）

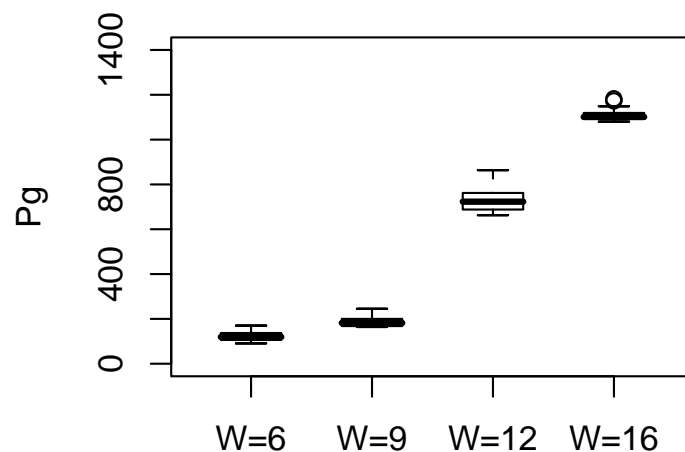


図 6-5 計算結果のばらつき（メニューの粒度のペナルティ）

図 6-6, 図 6-7, 図 6-8 を比較すると, 提案手法で生成されたメニューでは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”の一種である“QuickAccess”が有効活用されていることが分かる。また, 前方移動方式では, 利用頻度の高い項目が上のノードに移動されているが, 階層をまたがっての移動, “QuickAccess”への移動ができないことから, 提案手法とは異なったメニュー構造となっていることが分かる。

さらに比較のために,  $W = 6$ ,  $W = 12$ ,  $W = 16$  で生成されたメニューの例を図 6-9 から図 6-11 に示す。

“QuickAccess”への割り当てを見てみると,  $W = 6$ を除いて, 中間ノードが作られることなく, 利用頻度の高い URL が配置されている。“E-Mail”と“EZ-Web”の下に配置を見てみると,  $W = 6$ を除いて, 高利用頻度の項目が直下に配置されている。 $W = 6$ の場合には直下に配置できない高利用頻度の項目があるために,  $W = 9$ の場合よりも, 平均到達時間が長くなる。 $W = 12$ ,  $W = 16$ の場合には, 浅いメニュー構造とはなるが, 項目数が多くなることにより, 探索・意思決定時間が長くなるために,  $W = 9$ よりも平均到達時間が長くなる結果となっている。

$W = 16$ の場合には, 高利用頻度の項目はすべて“E-Mail”, “EZ-Web”, “QuickAccess”の直下に配置されており, 中間ノードの下に配置されていない。しかしながら, 項目数が多くなることにより, 探索・意思決定時間が長くなるために,  $W = 9$ よりも平均到達時間が長くなる結果となっている。また, 高利用頻度の項目が浅い階層に割り当てられるために, 低利用頻度の項目は中間ノードを介した深い階層に配置されるようになる。このために,  $W$ が大きくなるに従って, メニューの粒度のペナルティ項が大きくなる。



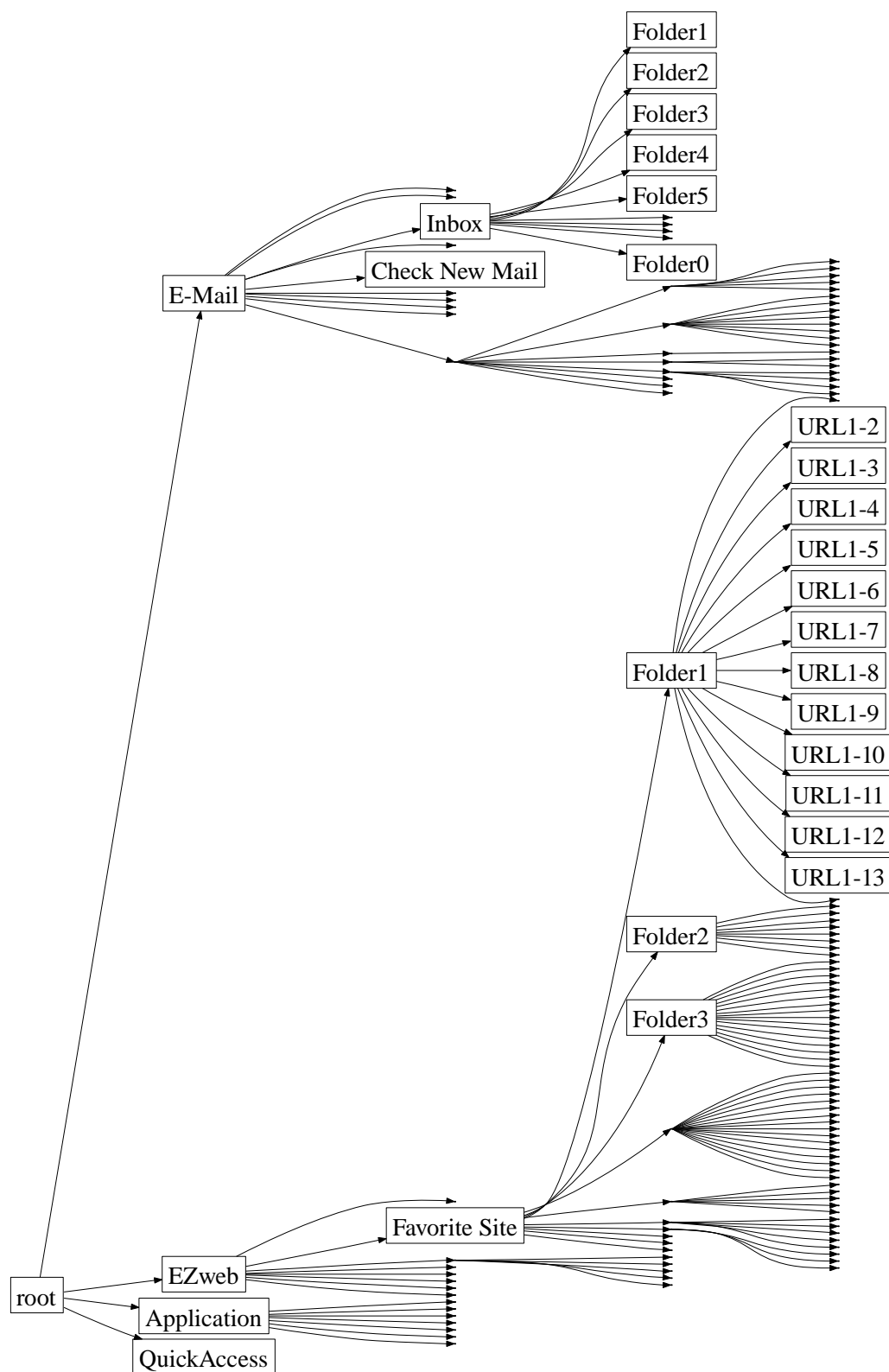


図 6-6 元のメニュー

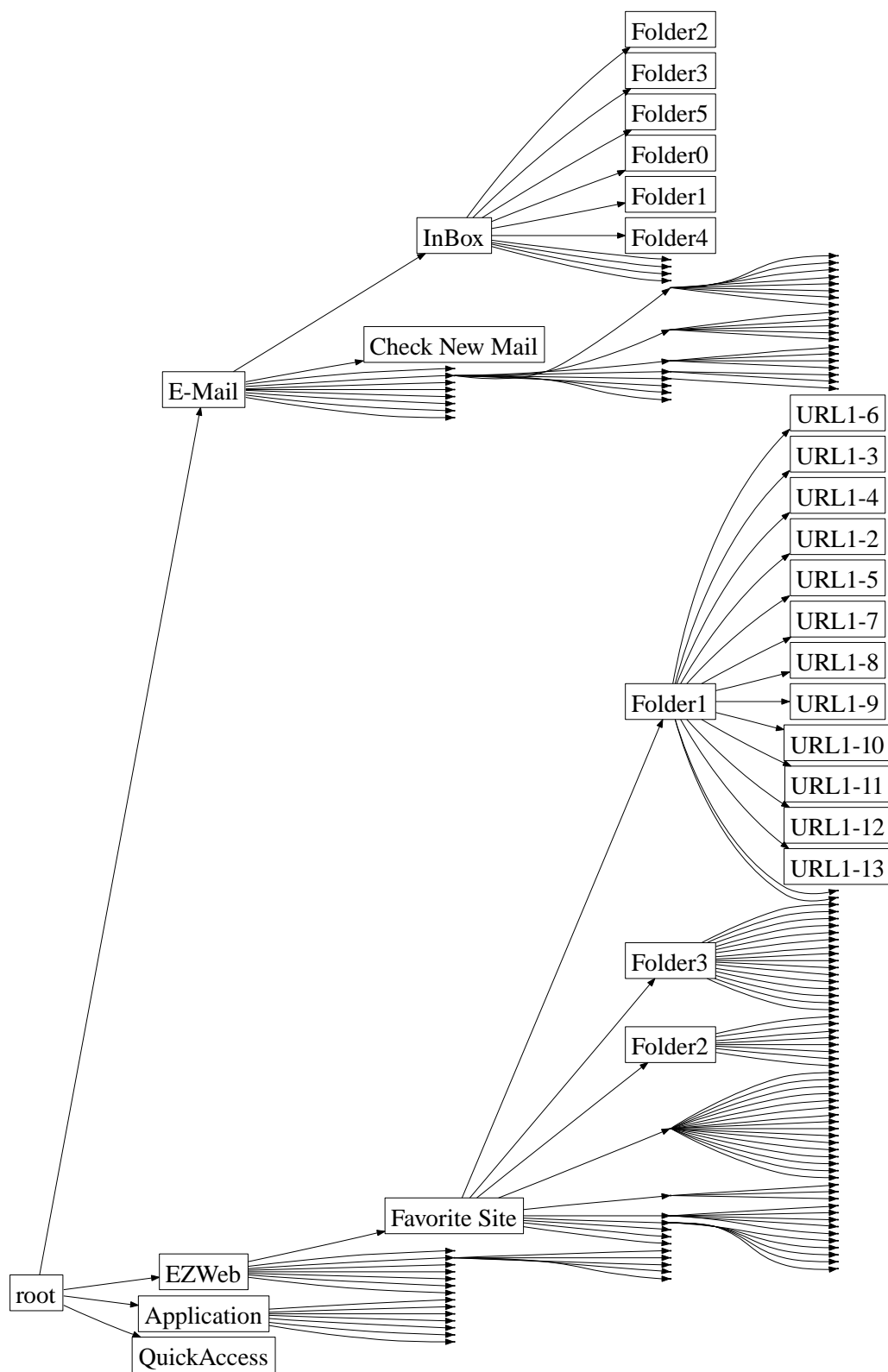
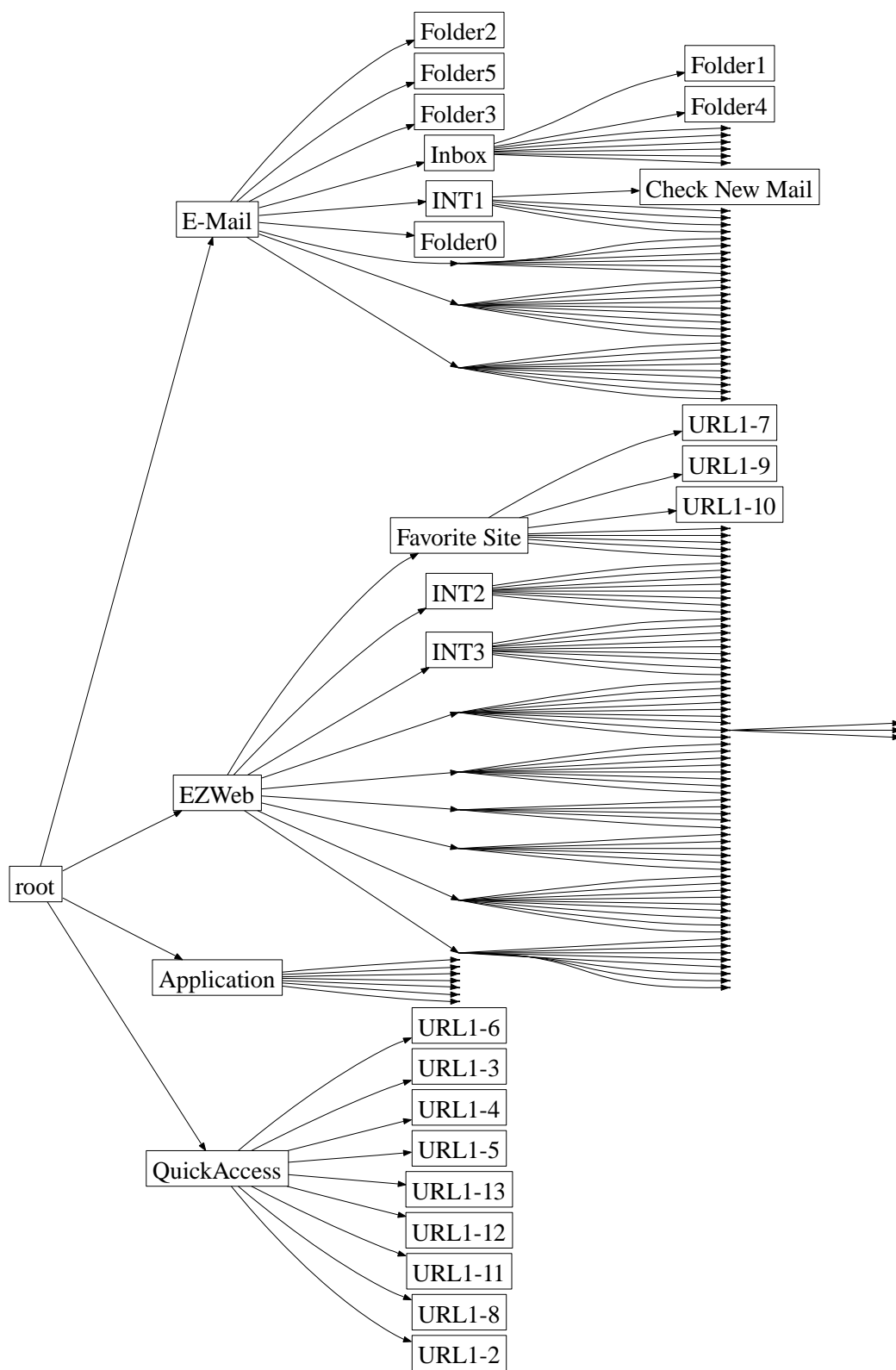
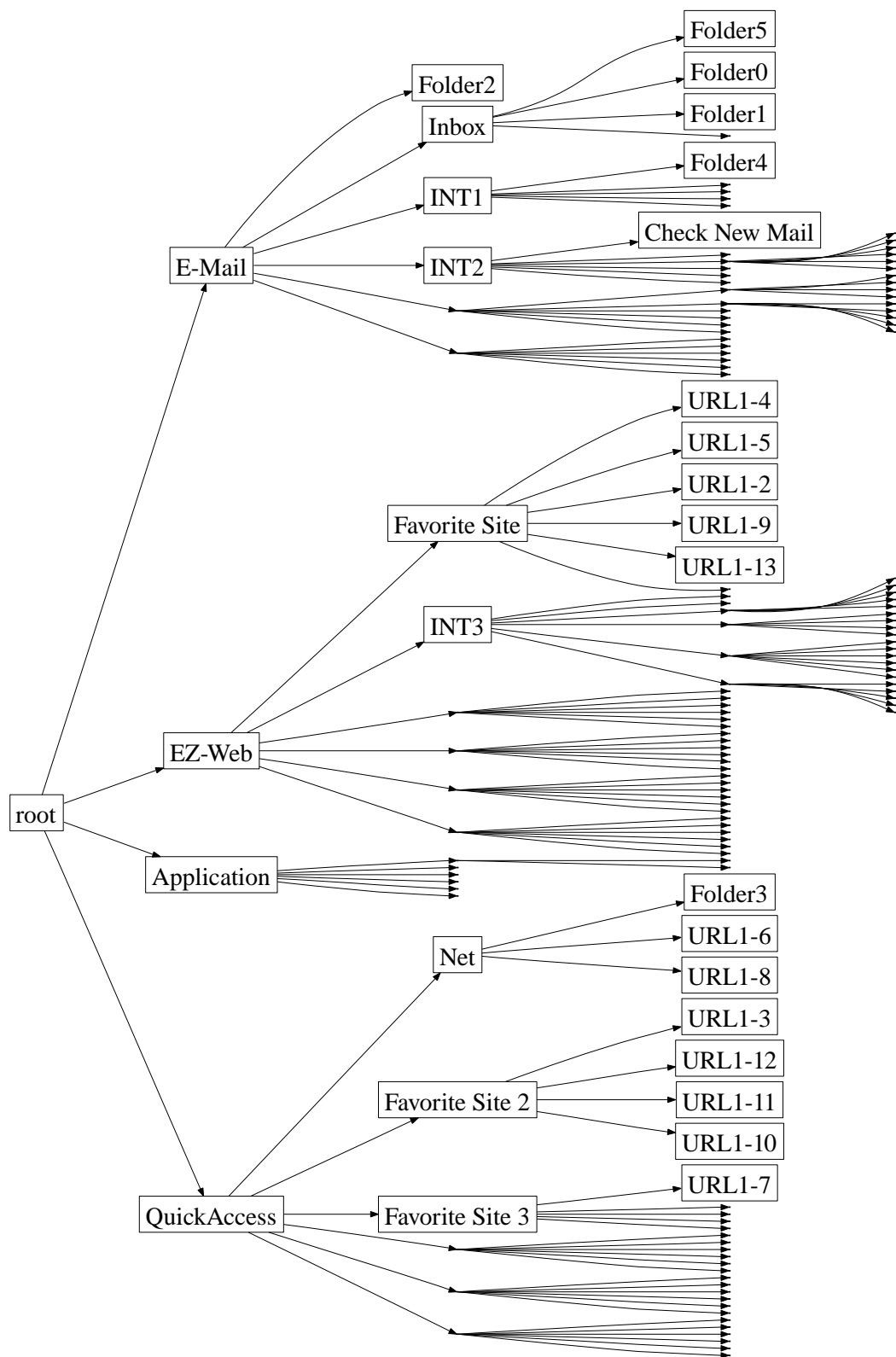
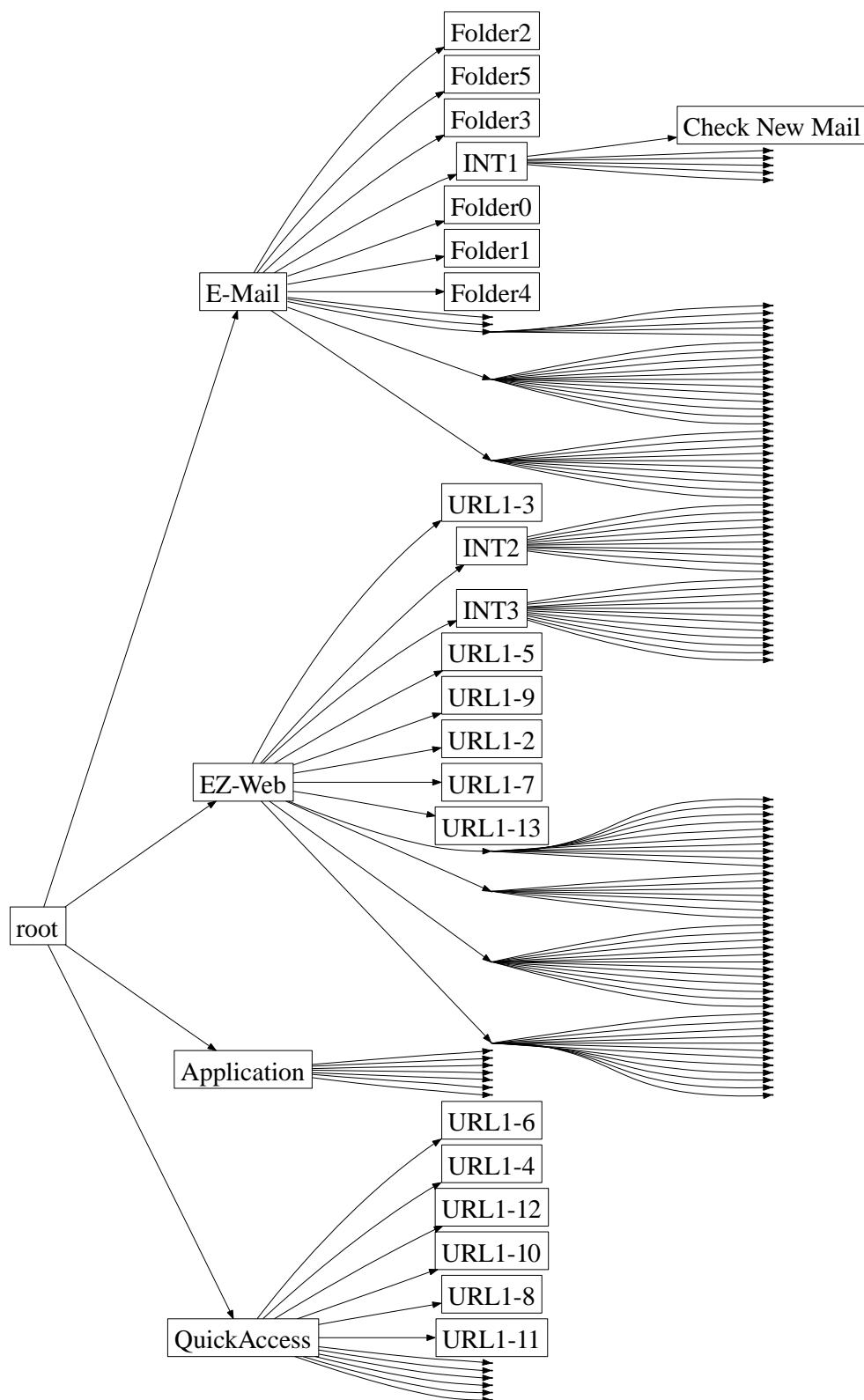
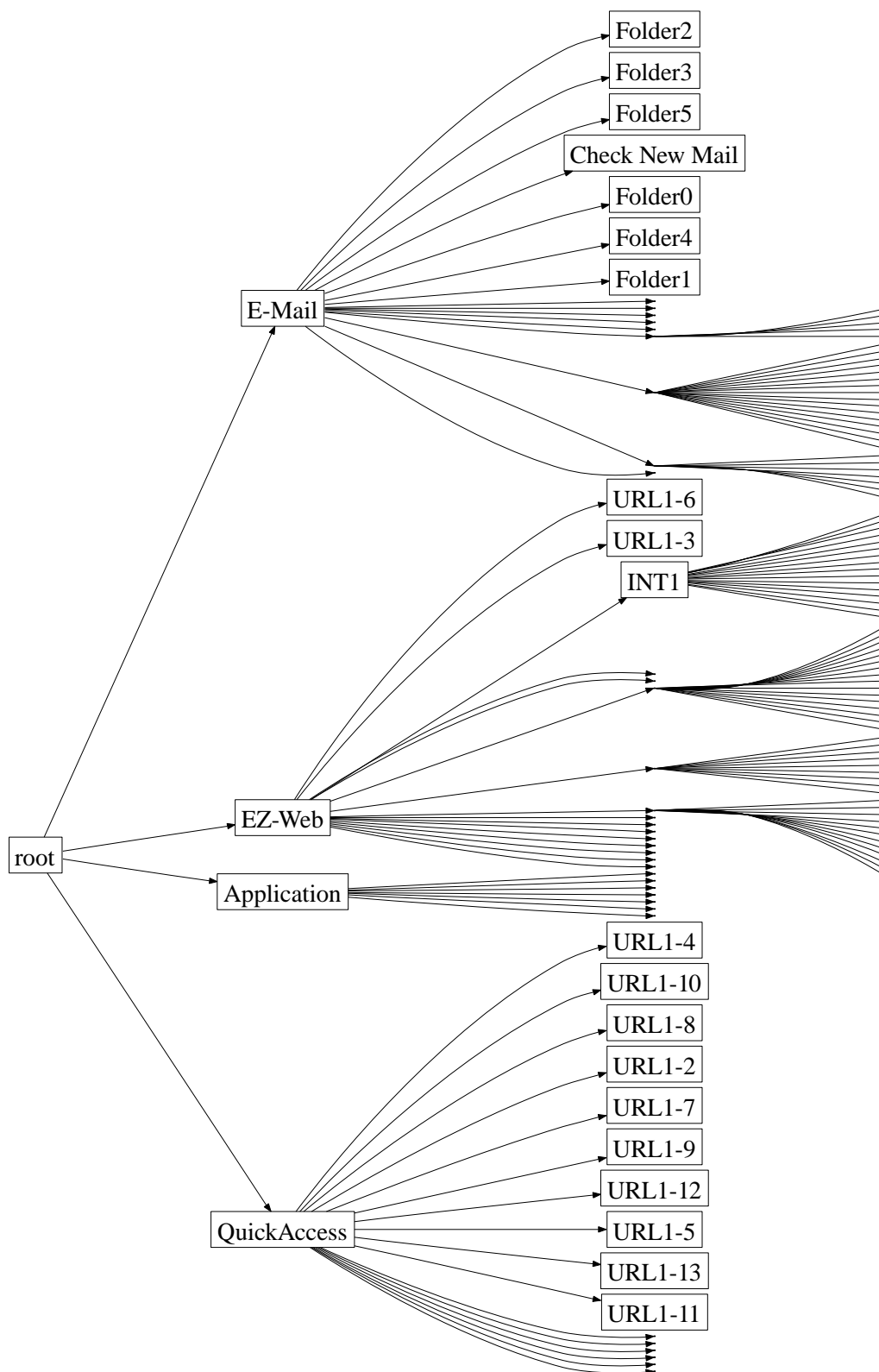


図 6-7 前方移動方式で生成されたメニューの例

図 6-8 生成されたメニューの例 ( $W = 9$ )

図 6-9 生成されたメニューの例 ( $W = 6$ )

図 6-10 生成されたメニューの例 ( $W = 12$ )

図 6-11 生成されたメニューの例 ( $W = 16$ )

## 6.3 ペナルティ項の影響

機能の類似度とメニューの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  $\alpha$ ,  $\beta$  の値を変えて計算した結果が表 6-3 と図 6-12 ~ 図 6-15 である。

機能の類似度のウェイト項の影響をみてみると、 $\alpha = 5$  と  $\alpha = 40$  では短縮率の平均はそれぞれ 47.6% と 42.4% であり、5.2 ポイント程度の違いが生ずる。平均到達時間は 2588 ms と 2848 ms であり、260 ms 程度の違いとなる。ベースケースの平均到達時間は 4941 ms であることを考えると、ウェイト項が平均到達時間に与える影響は小さいと考えてよい。

ウェイトを 0 とすることで平均到達時間を短くでき、短縮率も 50.8% となり、ベースケースの約半分の時間ですむことになる。しかし、ペナルティ項の値は大きくなり、得られたメニューをみると、異なるグループのメニュー項目が混在した形となっている。“QuickAccess” にはメール関連の項目と Web 関連の項目が混じり合って配置されている（図 6-16）。図では高頻度の項目のラベルのみを表示しているため、混在はあ

表 6-3 ペナルティ項の影響

$\alpha$	$\beta$	$T_{ave}$ (ms)		$Imp$ (%)		$P^s$		$P^g$	
0	0	2431	(50)	50.8	(1.0)	521	(11)	331	(81)
5	1	2588	(39)	47.6	(0.8)	362	(11)	182	(18)
10	1	2625	(46)	46.9	(0.9)	354	(6.1)	187	(23)
20	1	2678	(53)	45.9	(1.1)	351	(4.9)	191	(25)
40	1	2848	(149)	42.4	(3.0)	348	(5.7)	213	(40)
20	5	2697	(79)	45.4	(1.6)	353	(6.9)	171	(7.7)
20	10	2676	(37)	45.8	(0.7)	353	(6.0)	167	(5.0)

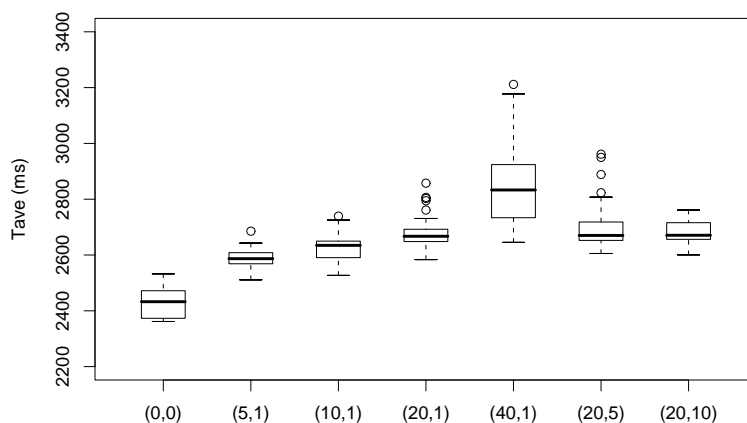


図 6-12 ( $\alpha$ ,  $\beta$ ) 毎の計算結果のばらつき (平均到達時間)

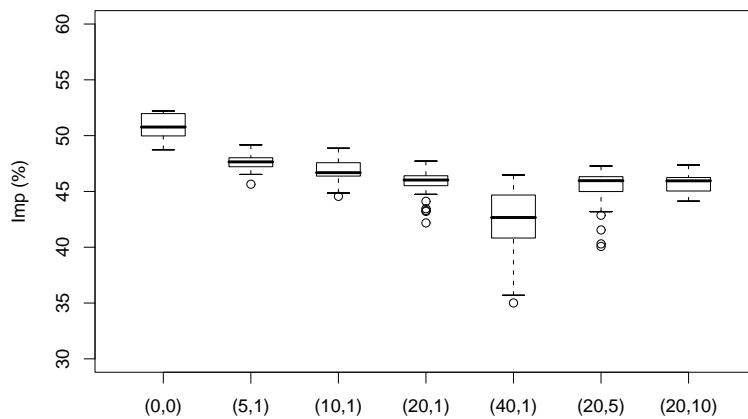


図 6-13  $(\alpha, \beta)$  毎の計算結果のばらつき (短縮率)

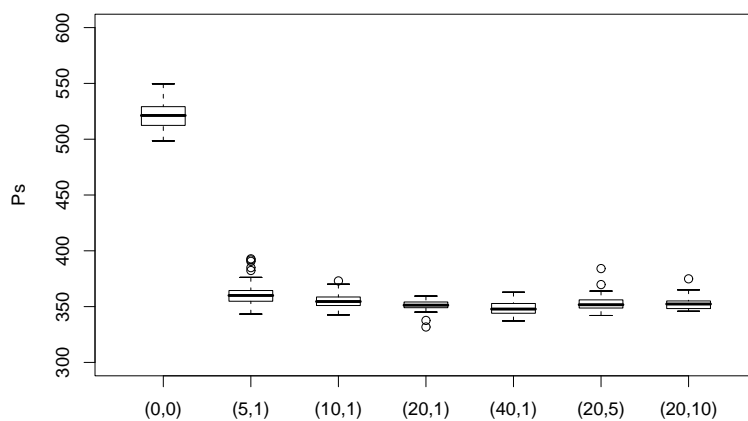


図 6-14  $(\alpha, \beta)$  毎の計算結果のばらつき (機能の類似度のペナルティ)

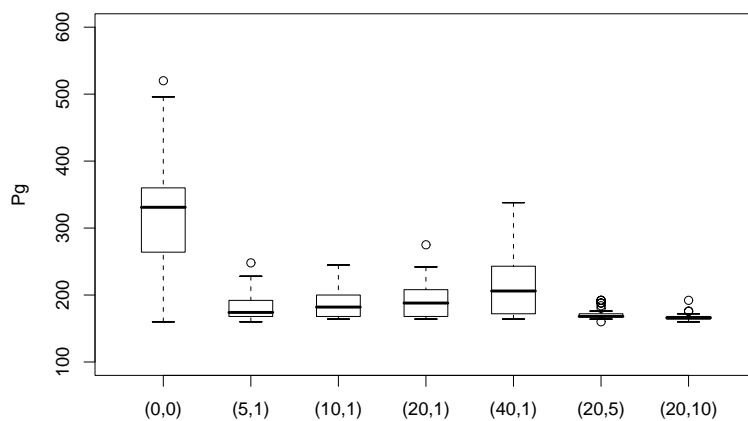


図 6-15  $(\alpha, \beta)$  毎の計算結果のばらつき (メニューの粒度のペナルティ)



まり目立たないが、生成されたメニューを見ると、記憶が困難なものとなっている。

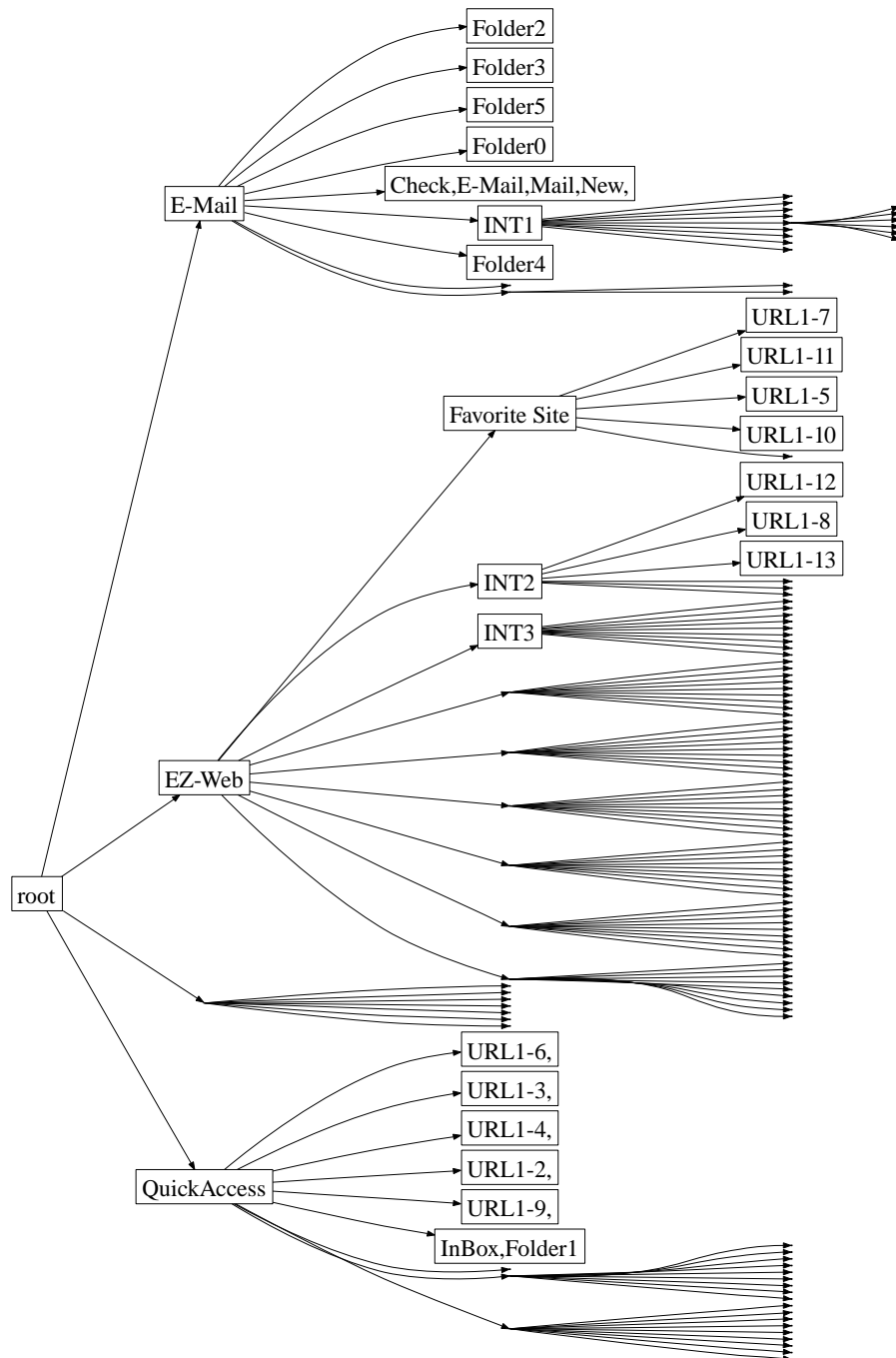


図 6-16 ペナルティ項の重みを 0 として生成したメニューの例 ( $W = 9$ )

表 6-4 推定式の係数の違いによる性能変化

係数				ベース	前方移動		提案手法			
$a^{sd}$	$b^{sd}$	$a^p$	$b^p$	$T_{ave}$	$T_{ave}(ms)$	$Imp(\%)$	$T_{ave}(ms)$	$Imp(\%)$	$P^s$	$P^g$
40	150	50	170	3137	2677	14.6	1691 (25)	46.1 (0.8)	352 (7.8)	180 (18)
40	150	80	220	3646	3052	16.2	1987 (28)	45.5 (0.8)	352 (6.3)	179 (18)
40	150	60	190	2421	2177	10.0	1343 (43)	44.5 (1.8)	352 (6.1)	178 (17)
90	300	60	190	5187	4674	9.9	2715 (36)	47.7 (0.7)	356 (5.3)	182 (18)
100	350	80	220	5936	5342	10.0	3105 (48)	47.7 (0.8)	357 (7.2)	189 (20)
100	400	100	300	6820	6009	11.9	3620 (64)	46.9 (0.9)	357 (7.1)	188 (19)

## 6.4 推定式の係数の影響

提案手法では実験データの回帰分析により求まる係数を4個使う．この係数の違いによる提案手法の性能をOriginalデータに対して $W = 9$ で確認した結果を表6-4に示す．表6-4で括弧内の数値は標準偏差を意味する．様々な係数の組み合わせに対して、ほぼ45%以上の短縮率が得られることが分かる．

このことは、推定式の係数が表6-4に示した範囲内であれば、提案手法により平均到達時間を大きく短縮できることを示している．またこの結果は、個人によるポインティング時間および探索・意思決定時間の相違に依らずに、提案手法が平均到達時間を大きく短縮できることをも示している．これは、PDAでの被験者実験の結果において、被験者毎の平均到達時間は異なるが、短縮率は似たような結果となっていることにも合致している．

## 6.5 様々な頻度データに対する性能

表6-5にメール利用の多いユーザを想定した頻度データ(Mail2, Mail3)とWeb利用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6-5では紙面の節約のために短縮率 $Imp(\%)$ の(%)は省略してある．また、50回の計算のばらつきの程度を図6-17～図6-20に示す．表6-5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により大幅に平均到達時間が短縮されている．

メールの頻度を高くしたMail2, Mail3とWebの頻度を高くしたWeb2, Web3の結果を比較すると、Webの頻度を高くした場合の方が、短縮率が大きい場合が多い．これは、元のメニュー構造では、Webのお気に入りや深い階層にあることに因っている．

表 6-5 平均到達時間の短縮効果 (Mail2, Mail3, Web2, Web3)

ケース	Mail2				Mail3				Web2				Web3			
	$T_{ave}$	$Imp$	$P^s$	$P^g$	$T_{ave}$	$Imp$	$P^s$	$P^g$	$T_{ave}$	$Imp$	$P^s$	$P^g$	$T_{ave}$	$Imp$	$P^s$	$P^g$
ベース	4701	—	399	793	4598	—	399	793	5251	—	399	793	5436	—	399	793
前方移動	4157	11.6	399	793	4043	12.1	399	793	4762	9.3	399	793	4966	8.6	399	793
$W = 16$	2826	39.9	681	1111	2836	38.3	699	1102	3201	39.0	683	1098	3238	40.4	701	1091
$W = 12$	2591	44.9	496	716	2557	44.4	502	697	2911	44.6	495	732	3015	44.5	494	719
$W = 9$	2501	46.8	360	191	2464	46.4	361	188	2754	47.6	359	181	2813	48.3	353	185
$W = 6$	2892	38.5	232	121	2866	37.7	234	116	3086	41.2	231	117	3117	42.7	230	115

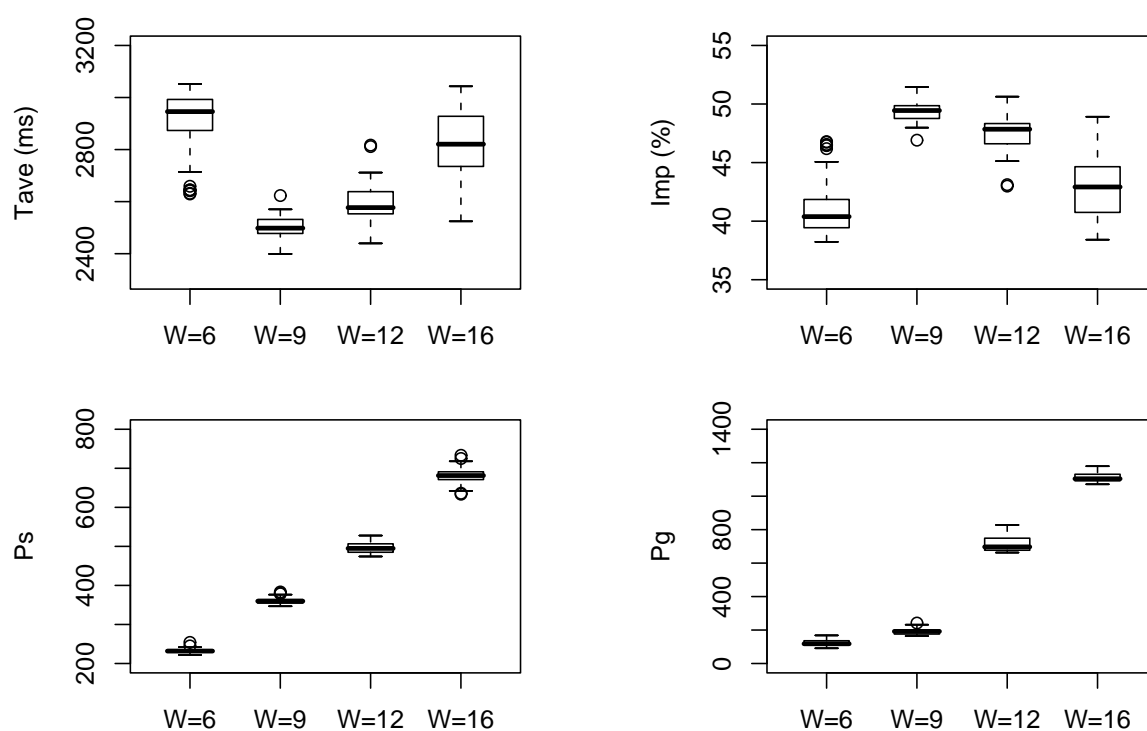


図 6-17 Mail2 データセットにおける計算結果のばらつき

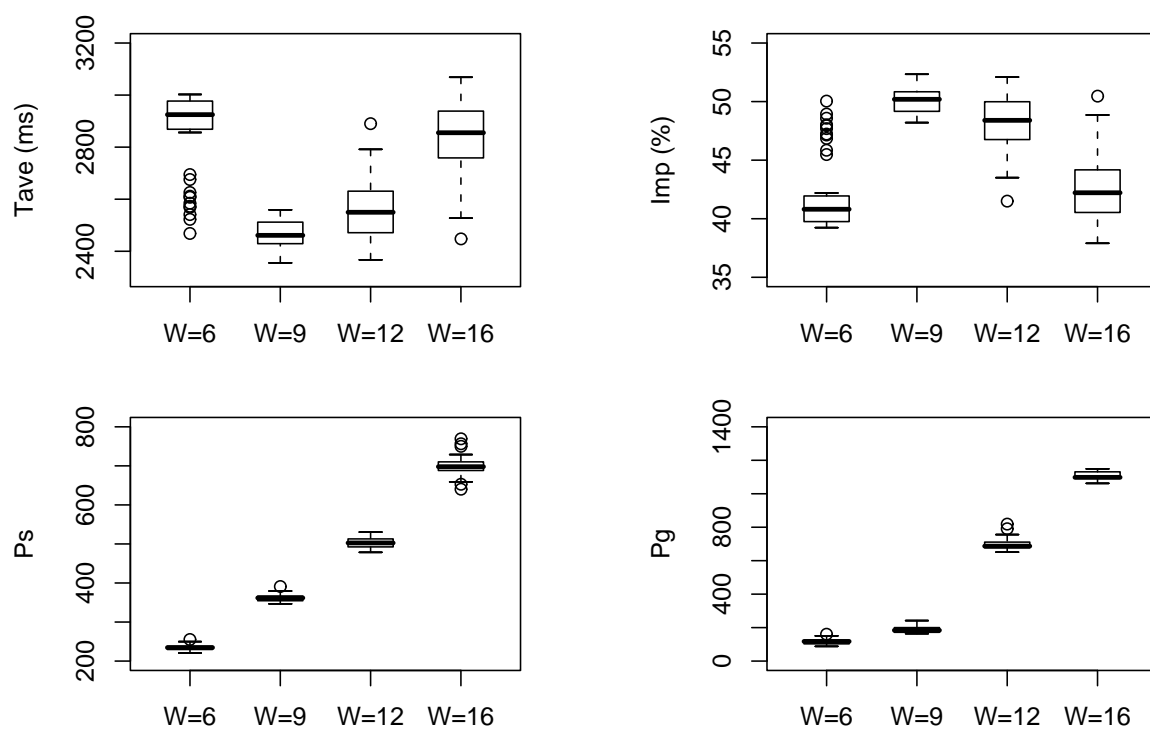


図 6-18 Mail3 データセットにおける計算結果のばらつき

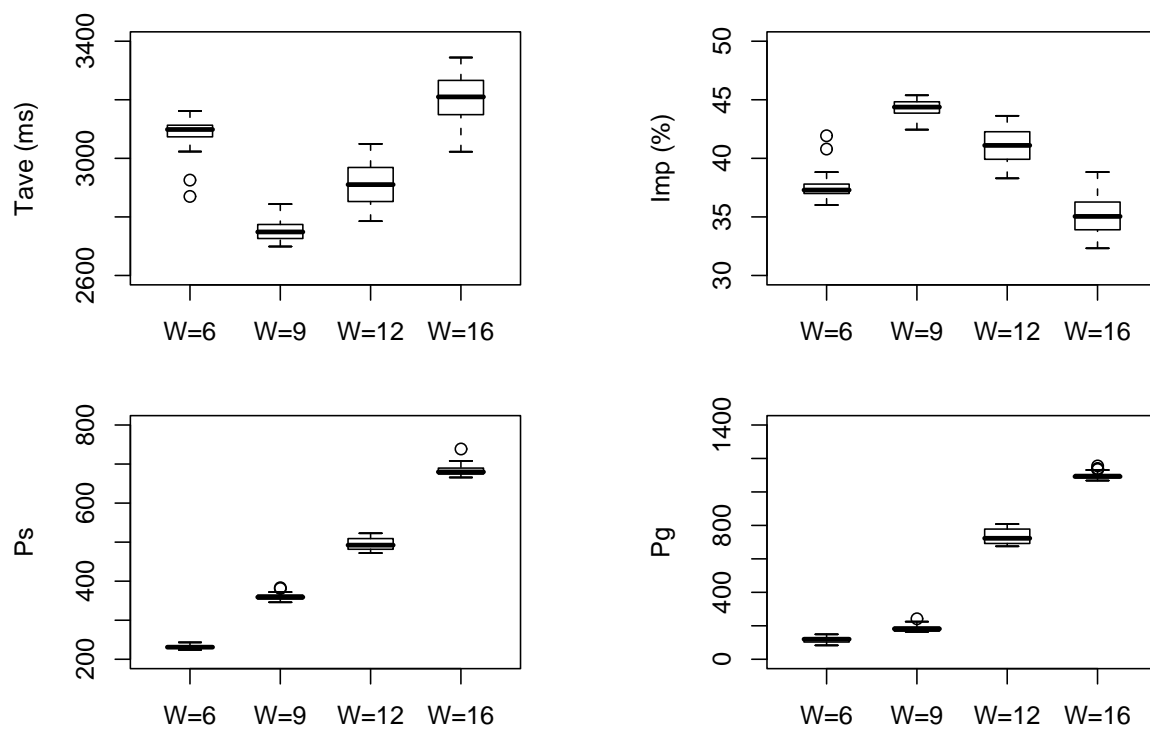


図 6-19 Web2 データセットにおける計算結果のばらつき

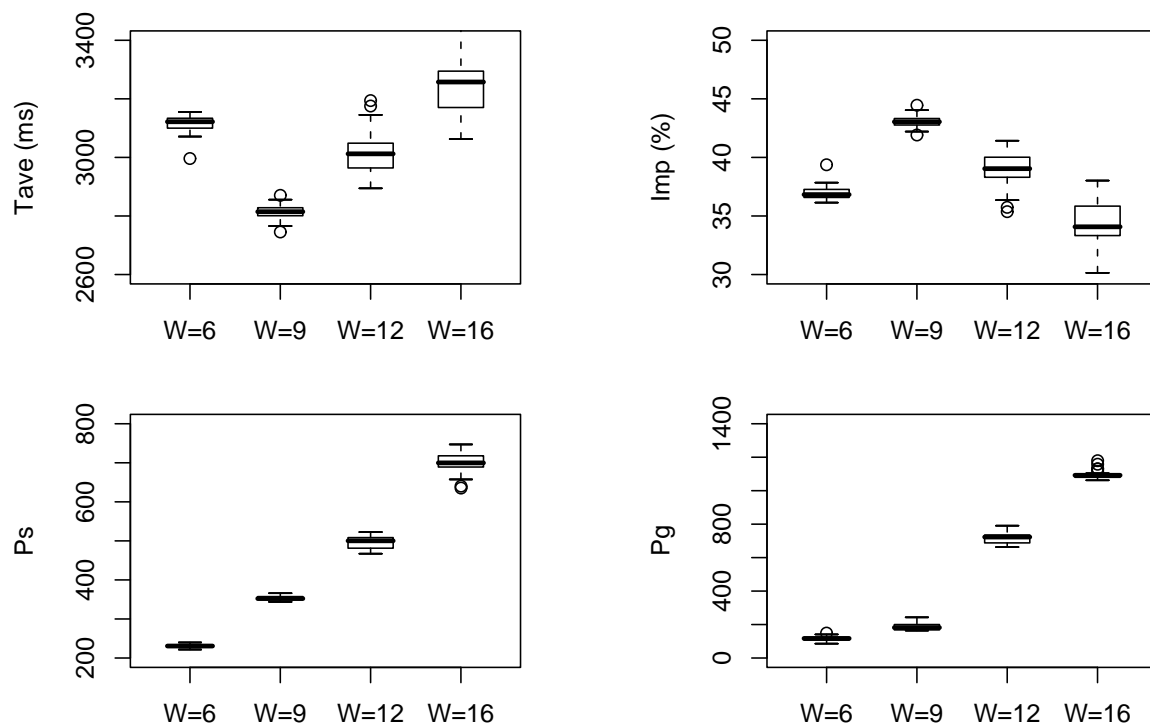


図 6-20 Web3 データセットにおける計算結果のばらつき

## 6.6 利用パターンに対する頑健性

前節までのシミュレーションで用いた利用パターンのデータは、本質的には1種類である。そこで、利用パターン、すなわち利用する項目数とそれらの利用頻度、が異なる場合に対する提案手法の頑健性を確認するための実験を行った。

Amant らが文献 [Amant 07] で述べているように、実ユーザの利用パターンを広範囲に収集することは非常に困難である。利用パターンを自動的に収集し、一定期間毎にセンターに送るような機能を持つ携帯電話などは存在しないことから、頻度分布の収集はアンケート調査などによるしかない。しかし、このようなアンケート調査の結果で、これまでに公開されているものはない。

Eagle らは、Bluetooth で通信可能な 100 台の同一機種スマートフォンにソフトウェアを追加して、9ヶ月に渡って様々なデータを収集し、ユーザの行動パターンを分析した結果を報告している [Eagle 06]。このような実験を可能とするためには膨大な資金が必要であることから、一般的に使える方式ではない。

そこで、ユーザ毎に大きく異なると考えられる利用パターンに対する提案手法の有効性を確認するために、Amant らが行ったと同様な、人工的に生成した複数の利用パターンを用いた実験を行った [松井 08c]。

### 6.6.1 利用パターンの生成

コンピュータのコマンドの利用頻度などは、 $N$  を項目数、 $k$  を順位、 $s$  を分布型のパラメータとして、

$$f(k; s, N) = \frac{1/k^s}{\sum_{n=1}^N 1/n^s} \quad (6.4)$$

の Zipf 分布で近似できる ([Kraut 83, Napier 92, Thomas 96, Linton 99] など)。

第 6.1.3 節で述べた方法で収集し、前節までの実験で用いた実測の利用頻度分布を  $s = 1$ ,  $s = 1.5$ ,  $s = 2$  の分布と比較したグラフが図 6-21 である。図 6-21 から、実測したデータは  $s = 1$  と  $s = 2$  の間にあり、Zipf 分布で十分近似できることが分かる。

前述のように、公開された利用頻度分布は存在せず、また利用するメニュー項目数と各項目の利用頻度は、利用者で大きく異なると考えられる。そこで、広範囲な利用パターンを想定して、前節までで用いたベースのメニュー構造のメニュー項目の利用頻度を以下の方法でランダムに変更したデータを生成した。

1.  $f(k; s, N)$  を用いて利用頻度分布を生成し、 $h = (h_1, h_2, \dots, h_{129})$  (ただし、 $h_i = 0 (i > N)$ ) とする。
2. 第 6.1.3 節のデータの利用頻度分布を  $o = (o_1, o_2, \dots, o_{129})$  としたとき、ランダムに選んだ  $o_i (1 \leq i \leq 129)$  を  $h_j (1 \leq j \leq 129)$  で置き換える。

具体的には,  $s = 1$ ,  $s = 1.5$ ,  $s = 2$  でメニュー項目数  $N$  が 32, 64, 129 の利用パターンを 20 パターンずつ作成した. 以下これらのデータを  $D(s, N)$  の形で表記する.

## 6.6.2 計算結果

推定式の係数, ペナルティ項の重みは 6.1.4 節と同じとした.  $W = 9$  として 50 回ずつ計算した結果を表 6-6 ~ 表 6-8 に示す. また, 前方移動と比較した場合の短縮率のばらつきを, 箱ヒゲ図の形で図 6-22 ~ 図 6-30 に示す. 表 6-6 ~ 表 6-8 では,  $MF$  は前方移動での短縮率(%)を,  $Imp$  は提案手法での短縮率(%)を, 括弧内の数値は標準偏差を表す. また, 図 6-22 ~ 図 6-30 では「pt.」はポイントを意味する. 表 6-6 ~ 表 6-8 と図 6-22 ~ 図 6-30 から以下が分かる.

- 頻度分布によって短縮率は異なるが, 提案手法は前方移動よりも大幅な短縮が可能である. 前方移動と提案手法の短縮率の差をみると, 小さい場合でも平均で 20 ポイント以上あり, 大きい場合には平均で 40 ポイント以上ある.
- 大きな短縮率が得られるのは, 深い階層にある項目の利用頻度が高い場合である. 前方移動では, 同一階層でポインティング時間の短い場所にしか移動できないが, 提案手法では, 浅い階層へと移動できることから, 短縮率が大きくなる.
- 提案手法の短縮率のばらつきは小さく, 標準偏差の最大は 3.5% 以下である.

以上から, 提案手法は様々な利用パターンに対して, 平均到達時間を大きく減少でき, また利用パターンの違いによる短縮率のばらつきは小さいことが分かる. これにより, 様々な利用パターンに対して提案手法は十分な頑健性を持つことが確認できた.

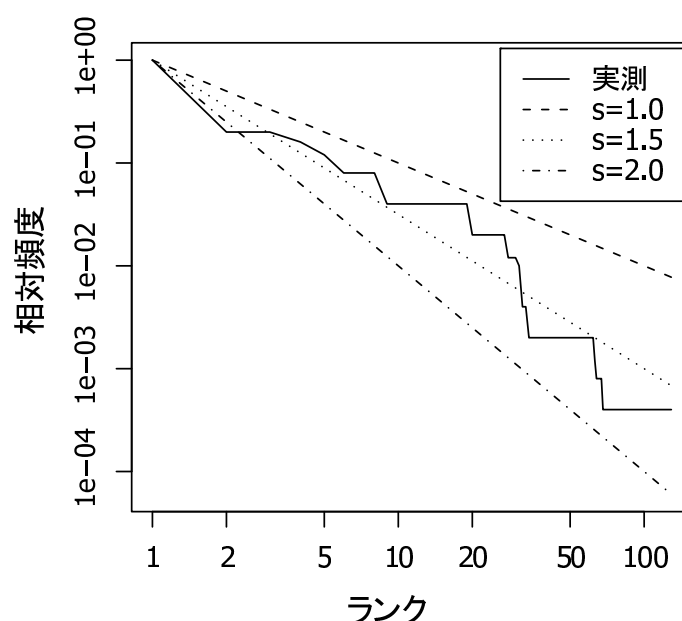


図 6-21 計測した頻度分布と Zipf 則での頻度分布の比較

表 6-6 Zipf 則に従った頻度分布での平均選択時間の短縮率 ( %) ( $D(1, *)$ )

ケース	$D(1, 32)$		$D(1, 64)$		$D(1, 129)$	
	<i>MF</i>	<i>Imp</i> (%)	<i>MF</i>	<i>Imp</i> (%)	<i>MF</i>	<i>Imp</i> (%)
1	24.8	65.8 (0.04)	19.8	48.1 (0.71)	17.9	45.8 (0.27)
2	17.2	55.7 (0.04)	22.3	48.4 (1.96)	19.0	48.7 (0.29)
3	21.2	61.4 (0.11)	17.4	54.3 (0.44)	16.6	41.8 (0.65)
4	25.0	64.2 (0.08)	19.8	46.7 (2.60)	15.3	38.6 (0.71)
5	22.7	55.0 (0.35)	27.4	56.8 (0.72)	18.0	41.5 (0.63)
6	17.3	57.3 (0.10)	21.0	49.2 (1.13)	23.4	47.2 (0.50)
7	26.2	62.9 (0.10)	18.5	53.9 (1.30)	15.6	37.3 (0.47)
8	27.6	61.9 (0.22)	15.3	44.5 (0.73)	17.2	43.7 (0.42)
9	21.4	61.5 (0.06)	22.2	56.2 (0.94)	14.0	36.1 (0.74)
10	21.1	65.6 (0.01)	21.6	48.5 (1.73)	20.4	46.7 (0.61)
11	28.2	66.1 (0.03)	28.5	60.5 (0.96)	23.3	46.9 (0.82)
12	18.2	57.4 (0.11)	23.6	54.9 (0.86)	14.5	39.1 (0.71)
13	23.9	63.7 (0.26)	18.2	48.8 (0.59)	17.1	40.9 (0.55)
14	23.0	65.6 (0.08)	19.3	49.6 (1.67)	19.6	43.3 (0.71)
15	15.6	52.5 (0.07)	24.6	50.9 (0.82)	23.1	47.4 (0.52)
16	22.1	62.1 (0.03)	23.4	56.9 (0.36)	22.4	49.8 (0.64)
17	24.9	65.3 (0.02)	20.3	55.5 (0.59)	21.6	48.8 (0.52)
18	18.7	59.6 (0.09)	22.6	54.6 (1.21)	19.6	46.5 (0.48)
19	19.2	47.9 (0.16)	24.7	58.3 (0.34)	15.9	42.5 (0.69)
20	23.9	63.6 (0.03)	18.7	54.5 (0.59)	19.7	49.5 (0.44)



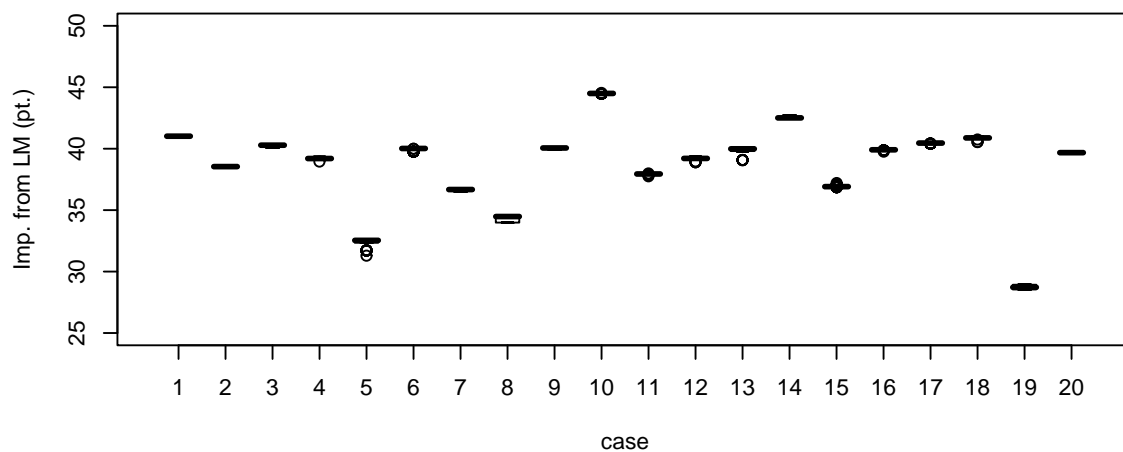
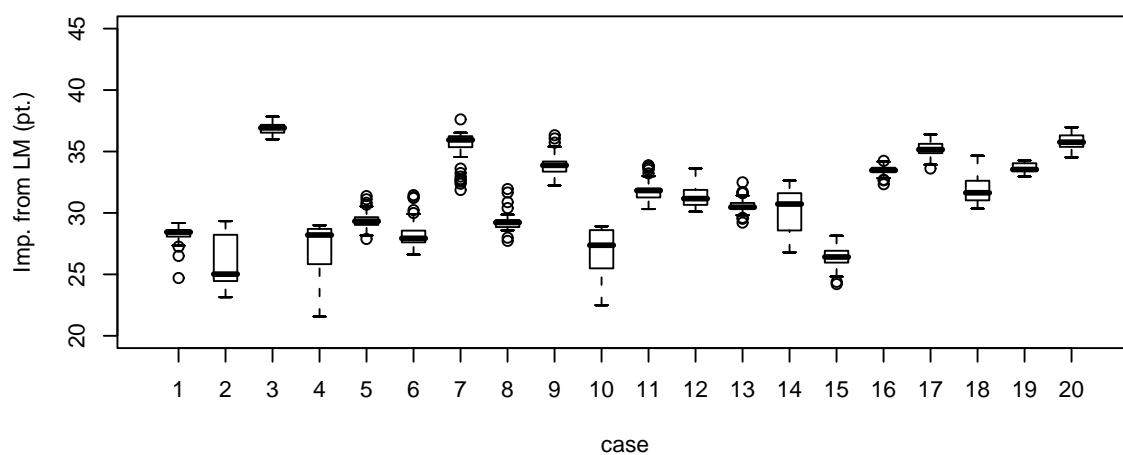
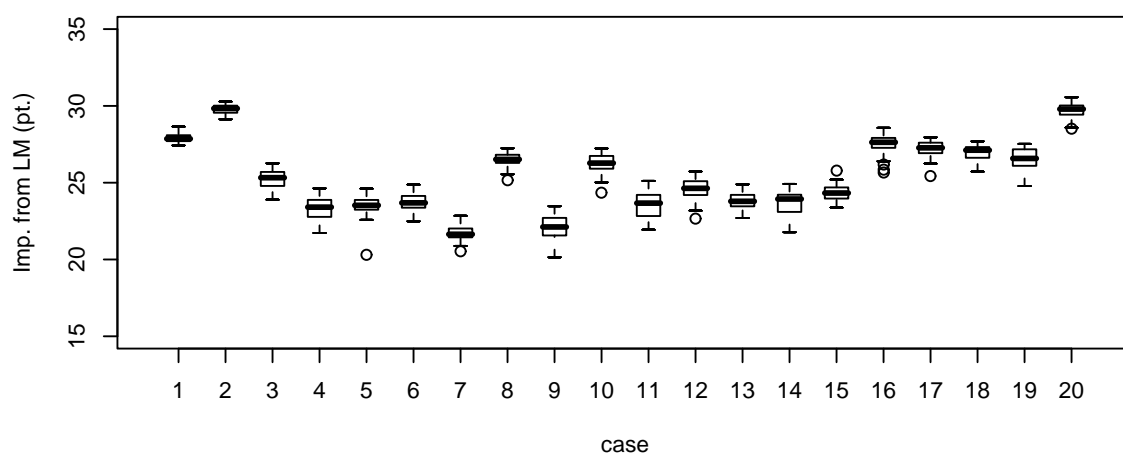
図 6-22 短縮率のばらつき ( $D(1, 32)$ )図 6-23 短縮率のばらつき ( $D(1, 64)$ )図 6-24 短縮率のばらつき ( $D(1, 129)$ )

表 6-7 Zipf 則に従った頻度分布での平均選択時間の短縮率 ( % ) ( $D(1.5, *)$ )

ケース	$D(1.5, 32)$		$D(1.5, 64)$		$D(1.5, 129)$	
	<i>MF</i>	<i>Imp (%)</i>	<i>MF</i>	<i>Imp (%)</i>	<i>MF</i>	<i>Imp (%)</i>
1	20.8	53.3 (0.13)	32.3	63.3 (0.59)	11.6	35.0 (1.39)
2	30.4	68.4 (0.38)	19.9	65.0 (0.64)	15.7	55.5 (0.39)
3	25.7	67.8 (0.12)	11.4	35.6 (1.00)	17.6	40.7 (1.16)
4	14.7	63.6 (0.03)	24.8	61.3 (1.67)	27.3	62.1 (0.31)
5	33.4	71.8 (0.05)	33.1	60.8 (2.67)	29.0	57.8 (0.60)
6	29.0	61.4 (0.23)	22.9	57.6 (0.76)	27.5	60.3 (0.33)
7	8.2	42.1 (0.10)	26.0	59.4 (1.29)	19.9	58.1 (0.21)
8	28.4	69.1 (0.39)	19.6	48.1 (1.16)	27.0	60.1 (0.35)
9	29.2	69.1 (1.12)	19.8	42.0 (1.77)	19.5	51.5 (0.52)
10	26.5	67.8 (0.12)	30.1	64.9 (1.46)	19.3	58.4 (0.26)
11	25.6	67.0 (0.02)	23.1	51.4 (1.81)	14.1	51.3 (0.34)
12	21.3	66.0 (0.04)	31.4	65.3 (1.68)	24.6	57.2 (0.64)
13	29.6	66.7 (0.07)	24.1	48.7 (0.88)	12.1	51.4 (0.41)
14	25.6	69.1 (0.03)	25.7	49.1 (1.40)	26.2	56.5 (0.65)
15	32.4	70.7 (0.04)	14.2	46.6 (2.24)	30.4	62.6 (0.39)
16	19.7	63.0 (0.12)	19.0	60.4 (0.87)	17.3	54.5 (0.40)
17	23.4	65.7 (0.19)	25.1	57.7 (1.67)	17.2	53.1 (0.38)
18	20.5	59.1 (0.01)	24.8	65.9 (0.60)	27.5	59.8 (0.49)
19	17.4	61.6 (0.07)	21.5	51.6 (1.18)	16.9	47.4 (0.48)
20	28.6	69.8 (0.06)	29.6	62.5 (1.80)	27.8	60.4 (0.43)

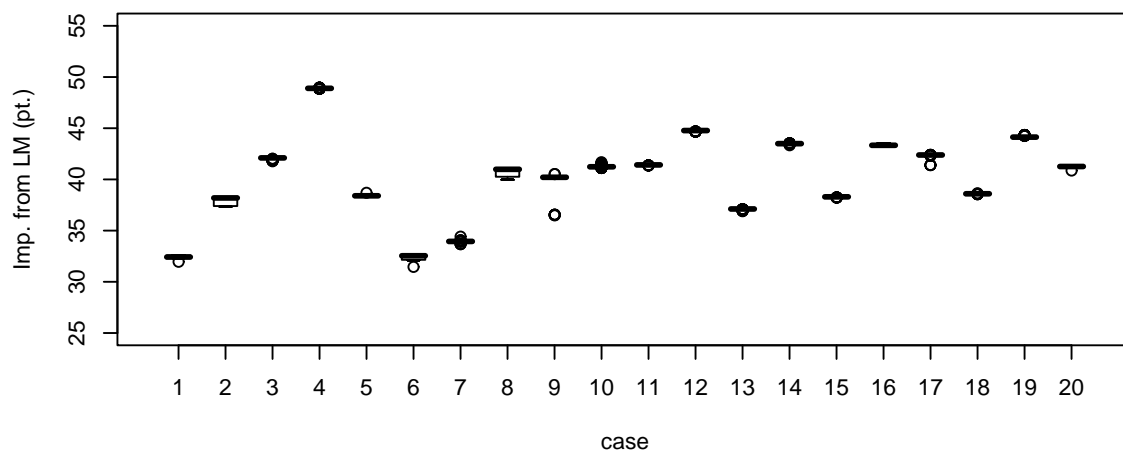
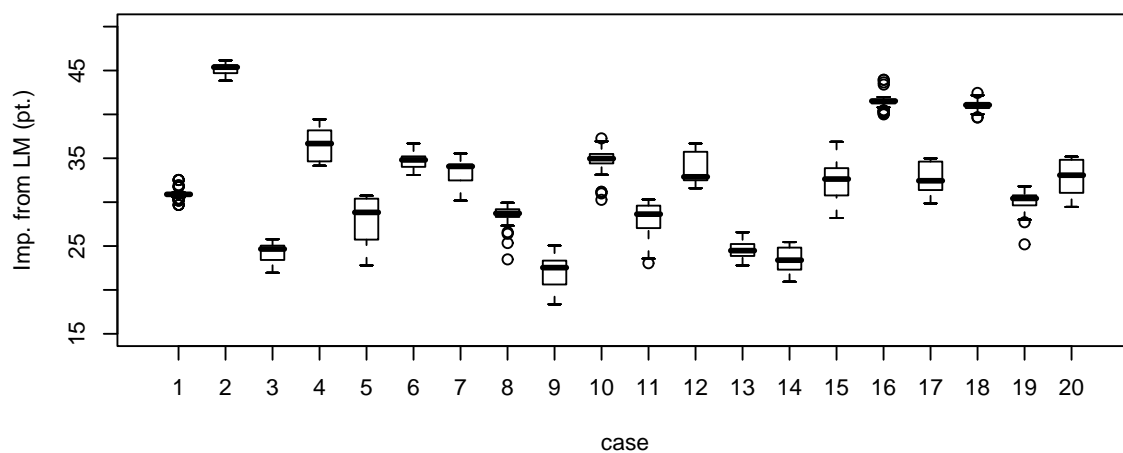
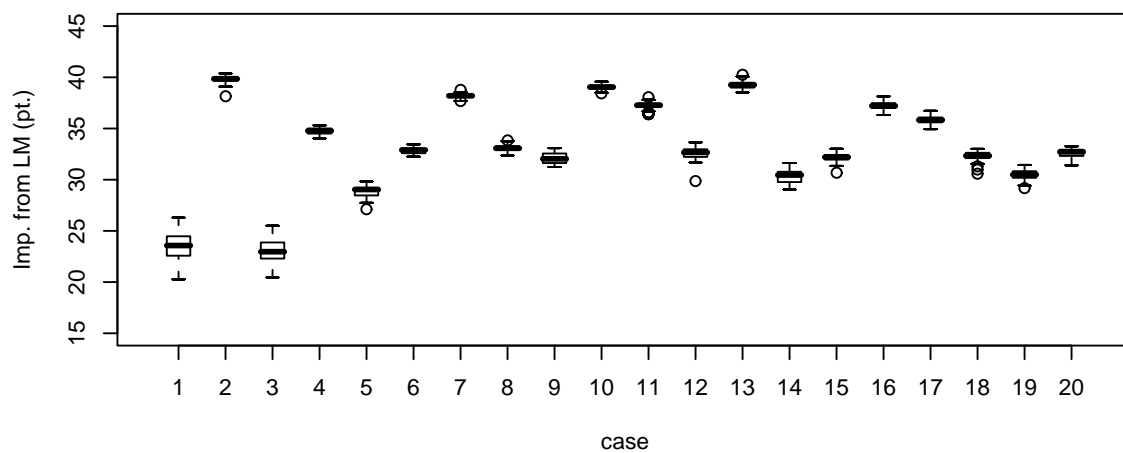
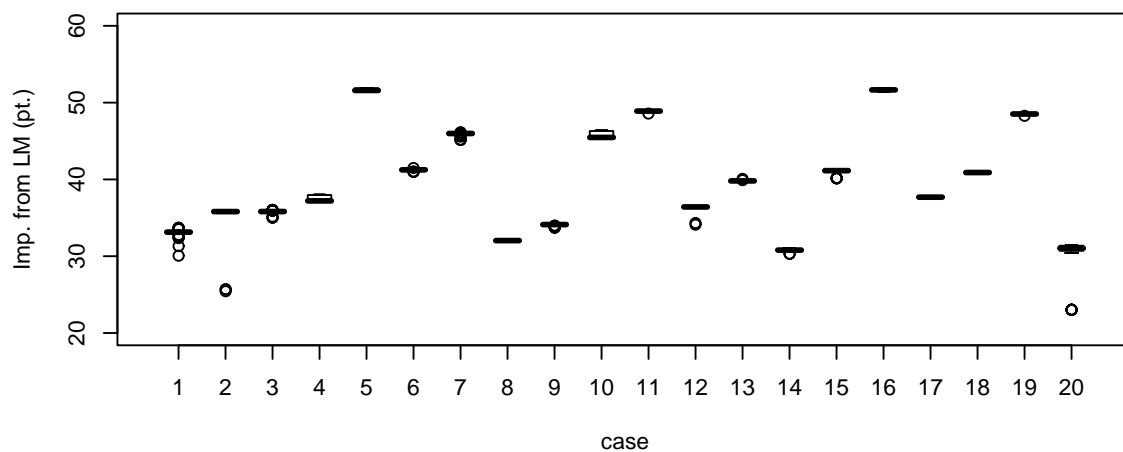
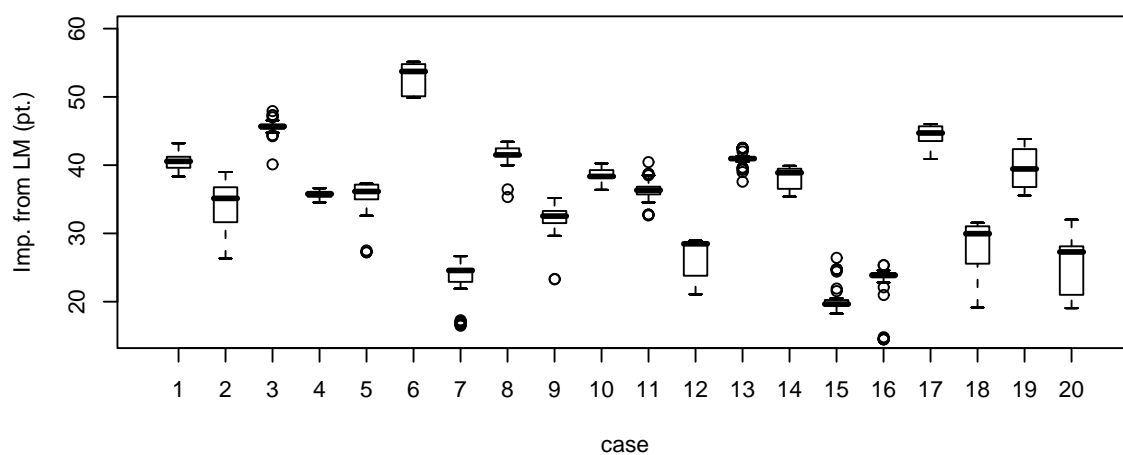
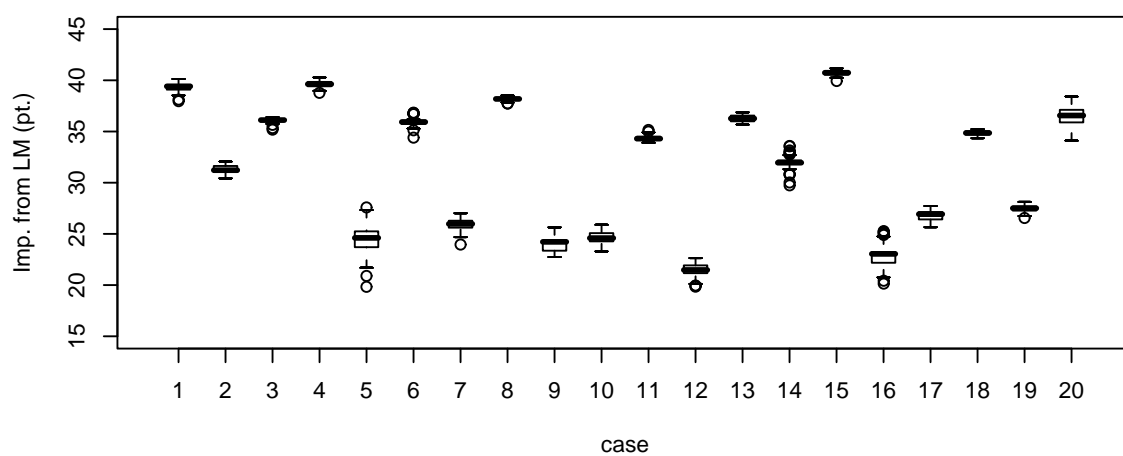
図 6-25 短縮率のばらつき ( $D(1.5, 32)$ )図 6-26 短縮率のばらつき ( $D(1.5, 64)$ )図 6-27 短縮率のばらつき ( $D(1.5, 129)$ )

表 6-8 Zipf 則に従った頻度分布での平均選択時間の短縮率 ( % ) ( $D(2, *)$ )

ケース	$D(2, 32)$		$D(2, 64)$		$D(2, 129)$	
	<i>MF</i>	<i>Imp (%)</i>	<i>MF</i>	<i>Imp (%)</i>	<i>MF</i>	<i>Imp (%)</i>
1	12.9	46.0 (0.58)	18.4	58.9 (1.05)	23.4	62.7 (0.41)
2	16.5	51.7 (2.46)	13.3	47.0 (3.49)	24.7	56.0 (0.37)
3	8.7	44.5 (0.22)	20.4	66.0 (1.07)	25.3	61.4 (0.24)
4	12.3	49.8 (0.37)	30.6	66.3 (0.54)	22.0	61.7 (0.33)
5	14.8	66.4 (0.15)	10.7	46.1 (2.67)	12.0	36.5 (1.52)
6	32.2	73.4 (0.08)	13.3	66.1 (2.16)	33.4	69.3 (0.46)
7	23.5	69.4 (0.24)	31.8	54.9 (3.26)	18.7	44.6 (0.59)
8	37.2	69.2 (0.04)	24.7	66.3 (1.43)	28.8	67.0 (0.17)
9	38.0	72.0 (0.10)	13.7	45.8 (2.34)	27.7	51.7 (0.69)
10	24.7	70.5 (0.42)	31.7	70.2 (0.92)	24.7	49.3 (0.57)
11	24.2	73.1 (0.05)	6.1	42.3 (1.51)	25.0	59.4 (0.29)
12	13.7	50.1 (0.45)	28.8	55.7 (2.83)	10.7	32.2 (0.69)
13	31.6	71.4 (0.06)	19.6	60.5 (0.93)	24.6	60.9 (0.32)
14	8.0	38.8 (0.13)	32.1	70.5 (1.41)	33.3	65.2 (0.73)
15	34.3	75.3 (0.30)	27.7	48.1 (1.96)	23.6	64.3 (0.24)
16	16.8	68.4 (0.08)	29.1	51.9 (3.14)	10.3	33.3 (1.18)
17	15.5	53.1 (0.07)	25.5	69.8 (1.68)	20.0	46.8 (0.50)
18	31.0	71.9 (0.03)	15.6	43.5 (3.87)	27.3	62.1 (0.21)
19	23.7	72.2 (0.04)	20.5	60.0 (2.91)	28.3	55.8 (0.33)
20	32.8	63.2 (2.22)	25.2	50.8 (4.03)	25.8	62.3 (0.93)

図 6-28 短縮率のばらつき ( $D(2, 32)$ )図 6-29 短縮率のばらつき ( $D(2, 64)$ )図 6-30 短縮率のばらつき ( $D(2, 129)$ )

## 6.7 計算時間，個体評価回数

計算時間はメニューの項目数にほぼ比例する部分と  $W$  の2乗にほぼ比例する部分からなる．C++で作成したプログラムをGCCでO3オプションでコンパイルし，Intel Xeon E5420 (2.5 GHz)，RAM 4GB で実行した場合に，1回の計算に要したCPU時間はメニューの項目数が129個の場合には， $W = 6$  では約30秒， $W = 9$  では約34秒， $W = 12$  では約40秒， $W = 16$  では48秒であった． $W = 9$  でメニュー項目数が32個の場合には12秒，64個の場合には17秒であった．

最良解が求まるまでに要した個体の評価回数は，メニューの項目数， $W$  に依らずにほぼ一定であった．メニューの項目数が129個の場合には， $W = 9$  の場合で平均82,770回，標準偏差11,914回であった． $W$  が異なる場合でも，同様な評価回数で最良解が求まった．ただし，Zipf分布に基づいて生成した利用パターンの場合には，平均が90,000回に近づく場合もあった．

## 6.8 まとめ

提案する手法の有効性を検証するために，多様な条件を想定してのコンピュータシミュレーションを行った結果について述べた．

本章では，実験に用いたデータについて述べた後で，携帯電話の端末の階層メニューを対象に，複数の利用パターンを用いて，提案する解法の性能をシミュレーションにより確認した結果を述べた．また，多様な利用パターンを想定して，Zipf分布を用いて生成した利用パターンにより，提案手法の頑健性を検証した結果についても述べた．加えて，計算時間について述べた．

実験で用いた範囲のデータに対しては，データに依らずに45%以上の時間短縮効果が得られることが分かった．また，多様な利用パターン（機能数と頻度分布）を対象としたシミュレーション実験から，多様な利用パターンに対しても提案手法は有効であることを確認した．このことから，提案手法は，多様なユーザに対して十分な性能を持つと考えられることを示した．

## 第7章 考察

本章では、提案手法の有効性と適用可能範囲について述べた後、適応干渉との関係、提案手法では明示的に考慮していないユーザのエラーモデルについて議論する。また、浅いメニューと深いメニューの関連について述べる。次に、GA に基づく提案手法の性能を確認するために行った、シミュレーテッドアニーリング (SA: Simulated Annealing) に基づく解法、交叉方式を変えた GA に基づく解法と比較した結果を述べ、解法の性能改善の方向性について述べる。

### 7.1 提案手法の有効性と限界

#### 7.1.1 提案手法の優位性

提案手法による平均到達時間の短縮率は、メニュー項目の利用頻度、ポインティング時間、探索・意思決定時間などユーザに起因するデータに依って変化する。これに対する頑健性の観点からは、実験で用いた範囲のデータに対しては、データに依らずに 45% 以上の時間短縮効果が得られることを示した。また、多様な利用パターン（利用する項目数と利用頻度）を対象としたシミュレーション実験により、多様な利用パターンに対しても提案手法は有効であることを確認した。このことから、提案手法は多様なユーザに起因するデータ、利用パターンに対して十分な性能を持つと考えられる。

Amant らの手法 [Amant 07] では、利用頻度の高い項目が上位階層に移動されるだけであり、“Shortcut” の一種である “QuickAccess” への移動はできない。これに対して提案手法では、“QuickAccess” への移動も可能である。携帯電話では “QuickAccess” は有効であることが報告されており [Bridle 06]、ここに配置する項目を自動的に決定できる提案手法の有効性は高いと考えられる。

#### 7.1.2 提案手法が有効でない場合

利用頻度に偏りが少ない場合で、“QuickAccess” に配置されない項目もある程度の頻度で選択される場合には、Howes らが報告しているように、実ユーザでは全体としての時間が増加する可能性もある [Howes 00]。

また、提案手法による改善効果があまり大きくない場合として、以下に示すような状況が考えられる。

- 元々のメニュー構造において、利用頻度の高いメニュー項目がレベルの浅い部分にだけ配置されている。
- 携帯電話を想定した場合に、利用しているメールのフォルダ数が多くなく、フォルダ利用頻度の偏りが少ない。

これらの状況は、既にある程度の最適化がなされている状況ともいえる。こういった状況がどの程度多いかなど、利用者の利用状況データを収集し、提案手法の実用的な改善効果を明らかにするのは今後の課題である。

## 7.2 提案手法の適用範囲

### 7.2.1 静的メニューへの適用

本論文で対象とした階層メニューは「固定的」で「ポップアップ」型である。Windows などでは、カスケード型のメニュー構造が一般的であるが、これに対しても提案手法は容易に適用可能と考えられる。Web ベースの階層型情報提示システムも「固定的」で「ポップアップ」型と考えることができるものが多いことから、このようなシステムに対しても提案手法は適用可能と考えられる。

Windows の「スタートメニュー」は、ユーザの指定した順番に項目が並ぶことから提案手法の適用は容易である。

これに対して、ソフトウェアが自動的に項目の並び替えを行う場合には、若干の工夫が必要になる。例えば、第2章に示した Thunderbird でのメッセージの移動の例（図 2-3）には、以下の形で適応可能と考えられる。

1. メッセージのフォルダは名称でソートされることから、メッセージの移動先の利用頻度に応じて、頻度の高いフォルダが上位に来るように、フォルダ名の先頭に“0”-“Z”を付けて頻度順に並べる。
2. 利用頻度の多いフォルダは「履歴」の下に上記と同じ方式で並べる。
3. 実際のフォルダ名を変更したくない場合のために、上記の方式は、表示項目のみに対して行う。

### 7.2.2 適応メニューへの適用

Cockburn らは、適応メニューとして、利用頻度に応じて項目の表示される領域が広くなる“Morphing”メニューを提案している [Cockburn 07]。これは図 7-1 の (a),(b) に示すように、利用頻度の高い項目の文字の大きさと高さを増加させ、逆に利用頻度の



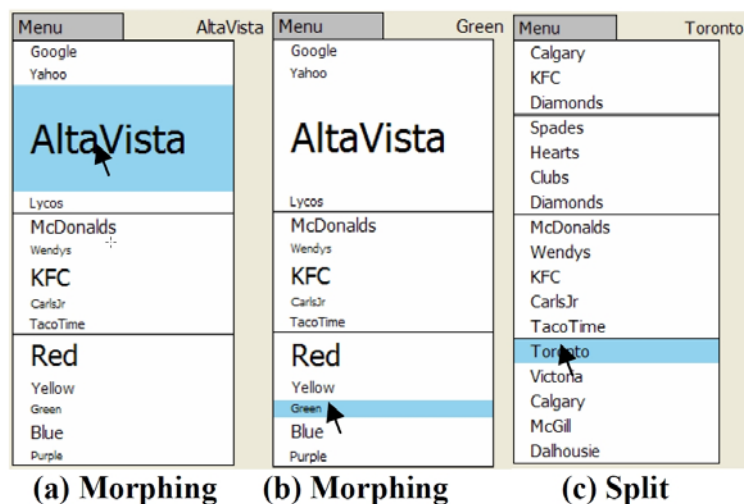


図 7-1 Morphing メニューの例 [Cockburn 07]

低いものの文字の大きさと高さを減少させる方式である．これにより，ポインティング時間が短くなる．この方式は動的であり，項目の順番は並べ替えない．

利用頻度の高い項目が占める領域を広くするという考え方は，提案手法でも利用可能である．具体的には，項目の領域は一定と仮定して計算しているポインティング時間を，項目毎に異なるものとして計算する方式に変更することで対応可能である．

### 7.2.3 適用のためには拡張が必要な事例

カスケード型では，次の階層が現れる位置が固定的でない場合もある．例えば，第 2.1 節の図 2-3 に示したメッセージの移動の例では，コンテキストメニューが現れる位置により，下位メニューが右側に出現するか，左側に出現するかが異なる．また，出現する位置によっては，サブメニュー中の項目の総てを表示できずに，スクロールが必要な場合もある．この場合には，定式化においてスクロールに必要な時間をポインティング時間に付加する必要がある．

提案手法では，メニューの項目数  $W$  全体を表示することを前提としている．このため，意思決定・選択時間は，初心者では  $W$  に比例する時間，熟練者では  $\log_2 W$  に比例する時間が必要となる．同時に表示される項目の利用頻度が偏っている場合には，低頻度のものはスクロールしないと表示されないようにする戦略も考えられる．これにより，高利用頻度の項目の意思決定・選択時間の短縮が可能となる．この場合には，メニュー項目の位置に依らずに決まる，意思決定・選択時間を，上位  $n$  項目と残りの  $W - n$  項目では異なるものとして定式化する必要がある．さらに，ポインティング時間にスクロールに要する時間を考慮した定式化が必要となる．

上記のような場合には，定式化の変更が必要であるが，解法の変更は必要ないと考えられる．詳細な検討は今後の課題である．

### 7.2.4 操作の系列を考慮した最適化

本論文では，図 7-2 に青色で示したパスを辿ってメニュー項目  $v_0^D$  が選択されるものと仮定している．すなわち，初期状態（根）から始まるパスを辿ってメニュー項目が選択されるものと仮定している．このことは操作の系列を考慮していないことに相当する．一般に，Windows のアプリケーションで使われているカスケード型のメニューでは，メニュー項目が選択されると表示されていたメニューは消えて，初期状態に戻る．このため，上記の仮定が成り立つ．これに対して，携帯電話，あるいは Web ベースのシステムでは，直前のメニュー階層に戻ることが可能な場合が多い．この場合には，図 7-2 に緑色で示したパスを辿って  $v_1^D$  を選択することが可能である．

初期状態に戻ることなく項目  $I_i$  の次に項目  $I_j$  を選ぶことが可能な場合には，操作の系列を考慮した最適化が有効と考えられる．操作の系列としては，項目数が 2 以上のものもあるが，簡単化のために以下では 2 組の操作を対象とする．操作の系列を考慮した最適化を行うためには，以下の対応が必要となる．

1. 各メニュー項目の頻度ではなく，操作系列のデータを収集し，緑色で示したパスで使われる項目を特定する．対象とするアプリケーション，機器によっては，データ収集が困難なことが多い．この場合には，利用頻度の高い組み合わせを明示的に指定する．
2. 遺伝的アルゴリズムを用いた解法で使っている，順列から木への変換に，2 組を 1 回の操作で割り当てする特別な番号を導入する．
3. 特定された組み合わせについては，同一メニューに配置されるという制約を追加し， $I_i \rightarrow I_j$  の操作については， $I_j$  の到達時間を根からのパスの時間ではなく， $I_i \rightarrow I_j$  の時間として求める．

基本的には上記で対応可能と考えられるが，詳細な検討は今後の課題である．

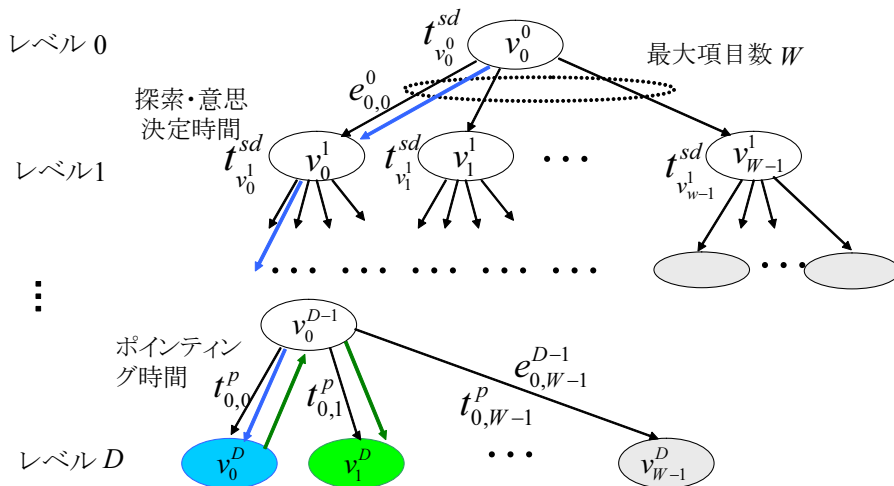


図 7-2 操作の系列の模式図

## 7.3 適応干渉

### 7.3.1 提案手法における適応干渉

提案手法により生成されたメニューは、新しいメニューであり、それに適応して慣れる必要がある。この場合には認知負荷が増加する。本論文で提案する手法は、メニュー項目の利用頻度が大きく変化した場合などに、ユーザの判断で最適化することを目的としており、短い時間間隔で適応的にメニュー構造を最適化するためのものではない。このため、最適化されたメニューにユーザが適応して慣れるための認知的負荷の増加は頻繁に起こるものではない。しかしながら、メニュー構造を最適化した直後には、そのような認知的負荷は高まる。

適応インタフェースの良否については長く議論されており、適応干渉 [Meltzoff 95, 山田 07] の問題も指摘されている。適応干渉 (adaptation interference) とは、「適応対象自体が相手の適応により操作される状態」[山田 07] である。提案手法が対象とした階層メニューとユーザで考えれば、ユーザが適応しようとしている階層メニューがユーザのメニュー項目の利用頻度の変化に伴って変化する状態である。

前述のように、提案手法は短い間隔での変化は想定していないため、適応干渉は頻繁に生ずるものではない。また、第2章「関連研究」でも述べたように、Gajos らの最近の研究によれば、ユーザの予測に合致した適応であれば、適応インタフェースの方が良いとされている [Gajos 08a]。提案手法は、頻度の高い項目が浅い階層に、同じ階層内では、頻度の高い項目が前の方に配置されることから、ユーザの予測も容易である。このことから、変化したメニューの認知的負荷も低減できると考えられる。

### 7.3.2 提案手法の拡張の方向性

利用頻度の高い項目数が多くない場合には、十分に慣れたメニューに対しては、意思決定・選択時間のモデルは熟練者のモデルが妥当と考えられる。提案手法により新たなメニューを生成してそれに慣れるまでには、初心者モデルが妥当と考えられる。この場合には、慣れるまでは、提案手法で生成したメニューでの平均到達時間の方が長くなる可能性もある。

Cockburn らは「初心者」から「熟練者」への遷移をモデル化する方式を提案し、実験によりその妥当性を示している [Cockburn 07]。「初心者」から「熟練者」への遷移は、 $e_i$  を項目  $i$  の習熟度を表す変数 ( $0 \leq e_i \leq 1$ )、初心者の意思決定・選択時間は  $t_i^n$  で、熟練者のそれは  $t_i^e$  で表すとして、項目  $i$  の意思決定・選択時間  $t_i^{sd}$  を次式でモデル化している。

$$t_i^{sd} = (1 - e_i)t_i^n + e_i t_i^e \quad (7.1)$$

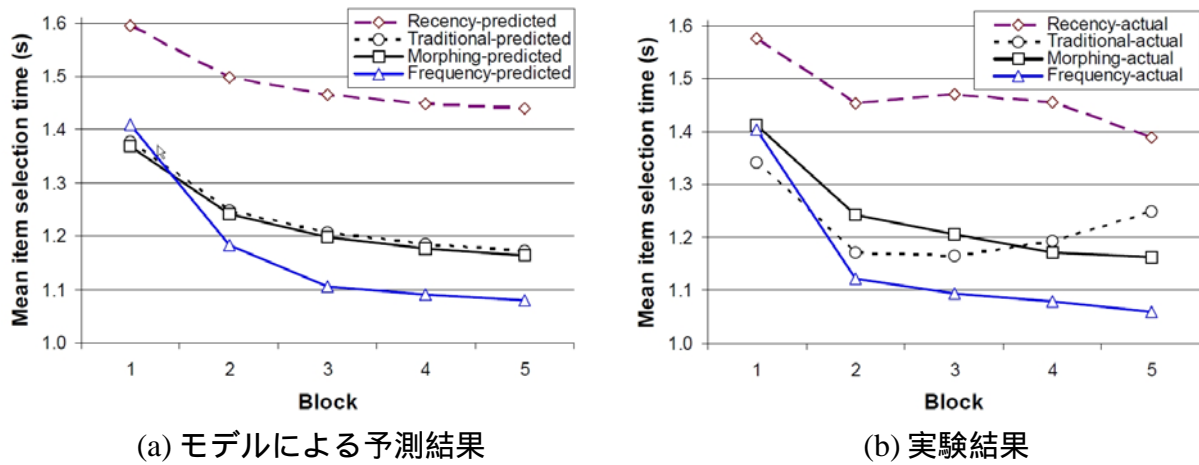


図 7-3 習熟の例 [Cockburn 07]

また,  $L$  をインタフェースの「習熟の容易性」を表すパラメータ ( $0 \leq L \leq 1$ ),  $t_i$  を項目  $i$  がこれまでに選択された回数として, 習熟度は以下の形でモデル化することを提案している.

$$e_i = L(1 - 1/t_i) \quad (7.2)$$

メニュー項目の位置が固定的な場合には,  $L = 1$  とし, 項目の位置が動的に変化する場合には, 項目の平均移動数とメニューの項目数の半分の比率を 1 から引いたものとして推定することを提案している. 項目の移動がランダムな場合には, 項目数を  $l$  とすると  $L = 1 - 0.5l/0.5l = 0$  となる.

このモデルを用いて, 4 種類のメニュー方式について実験した結果が図 7-3 である. ここで, “Traditional” は従来型の固定的なメニューである. メニュー項目を選択し易い位置に移動する, 基準として, “Recency” は直近の利用を基準, “Frequency” は利用頻度を基準としたものである. “Morphing” は図 7-1 に示した, 利用頻度に応じて, 項目の高さを変化させる方式である. 従来型での予測結果と実験結果の合致度が低いこと, 実験結果では従来型は習熟効果が見られないことを除けば, 予測と実験は良く合致している. また, 数回の反復で習熟することも分かる.

この結果は, K.L. Norman の書籍 [Norman 91] に示された学習による到達時間の短縮結果によれば, 数回の練習で時間は 3/5 程度に短縮できることが多いとされていることとも合致している.

上述の結果からは, 与えられたメニューに習熟するために必要な時間は長くはないと想定される. しかし, 生成されるメニューと現状のメニューとの違いの大きさにより, 慣れるために要する時間が異なるかどうかは未解明である.

習熟に要する時間も目的関数に採り入れた最適化手法の検討は, 今後の課題の一つである. このためには, メニュー間の距離として適切なものが必要であり, またメニュー間の距離と習熟の関連についての被験者実験も必要となる.

## 7.4 ユーザのエラーモデル

本論文では、ユーザが理想的な行動をした場合の平均到達時間を考えた。すなわち、ユーザは目的とする項目を誤りなく選択することを前提とした。つまり、ユーザは目的の項目のあるサブメニューまで誤りなく辿り着くものとし、サブメニュー内の項目を誤りなく選択するものと仮定した。階層メニューで発生するエラーには、辿るべき階層を間違えるエラーとポインティングエラーが考えられる。以下では、この二つのエラーについて議論する。

### 7.4.1 辿るべき階層の誤り

利用頻度の高いメニュー項目は浅い階層に置かれることから、利用頻度の高いメニュー項目に対しては、辿るべき階層を間違えた場合の回復コストは低いと考えられる。しかし、利用頻度の低い項目に対しては、回復コストが増加する可能性がある。

辿るべき階層が誤っていたことにより生ずる回復コストをモデル化するためには、以下の項目に関する検討と適切なモデル化が必要である。

- エラーに気づく時点のモデル化、すなわち、終端ノードが含まれるサブメニューまで達して気づくのか、あるいは途中のサブメニューで気づくのか。
- エラーからの回復の方法、すなわち、上位階層まで順次戻るのか、あるいは根ノードまで直接戻るのか。

### 7.4.2 辿るべき階層の誤りのモデル化の方向性

定式化で示したように、終端ノードを選択するには、根からレベル  $l$  まで辿り着く必要があり、終端ノード  $v_k^T$  への到達時間  $T_k$  は根から目的ノードまでのパス上の  $t_{v_i^l}$  の総和となる。このパスを  $R = (v_0^0, \dots, v_{i_k}^k, k \leq D)$  としたとき、到達時間  $T_k$  は以下で表現できる。

$$T_k = \sum_{v \in R} t_{v_i^l} \quad (7.3)$$

平均到達時間  $T_{avg}$  は、終端ノード  $v_k^T$  の選択確率を  $H_k$  として、以下で表現できる。

$$T_{avg} = \sum_{k=1}^N H_k T_k \quad (7.4)$$

エラーに気づくのは終端ノードが含まれる階層に到達した時点であり、エラーに気づいた場合には根ノードまで直接戻ると仮定した場合の、最も単純なモデル化は以下のようなになる。

終端ノード  $v_k^T$  に誤りなく辿り着く確率を  $h_k$  とし、エラーに気づくまでのパスを  $R_e = (v_0^0, \dots, v_{i_j}^m, m \leq D)$  とし、根ノードに戻るのに要する時間は  $T_k^r$  であるとする。この場合には、エラーに気づくまでの時間  $T_k^e$  は以下となる。

$$T_k^e = \sum_{v \in R_e} t_{v_i^l}. \quad (7.5)$$

このとき、平均到達時間  $T_{avg}$  は以下で表現できる。

$$T_{avg} = \sum_{k=1}^N H_k \{h_k T_k + (1 - h_k)(T_k^e + T_k^r)\}. \quad (7.6)$$

デバイス毎に  $T_k^r$  は異なると考えられるが、メニューの構成とデバイスの機能から容易に推定可能である。しかしながら、 $h_k$  の推定は困難である。最も単純には、利用頻度の高いノードのエラー率は低く、利用頻度の低いノードのエラー率は高いと考えられるが、実態については被験者実験などによる確認が必要である。

上記とは異なる仮定の下でのモデル化も同様な考え方で可能であるが、エラーに気づく階層、戻り方などに関するパラメータが増加することになる。

上記のモデル化と異なるアプローチとしては、以下が考えられる。北島らは、潜在意味解析 (LSA) と Markov 連鎖モデルを組み合わせる階層メニューの探索過程をモデル化し、メニュー構成を評価する方式を提案している [北島 02]。この提案の中で、エラーからの回復についていくつかのユーザモデルを提案している。例えば、過去の履歴に関係なく上位層に戻るモデル、2 階層以上戻るモデル、エラーであったノードは再度訪問しないモデルなどを提案している。ユーザのモデル毎に、成功率、平均クリック数などをシミュレーションで比較検討している。

生成されたメニューについて、この方式を用いて評価することは可能である。しかしながら、どのユーザモデルが現実的であるかについては明らかではないという問題点が残っている。

以上の課題の解決は今後の課題の一つである。

### 7.4.3 ポインティングエラー

ポインティングエラーは、最適化の有無に因らず発生するものである。最近提案されたポインティングエラーを予測するモデル [Wobbrock 08] によれば、ターゲットまでの距離とターゲットの幅、さらにポインティング時間によりエラー率が異なる。この特質も考慮したユーザのエラーモデルも組み込んだ最適設計も今後の課題の一つである。

## 7.5 浅いメニューあるいは深いメニュー

2.4.1 節「浅いメニューあるいは深いメニュー」で述べたように、表示画面が広い場合には、同時に表示する項目数が多く、階層数が少ない浅いメニューの方が、項目数が少なく、階層数が多い深いメニューよりも良いことが知られている。ただし、Gevenらが携帯電話を対象として行った実験結果からは、項目数が多すぎると良くないことが分かっている。

シミュレーション実験結果からは、 $W = 9$  として生成したメニューの方が、 $W = 6, 12, 16$  として生成したメニューよりも時間短縮率が大きいことが分かっており、これは上述の先行研究の結果と合致するものとなっている。

提案手法では、浅いメニューが生成される割り当て方式を用いて、同時に表示する最大項目数を  $W$  で制御する方式を採用しているため、 $W$  を変えてメニューを生成することで、平均到達時間の小さなメニューを生成することができる。ただし、対象とするデバイスによっては、 $W$  のとれる範囲には制限がある。例えば、携帯電話では、メニュー項目名の前に電話機のボタンに相当する記号 (“1”–“9”, “\*”, “0”, “#”) を付加して、その記号で選択可能としている機種が多い。この場合には、 $W \leq 12$  でないと記号で選択できない項目が生ずることになる。また、ボタンの配置からは  $W$  としては3の倍数が自然と考えられる。したがって、検討すべき  $W$  の数は少ないため、計算量もさほど多くない。

これに対して、コンピュータディスプレイの場合には、 $W$  の範囲の制約は少ないため、検討すべき  $W$  の数が多くなり、計算量も比例して多くなる懸念がある。最近のPCは高速であり、またマルチコアのCPUの価格も急激に低下していることから、多数の計算を行うことはさほどの負荷ではないことから、この懸念は大きな問題にはならないと考えられる。

できるだけ計算量を減らしたい場合には、 $W$  のとれる値を予め適切な範囲として定め、遺伝的アルゴリズムにより  $W$  も同時に決定する解法を使うことが考えられる。この解法では、染色体上では  $W$  の部分と割当順序の部分は異なった意味を持つことから、交叉方式を工夫する必要がある。

## 7.6 SAに基づく解法との性能比較

提案する手法の性能を確認するために，GA で用いる交叉の方法を変えた解法，シミュレーテッドアニーリング (Simulated Annealing; SA) に基づく解法との比較を行った [Matsui 08c] ．

### 7.6.1 SA に基づく解法

SA に基づく解法の概要を図 7-4 に示す．この解法は，Michalewicz と Fogel による書籍 [Michalewicz 00] に示されているものである．

SA での解の表現方法は GA と同様に割当順序を表す順列とした．従って，図 7-4 での  $V_c$  と  $V_n$  は，メニュー項目  $I_i$  に対応する番号の順列である．近傍点は，GA で用いた突然変異操作と同じ，ランダムに選んだ順列の 2 要素を入れ替える操作とした．

予備実験の結果に基づいて，パラメータは以下の設定で計算を行った．

初期温度：  $T_{max} = 10$

最低温度：  $T_{min} = 0.001$

冷却率  $\gamma = 0.9$

冷却間隔：  $k_T = 100$

### 7.6.2 オリジナルデータでの比較結果

前章と同様な設定で計算を行った．すべてのケースについて，50 回ずつ計算を行った．図 7-5 に平均到達時間の比較結果を，箱ヒゲ図の形式で示す．これから以下が分かる．

**STEP1:**  $T \leftarrow T_{max}$

ランダムな順列  $V_c$  を生成し暫定解とする．

**STEP2:**  $V_c$  の近傍点  $V_n$  を選ぶ．

if  $eval(V_n)$  が  $eval(V_c)$  より良い解

then 近傍点を暫定解とする ( $V_c \leftarrow V_n$ ) ．

else 確率  $e^{-\frac{\Delta eval}{T}}$  で近傍点を暫定解とする．

repeat  $k_T$  回繰り返す．

**STEP3:**  $T \leftarrow \gamma T$  とする．

if  $T \geq T_{min}$  then STEP2 に， else STEP1 に

図 7-4 SA に基づく解法の概要



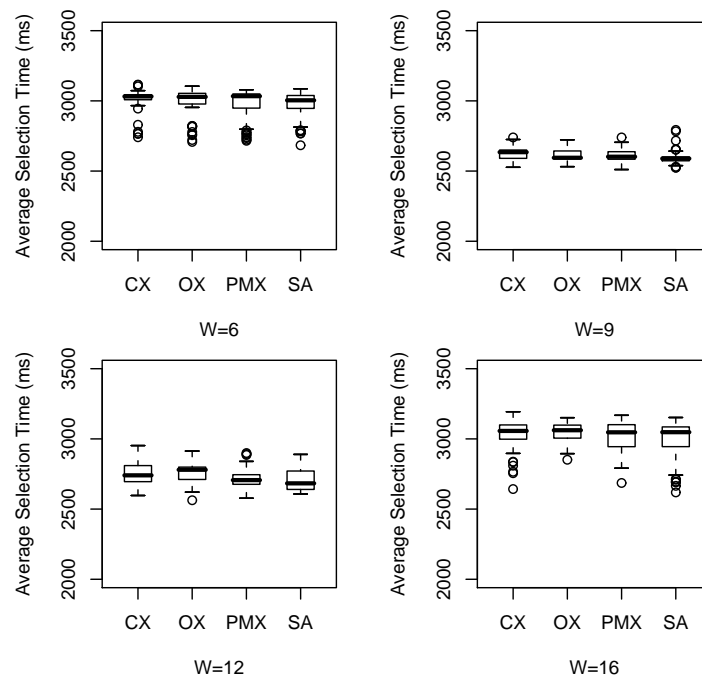


図 7-5 平均到達時間の比較（オリジナル）

- 平均値でみると， $W = 12, 9, 6$  の場合には，SA に基づく解法の性能が若干高い． $W = 16$  の場合には，交叉方式として PMX を用いる GA の性能が若干高い．しかし，統計的検定による有意差はない．
- 平均到達時間のばらつきの観点からは，解法による差は大きくない．ただし， $W = 9$  の場合に SA に基づく解法のばらつきが大きい．
- 平均値には統計的な有意差がないこと，ばらつきが小さいこと，最良解が他と遜色がないことから CX を用いる GA の性能は十分であると考えられる．

### 7.6.3 他のデータセットでの比較結果

図 7-6 から図 7-9 に箱ヒゲ図の形式で，“Mail2”，“Mail3”，“Web2”，“Web3” データセットに対する性能比較結果を示す．これらの図から以下がいえる．

- 平均到達時間の観点からは SA による解法の平均値が若干小さい．最良解の観点からは，GA に基づく解法の方が若干性能が高い
- “Mail2” と “Mail3” データセットに対しては，GA に基づく解法により，平均到達時間が最小の解が得られている．

GA と SA で得られる解に大きな差が生じない理由としては，SA と GA で用いている解の表現方式が同じものであることが挙げられる．SA の近傍生成は GA での突然変異と同じものであることから，SA は交叉のない GA と同じような探索を行うためと考えられる．

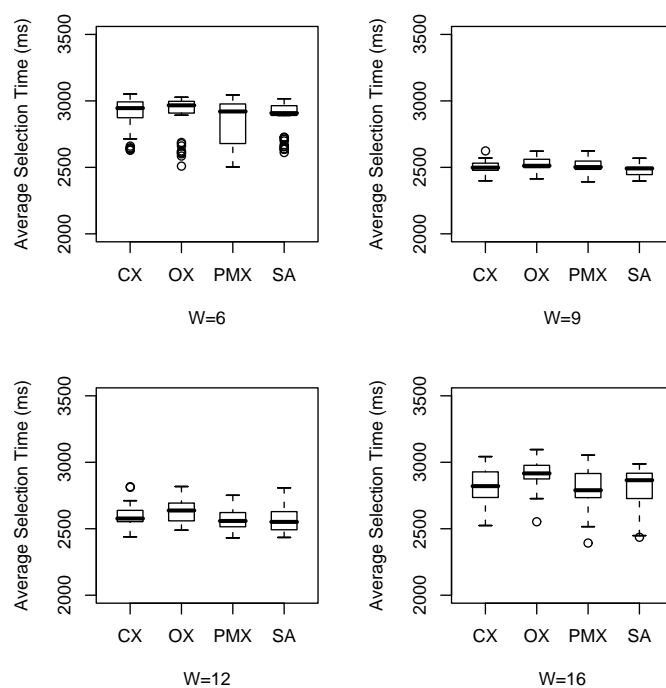


図 7-6 Mail2 データセットでの平均到達時間の比較

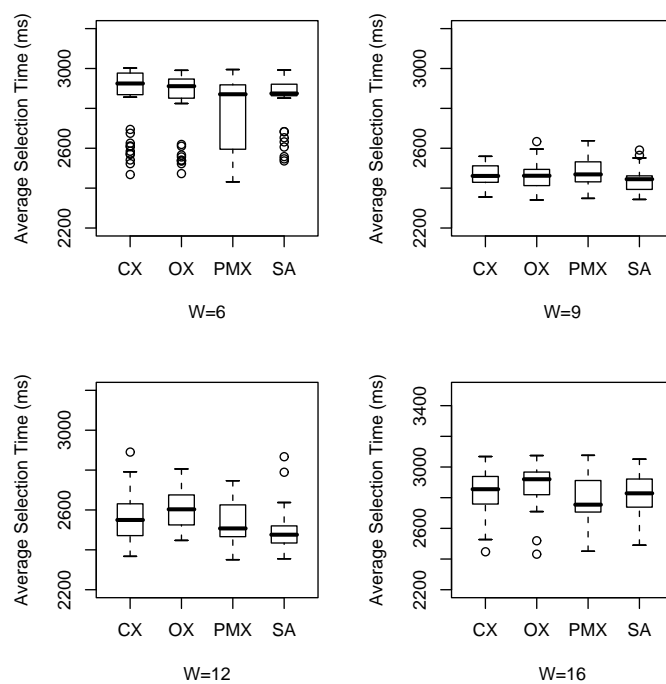


図 7-7 Mail3 データセットでの平均到達時間の比較

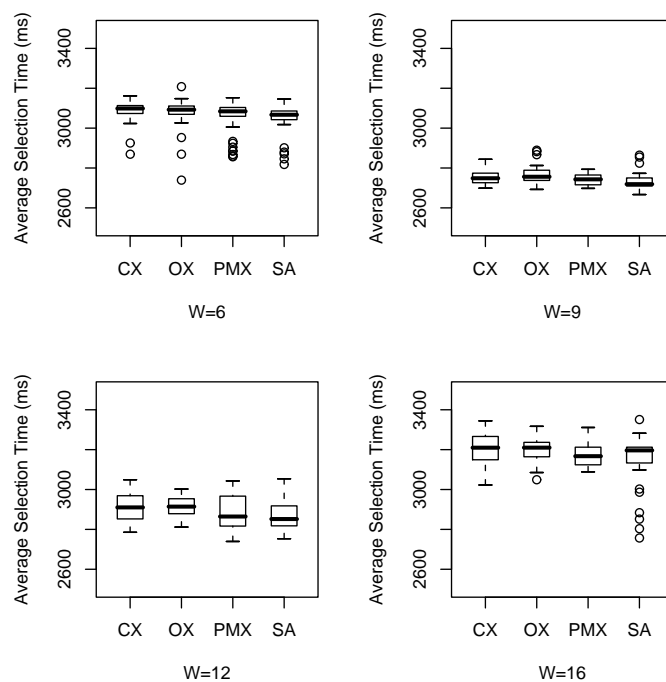


図 7-8 Web2 データセットでの平均到達時間の比較

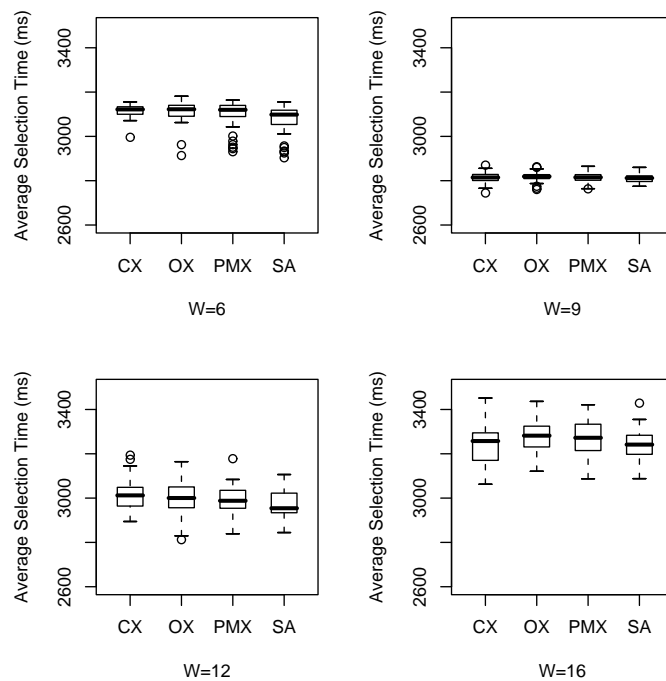


図 7-9 Web3 データセットでの平均到達時間の比較

## 7.7 解法の性能向上方策

本論文では，遺伝的アルゴリズムを用いた解法を提案し，解法の性能について，交叉方式の比較，SA との比較を行い，提案手法が実用的には十分な性能を持つことを示した．しかしながら，解法の性能，頑健性をさらに向上させる余地は残されていると考えられる．

提案手法をオリジナルの利用頻度データに対して  $W = 9$  で適用した場合の，収束過程の例を示す．目的関数値と平均到達時間について，50 回の試行の中の最良解と最悪解についてプロットしたものが図 7-10 である．この図を見ると，計算の初期段階では最悪解の方が目的関数値が小さいが，計算が進んでも解の改良が少なくなる．これは「初期収束」の状態に陥っていると考えることができる．平均到達時間でみると，この初期収束が早い時点で生じていることが分かる．

提案手法では，本来は巡回セールスマン問題のための考案された交叉方式を採用しているが，問題に合わせた交叉方式を考案することで，より性能が向上できる可能性がある．

一般的には，遺伝的アルゴリズムでは，交叉率，突然変異率などのいくつかのパラメータを適切に設定する必要がある．これらを動的に，あるいは適応的に制御する方式も提案されている．提案手法では，予備的検討結果を基に，固定した値を用いたが，動的あるいは適応的な方式を採用することで，初期収束の問題が解決できる可能性もある．

解法自体の性能，頑健性のさらなる向上は今後の課題の一つである．

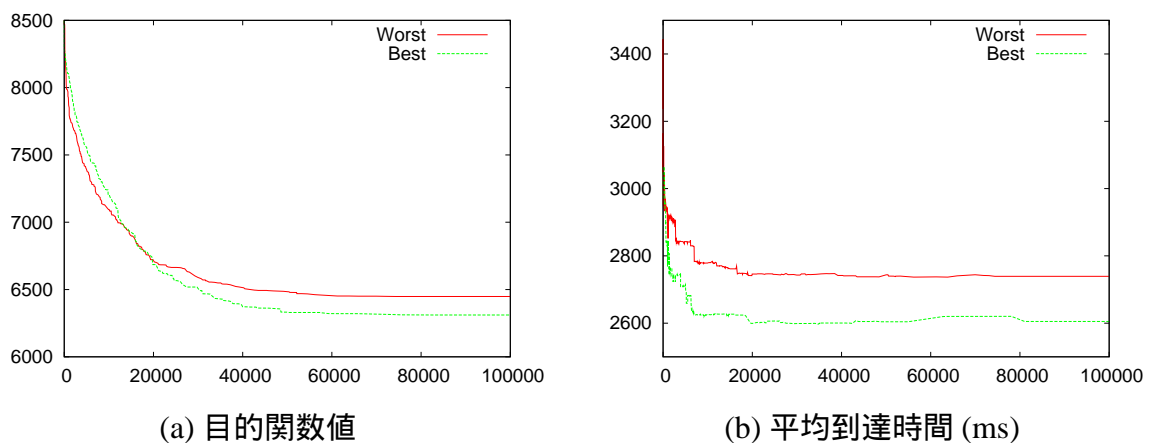


図 7-10 GA での収束過程の例 (交叉方式: CX,  $W = 9$ )

## 第8章 結論

階層メニューは、GUIでコマンドを指定する目的でオフィスアプリケーション、携帯電話、Webアプリケーションなどの様々な分野で広く用いられている。階層メニューは機能が配置されたメニュー項目と下位のメニューを持つメニュー項目が複数配置された構造を持つ。本論文では、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した。

この性能は、メニューの構造、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。また、単一階層のメニューの最適化についてはいくつかの研究が行われてきたが、階層メニューの構造を変更することで、性能を向上する最適化問題としてとらえた研究はほとんどなかった。従来の研究で扱われてきたのは、単一階層のメニューのみであり、また階層メニューの設計問題では使い勝手を明示的に考慮していない。

これに対して、本論文では、従来最適化問題として扱われてこなかった階層メニューの設計問題を、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、かつ使い勝手を表す指標も目的関数に採り入れて、平均到達時間と使い勝手を表す指標の加重和を最小化する問題として定式化した。定式化では、ユーザが理想的な行動をした場合の平均到達時間を考えた。すなわち、ユーザは目的とする項目を誤りなく選択することを前提とした。つまり、ユーザは目的の項目のあるサブメニューまで誤りなく辿り着くものとし、サブメニュー内の項目を誤りなく選択するものと仮定した。この仮定の下で、ユーザの利用頻度に従って、平均到達時間を最小化するメニューを設計する問題として定式化した。

対象とする問題は、非常に複雑な組み合わせ最適化問題であり、整数計画法などの厳密解法では、解を求めるために項目数の階乗に比例する時間が必要であり、現実的な時間では解を求めることができない。そこで、良い近似解が短時間で求まる解法を考案した。提案する解法は、遺伝的アルゴリズムに基づくものである。提案方式では浅いメニューが生成されるように、ノードに項目を割り当てていく割当順序を求める方式を採用した。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方式を提案した。

単一階層のメニューに対しては既存研究でその妥当性、近似精度が検証されているが、複数階層での検証は行われていないことから、モデル化の妥当性、近似精度を検

証するために、PDA を用いた被験者実験を行った。19 個のメニュー項目からなる小規模なポップアップ型のメニューを PDA 上に表示し、10 名の被験者により平均到達時間を測定した。この結果、モデルによる平均到達時間の予測値 47% に対して、10 名の平均は 43% となり、提案するモデル化が十分な精度を持つことが確認できた。

次に、提案手法の有効性を検証した結果を示した。本論文では、ユーザは目的とする項目を誤りなく選択することを前提としたことから、被験者実験ではなく、コンピュータシミュレーションにより、提案手法の有効性を確認した結果を示した。携帯電話のメニューを対象として、利用パターン、モデルで用いるパラメータを様々な値に設定した実験を行い、提案手法により時間を 40% 以上短縮できるメニュー構造を生成できることを確認した。また、Zipf 分布を用いて多様な利用パターンを生成し、多様な利用パターンに対しても、提案手法は平均到達時間を 40% から 60% 程度短縮できることを示し、利用パターンに対する頑強性を確認した。

以上により、本論文では、従来最適化問題として扱われてこなかった階層メニューの設計問題を、平均到達時間と使い勝手を表す指標の加重和を最小化する問題として新たに定式化し、遺伝的アルゴリズムを用いた新しい解法を提案し、モデルの精度を被験者実験で確認し、さらに多様な利用パターンに対して提案手法が有効であることを示した。

考察において、提案手法の有効性と適用可能領域について示した。提案手法はコンテキストメニューを含む GUI で現れる多様なメニューに適用可能であることを示し、拡張の方向性を示した。また提案手法と適用干渉との関係を述べ、新しく生成されるメニューへの適用も考慮した定式化の方向性を示した。

また、提案手法では明示的に考慮していないユーザのエラーモデルについて述べた。階層メニューで発生するエラーには、辿るべき階層を間違えるエラーとポインティングエラーが考えられるが、本論文のモデルでは、これらのエラーは考慮していない。利用頻度の高いメニュー項目は浅い階層に置かれることから、利用頻度の高いメニュー項目に対しては、前者のエラーからの回復コストは低いと考えられるが、利用頻度の低い項目に対しては、回復コストが増加する可能性があることを示した。

ポインティングエラーは、最適化の有無に因らず発生するものであるが、ターゲットまでの距離とターゲットの幅、さらにポインティング時間によりエラー率が異なる。この特質も考慮したユーザのエラーモデルも組み込んだ最適設計は今後の課題であることを示した。

さらに、浅いメニューと深いメニューどちらが良いかは提案手法により計算で判定できることを示した。加えて、GA に基づく提案手法の性能を確認するために行った、シミュレーテッドアニーリングに基づく解法、交叉方式を変えた GA に基づく解法と比較した結果から、提案手法の性能を確認した結果を示し、解法の性能改善の方向性について示した。

## 謝辞

本学位論文をまとめるにあたって、有益な助言を賜った以下の方々に深謝いたします。

国立情報学研究所 山田誠二教授には、指導教官として研究のアイデア段階から論文の取りまとめの段階まで、多大な助言を賜りました。2000年の夏に GECCO が開催された Las Vegas でお会いしたのが最初の出会いでした。その後、縁あって総合研究大学院大学に入学することになり、入学の時から大変お世話になりました。大学院でのご指導に加えて、勤務先まで出向いて頂いてのご指導、さらには様々な機会での激励など、先生のご指導がなければ、本論文は完成できませんでした。論文発表に出かけた海外では、二人だけの場合には道に迷ったりしましたが、研究では迷うことなく論文を完成させることができました。心より厚く御礼申し上げます。

論文審査委員の先生方からも多大なご指導とご鞭撻を賜りました。佐藤健教授には、幅広く深い見識に基づいて、本論文の力点についての鋭く重要なご助言を賜りました。相澤彰子教授には、低利用頻度のメニュー項目は隠すといった新しい戦略についてのアイデアを賜りました。また遺伝的アルゴリズムに関する有益なご助言を賜りました。古山宣洋准教授には、認知心理学に関する深い見識に基づいた、ご指摘を賜りました。相原健郎准教授には、ヒューマン・コンピュータ・インタラクションに関する深い見識に基づいたご指摘と被験者実験の重要性などのご助言を賜りました。また、審査委員の先生方には、論文の素案の段階から丁寧に読んでいただき、様々なご指摘、ご助言を賜りました。これにより、本論文をこの形にまとめることができ、より質の高いものとすることができました。重ねて御礼申し上げます。

中間発表1、中間発表2においては、国立情報学研究所の佐藤健教授、相澤彰子教授、古山宣洋准教授、相原健郎准教授、井上雅史助教の諸先生方から、それぞれの御専門の立場から、有益なご助言を賜りました。これらのご助言によって人工知能学会への論文が完成でき、本論文もまとめることができました。さらに、佐藤健教授と相澤彰子教授には、入学前の事前面接、サブアドバイザーとして大変お世話になりました。重ねて御礼申し上げます。

山田誠二研究室のメンバーにも大変お世話になりました。間瀬心博氏、Ganesan Velayathan 氏、李鵬氏、Richard Nadine 氏、村田博士氏、Anja Austermann 氏、岩本祐子氏に感謝いたします。メンバーの研究テーマと著者のテーマはかなり異質であったにも関わらず、研究室ミーティングで様々なコメントをいただきました。深謝いたします。

総合研究大学院大学での研究にご理解をいただきました，(財)電力中央研究所システム技術研究所の方々に感謝いたします．谷口治人前所長（現首席研究員），栗原郁夫所長に感謝いたします．

同僚の方々に感謝いたします．小野田崇博士，堤富士雄博士には，研究を進めるに当たっての有益な議論をさせていただきました．小野田崇博士には本研究の最初の論文を国際会議で代理発表していただきました．本論文以外の実問題への遺伝的アルゴリズムへの適用研究を共同実施し，アルゴリズムの勘どころを習得することに協力して頂いた，所健一博士と渡邊勇氏にも感謝いたします．

特に，篠原靖志博士と小野田崇博士には，研究所の業務の多くを代行していただきました．また，二方厚志氏には，プロジェクト研究の業務の多くを代行していただきました．ここに深謝いたします．さらに，被験者実験に協力して頂いた同僚の方々にも深謝いたします．

最後に，これまでの研究と本論文の作成は，いずれも家族の支援の上で成し遂げた成果であり，心の支えとなり元気づけてくれた家族に感謝します．



## 参考文献

- [足立 00] 足立 進, 澤井 秀文: 並列分散パラメータフリー遺伝的アルゴリズムにおける移民選択法の効果, 電子情報通信学会論文誌 (D-I), Vol. J83-D-I, No. 8, pp. 834–843 (2000)
- [Ahlström 05] Ahlström, D.: Modeling and Improving Selection in Cascading Pull-down Menus Using Fitts' Law, the Steering Law and Force Fields, in *CHI '05: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 61–70, ACM Press (2005)
- [Amant 04] Amant, R. S., Horton, T. E., and Ritter, F. E.: Model-based Evaluation of Cell Phone Menu Interaction, in *CHI '04: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 343–350, ACM Press (2004)
- [Amant 07] Amant, R. S., Horton, T. E., and Ritter, F. E.: Model-based Evaluation of Expert Cell Phone Menu Interaction, *ACM Transaction on Computer-Human Interaction*, Vol. 14, No. 1, pp. 1–23 (2007)
- [Anderson 93] Anderson, J. R.: *Rules of the Mind*, Lawrence Erlbaum Association (1993)
- [Anderson 97] Anderson, J. R., Matessa, M., and Lebiere, C.: ACT-R: A Theory of Higher Level Cognition and Its Relation to Visual Attention, *Human-Computer Interaction*, Vol. 12, No. 4, pp. 439–462 (1997)
- [Atallah 98] Atallah, M. (ed.): *Algorithms and Theory of Computation Handbook*, CRC Press (1998)
- [Baeza-Yates 99] Baeza-Yates, R. and Ribeiro-Neto, B.: *Modern Information Retrieval*, Addison-Wesley (1999)
- [Beck 06] Beck, J., Han, S. H., and Park, J.: Presenting a Submenu Window for Menu Search on a Cellular Phone, *International Journal of Human-Computer Interaction*, Vol. 20, No. 3, pp. 233–245 (2006)
- [Bierwirth 96] Bierwirth, C., Mattfeld, D. C., and Kopfer, H.: On Permutation Representations for Scheduling Problems, in *PPSN IV: Proceedings of the 4th International Conference on Parallel Problem Solving from Nature*, pp. 310–318, Springer-Verlag (1996)
- [Bridle 06] Bridle, R. and McCreath, E.: Inducing Shortcuts on a Mobile Phone Interface, in *IUI '06: Proceedings of the 11th International Conference on Intelligent User Interfaces*, pp. 327–329, ACM (2006)

- [Card 83] Card, S. K., Moran, T. P., and Newell, A.: *The Psychology of Human-computer Interaction*, Lawrence Erlbaum Associates (1983)
- [Carroll 03] Carroll, J. M. (ed.): *HCI Models, Theories, and Frameworks: Toward a Multi-disciplinary Science*, Morgan Kaufmann (2003)
- [Cockburn 07] Cockburn, A., Gutwin, C., and Greenberg, S.: A Predictive Model of Menu Performance, in *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 627–636, ACM Press (2007)
- [Eagle 06] Eagle, N. and Pentland, A. S.: Reality Mining: Sensing Complex Social Systems, *Personal Ubiquitous Computing*, Vol. 10, No. 4, pp. 255–268 (2006)
- [Findlater 04] Findlater, L. and McGrenere, J.: A Comparison of Static, Adaptive, and Adaptable Menus, in *CHI '04: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 89–96, ACM Press (2004)
- [Fischer 01] Fischer, G.: User Modeling in Human–Computer Interaction, *User Modeling and User-Adapted Interaction*, Vol. 11, No. 1-2, pp. 65–86 (2001)
- [Fitts 54] Fitts, P. M.: The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement, *Journal of Experimental Psychology*, No. 47, pp. 199–210 (1954), ( *Journal of Experimental Psychology*, Vol. 121, No.3, pp.262–269 に再掲されている )
- [Francis 00] Francis, G.: Designing Multifunction Displays: An Optimization Approach, *International Journal of Cognitive Ergonomics*, Vol. 4, No. 2, pp. 107–124 (2000)
- [Francis 02] Francis, G. and Rash, C. E.: MFDTool(Version 1.3): A Software Tool for Optimizing Hierarchical Information on Multifunction Displays, Technical Report 2002-22, USAARL (2002)
- [Gajos 04] Gajos, K. and Weld, D. S.: SUPPLE: Automatically Generating User Interfaces, in *IUI '04: Proceedings of the 9th International Conference on Intelligent User Interfaces*, pp. 93–100, ACM (2004)
- [Gajos 05] Gajos, K. and Weld, D. S.: Preference Elicitation for Interface Optimization, in *UIST '05: Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, pp. 173–182, ACM (2005)
- [Gajos 08a] Gajos, K. Z., Everitt, K., Tan, D. S., Czerwinski, M., and Weld, D. S.: Predictability and Accuracy in Adaptive User Interfaces, in *CHI '08: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1271–1274, ACM (2008)
- [Gajos 08b] Gajos, K. Z., Wobbrock, J. O., and Weld, D. S.: Improving the Performance of Motor-Impaired Users with Automatically-Generated, Ability-Based Interfaces, in *CHI '08: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1257–1266, ACM (2008)

- [Geven 06] Geven, A., Sefelin, R., and Tscheligi, M.: Depth and Breadth Away from the Desktop: The Optimal Information Hierarchy for Mobile Use, in *MobileHCI '06: Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services*, pp. 157–164, ACM (2006)
- [Goldberg 89] Goldberg, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley (1989)
- [Green 04] Green, N., Kruger, J., Faldu, C., and Amant, R. S.: A Reduced QWERTY Keyboard for Mobile Text Entry, in *CHI '04: CHI '04 Extended Abstracts on Human Factors in Computing Systems*, pp. 1429–1432, ACM (2004)
- [Grefenstette 85] Grefenstette, J. J., Gopal, R., Rosmaita, B. J., and Gucht, D. V.: Genetic Algorithms for the Traveling Salesman Problem, in *Proceedings of the 1st International Conference on Genetic Algorithms*, pp. 160–168 (1985)
- [Grossman 07] Grossman, T., Kong, N., and Balakrishnan, R.: Modeling Pointing at Targets of Arbitrary Shapes, in *CHI '07: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 463–472, ACM (2007)
- [Han 01] Han, S. H., Yun, M. H., Kwahk, J., and Hong, S. W.: Usability of Consumer Electronic Products, *International Journal of Industrial Ergonomics*, Vol. 28, No. 3–4, pp. 143–151 (2001)
- [Han 04] Han, S. H., Kim, K. J., Yun, M. H., Hong, S. W., and Kim, J.: Identifying Mobile Phone Design Features Critical to User Satisfaction, *Human Factors and Ergonomics in Manufacturing*, Vol. 14, No. 1, pp. 15–29 (2004)
- [Helander 97] Helander, M., Landauer, T., and Prabhu, P. (eds.): *Handbook of Human-Computer Interaction*, Elsevier Science (1997)
- [Howes 00] Howes, A., Payne, S. J., and Woodward, A.: The Trouble with Shortcuts, in *CHI '00: CHI '00 Extended Abstracts on Human Factors in Computing Systems*, pp. 267–268, ACM Press (2000)
- [Huang 07] Huang, S.-C., Chou, I.-F., and Bias, R. G.: Empirical Evaluation of a Popular Cellular Phone 's Menu System: Theory Meets Practice, *Journal of Usability Studies*, pp. 91–108 (2007)
- [伊庭 94] 伊庭 斉志 : 遺伝的アルゴリズムの基礎 ~ GA の謎を解く ~ , オーム社 (1994)
- [John 04] John, B. E., Prevas, K., Salvucci, D. D., and Koedinger, K.: Predictive Human Performance Modeling Made Easy, in *CHI '04: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 455–462, ACM (2004)
- [Karahoca 06] Karahoca, A., Karahoca, D., Yengin, I., Yuce, B., Berkman, I., Simsek, S., Dagyar, C., and Yalcin, S.: Individual Differences in Usability of Cell Phone SMS Menus, in *Proceedings of the 5th WSEAS International Conference on Telecommunications and Informatics*, pp. 485–490 (2006)

- [KDDI 04] KDDI: Manual for CASIO W43CA (2004)
- [Keijzers 08] Keijzers, J., Ouden, den E., and Lu, Y.: Usability Benchmark Study of Commercially Available Smart Phones: Cell Phone Type Platform, PDA Type Platform and PC Type Platform, in *MobileHCI '08: Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 265–272, ACM (2008)
- [Kieras 06] Kieras, D.: A Guide to GOMS Model Usability Evaluation using GOMSL and GLEAN4 (2006), [ftp://www.eecs.umich.edu/people/rchong/kieras/GOMS/GOMSL\\_Guide.pdf](ftp://www.eecs.umich.edu/people/rchong/kieras/GOMS/GOMSL_Guide.pdf)
- [Kiger 84] Kiger, J. I.: The Depth/Breadth Trade-off in the Design of Menu-driven User Interfaces, *International Journal of Man-Machine Studies*, Vol. 20, No. 2, pp. 201–213 (1984)
- [北島 02] 北島 宗雄, 高木 英明, 山本 哲生, 張 勇兵: 潜在意味解析 (LSA) を利用した Markov 連鎖モデルによる階層メニュー探索過程の評価, *情報処理学会論文誌*, Vol. 43, No. 12, pp. 3722–3732 (2002)
- [北野 93] 北野 宏明 (編): 遺伝的アルゴリズム, 産業図書, 東京 (1993)
- [木津 99] 木津 左千夫, 澤井 秀文, 足立 進: 可変な局所集団の適応的探索を用いたパラメータフリー遺伝的アルゴリズムとその並列分散処理への拡張, *電子情報通信学会論文誌 (D-II)*, Vol. J82-D-II, No. 3, pp. 512–521 (1999)
- [小林 93] 小林 重信, 山村 雅幸: 遺伝的アルゴリズムにおけるコード化と交叉, pp. 43–60 (1993), in [北野 93]
- [Kraut 83] Kraut, R. E., Hanson, S. J., and Farber, J. M.: Command Use and Interface Design, in *CHI '83: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 120–124, ACM (1983)
- [Landauer 85] Landauer, T. K. and Nachbar, D. W.: Selection From Alphabetic and Numeric Menu Trees Using A Touch Screen: Breadth, Depth, and Width, in *CHI '85: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 73–78, ACM (1985)
- [Larson 98] Larson, K. and Czerwinski, M.: Web Page Design: Implications of Memory, Structure and Scent for Information Retrieval, in *CHI '98: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 25–32 (1998)
- [Lee 06] Lee, Y. S., Hong, S. W., Smith-Jackson, T. L., Nussbaum, M. A., and Tomioka, K.: Systematic Evaluation Methodology for Cell Phone User Interfaces, *Interacting with Computers*, Vol. 18, No. 2, pp. 304–325 (2006)
- [Linton 99] Linton, F., Joy, D., and Schaefer, H.-P.: Building User and Expert Models by Long-term Observation of Application Usage, in *UM '99: Proceedings of the Seventh International Conference on User Modeling*, pp. 129–138 (1999)

- [Liu 02] Liu, B., Francis, G., and Salvendy, G.: Applying Models of Visual Search to Menu Design, *International Journal of Human-Computer Studies*, Vol. 56, pp. 307–330 (2002)
- [Lyons 03] Lyons, K., Patel, N. J., and Starner, T.: KeyMenu: A Keyboard Based Hierarchical Menu, in *ISWC '03: Proceedings of the 7th IEEE International Symposium on Wearable Computers*, pp. 240–241, IEEE Computer Society (2003)
- [Mackenzie 99] Mackenzie, I. S., Zhang, S. X., and Soukoreff, R. W.: Text Entry Using Soft Keyboards, *Behaviour & Information Technology*, Vol. 18, No. 4, pp. 235–44 (1999)
- [MacKenzie 03] MacKenzie, I. S.: *Motor Behaviour Models for Human-Computer Interaction*, pp. 27–54 (2003), in [Carroll 03]
- [Matsui 00] Matsui, S. and Tokoro, K.: A New Genetic Algorithm for Minimum Span Frequency Assignment Problem using Permutation and Clique, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pp. 682–689, Las Vegas (2000)
- [Matsui 01] Matsui, S. and Tokoro, K.: Improving the Performance of a Genetic Algorithm for Minimum Span Frequency Assignment Problem with an Adaptive Mutation Rate and a New Initialization Method, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pp. 1359–1366, San Francisco (2001)
- [Matsui 02] Matsui, S., Watanabe, I., and Tokoro, K.: A Parameter-free Genetic Algorithm for a Fixed Channel Assignment Problem with Limited Bandwidth, in *Proceeding of 7th International Conference on Parallel Problem Solving from Nature—PPSN VII*, pp. 789–799, Granada, Spain (2002)
- [Matsui 03] Matsui, S., Watanabe, I., and Tokoro, K.: An Efficient Hybrid Genetic Algorithm for a Fixed Channel Assignment Problem with Limited Bandwidth, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2003)*, pp. 2240–2251, Chicago (2003)
- [Matsui 07] Matsui, S. and Yamada, S.: Optimization of Hierarchical Menu by Genetic Algorithm, in *Proceedings of the 8th International Symposium on Advanced Intelligent Systems (ISIS 2007)*, pp. 147–152 (2007)
- [Matsui 08a] Matsui, S. and Yamada, S.: Genetic Algorithm Can Optimize Hierarchical Menus, in *CHI '08: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Note*, pp. 1385–1388, ACM (2008)
- [Matsui 08b] Matsui, S. and Yamada, S.: A Genetic Algorithm for Optimizing Hierarchical Menus, in *WCCI '08: Proceedings of 2008 IEEE World Congress on Computational Intelligence*, pp. 2856–2863, IEEE (2008)
- [Matsui 08c] Matsui, S. and Yamada, S.: Optimizing Hierarchical Menus by Genetic Algorithm and Simulated Annealing, in *GECCO '08: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pp. 1587–1594, ACM (2008)

- [松井 00] 松井 正一, 所 健一: 割当順序表現を用いる遺伝的アルゴリズムによる移動通信における固定チャネル割当ての高速解法, 電子情報通信学会論文誌 B, Vol. J83-B, No. 5, pp. 645–653 (2000)
- [松井 01] 松井 正一, 所 健一: 許容割当ての存在しない固定チャネル割当てのための遺伝的アルゴリズム, 電子情報通信学会論文誌 B, Vol. J84-B, No. 6, pp. 982–987 (2001)
- [松井 02a] 松井 正一, 渡邊 勇, 所 健一: パラメータフリー遺伝的アルゴリズムの固定チャネル割当て問題への適用方式, 電子情報通信学会論文誌 D-I, Vol. J85-D-I, No. 4 (2002)
- [松井 02b] 松井 正一, 所 健一: 固定チャネル割当て問題のための遺伝的アルゴリズムの性能向上方式, 電子情報通信学会論文誌 D-I, Vol. J85-D-I, No. 1, pp. 10–20 (2002)
- [松井 03] 松井 正一, 渡邊 勇, 所 健一: 利用帯域が限定された固定周波数割当て問題のための効率的な遺伝的アルゴリズム, 電子情報通信学会論文誌 D-I, Vol. J86-D-I, No. 9, pp. 643–649 (2003)
- [松井 08a] 松井 正一, 山田 誠二: 階層メニューの最適化手法の提案, 第 11 回 Web インテリジェンスとインタラクション研究会, pp. 65–70 (2008)
- [松井 08b] 松井 正一, 山田 誠二: 遺伝的アルゴリズムによる階層メニューの最適化, 第 22 回人工知能学会全国大会, No. 2F2-3 (2008)
- [松井 08c] 松井 正一, 山田 誠二: 遺伝的アルゴリズムによる階層メニューの最適化, 人工知能学会論文誌, Vol. 23, No. 6B, pp. 494–504 (2008)
- [Meltzoff 95] Meltzoff, A. N.: Understanding the Intention of Others: Re-Enactment of Intended Acts by 18-Month-Old Children, *Developmental Psychology*, Vol. 31, No. 5, pp. 838–850 (1995)
- [Michalewicz 00] Michalewicz, Z. and Fogel, D. B.: *How to Solve It: Modern Heuristics*, Springer (2000)
- [Mitchell 96] Mitchell, M.: *An Introduction to Genetic Algorithms*, MIT Press (1996)
- [MMD 07] 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)
- [村上 00] 村上 誉, 小川 恭孝, 大鐘 武雄: 遺伝的アルゴリズムを用いた移動通信における固定チャネル割当て法, 電子情報通信学会論文誌, Vol. J83-B, No. 6, pp. 769–779 (2000)
- [Napier 92] Napier, H. A., Batsell, R. R., Lane, D. M., and Guadagno, N. S.: Knowledge of Command Usage in a Spreadsheet Program, *ACM SIGMIS Database*, Vol. 23, No. 1, pp. 13–21 (1992)
- [Ngo 98] Ngo, C. Y. and Li, V. O. K.: Fixed Channel Assignment in Cellular Radio Networks Using a Modified Genetic Algorithm, *IEEE Transaction on Vehicular Technology*, Vol. VT-47, No. 1, pp. 163–172 (1998)

- [JIPDEC 07] JIPDEC 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—, BCN (2007)
- [Norman 91] Norman, K. L.: *The Psychology of Menu Selection — Designing Cognitive Control at the Human/Computer Interface*, Ablex Publishing Corporation (1991)
- [Parush 04] Parush, A. and Yuviler-Gavish, N.: Web Navigation Structures in Cellular Phones: The Depth/Breadth Trade-Off Issue, *International Journal of Human-Computer Studies*, Vol. 60, No. 5-6, pp. 753–770 (2004)
- [Quiroz 07] Quiroz, J. C., Louis, S. J., and Dascalu, S. M.: Interactive Evolution of XUL User Interfaces, in *GECCO '07: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, pp. 2151–2158, ACM (2007)
- [Raento 05] Raento, M., Oulasvirta, A., Petit, R., and Toivonen, H.: ContextPhone: A Prototyping Platform for Context-aware Mobile Applications, *Pervasive Computing, IEEE*, Vol. 4, No. 2, pp. 51–59 (2005)
- [Rothlauf 00] Rothlauf, F., Goldberg, D. E., and Heinzl, A.: Bad Coding and the Utility of Well-designed Genetic Algorithms, in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pp. 355–362, Las Vegas (2000)
- [澤井 98a] 澤井 秀文, 木津 左千夫, 遠藤 哲郎: パラメータの設定を不要にした遺伝的アルゴリズム, 電子情報通信学会論文誌 (D-II), Vol. J81-D-II, No. 2, pp. 450–452 (1998)
- [澤井 98b] 澤井 秀文, 木津 左千夫, 遠藤 哲郎: パラメータフリー遺伝的アルゴリズムと定常状態遺伝的アルゴリズムの性能比較, 電子情報通信学会論文誌 (D-II), Vol. J81-D-II, No. 6, pp. 1455–1459 (1998)
- [澤井 00] 澤井 秀文, 木津 左千夫, 足立 進: パラメータ設定不要の遺伝的アルゴリズムとその並列分散処理, 第3章, pp. 79–111, 産業図書 (2000)
- [Schultz 86] Schultz, E. E. and Curran, P. S.: Menu Structure and Ordering of Menu Selection: Independent or Interactive Effects?, *ACM SIGCHI Bulletin*, Vol. 18, No. 2, pp. 69–71 (1986)
- [Silfverberg 00] Silfverberg, M., MacKenzie, I. S., and Korhonen, P.: Predicting Text Entry Speed on Mobile Phones, in *CHI '00: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 9–16 (2000)
- [Tange 98] Tange, H. J., Schouten, H. C., Kester, A. D. M., and Hasman, A.: The Granularity of Medical Narratives and Its Effect on the Speed and Completeness of Information Retrieval, *Journal of the American Medical Informatics Association*, Vol. 5, No. 6, pp. 571–582 (1998)
- [Thomas 96] Thomas, R. C.: Long-term Variation in User Actions, *ACM SIGCHI Bulletin*, Vol. 28, No. 2, pp. 36–38 (1996)
- [Toms 01] Toms, M. L., Cummings-Hill, M. A., Curry, D. G., and Cone, S. M.: Using Cluster Analysis for Deriving Menu Structures for Automotive Mobile Multimedia Ap-

- plications, Technical Report 2001-01-0359, SAE (2001)
- [Weld 03] Weld, D. S., Anderson, C., Domingos, P., Etzioni, O., Gajos, K., Lau, T., and Wolfman, S.: Automatically Personalizing User Interfaces, in *IJCAI03: Proceedings of International Joint Conferences on Artificial Intelligence 03* (2003)
- [Wobbrock 08] Wobbrock, J. O., Cutrell, E., Harada, S., and MacKenzie, I. S.: An Error Model for Pointing Based on Fitts' Law, in *CHI '08: Proceeding of the SIGCHI conference on Human factors in computing systems*, pp. 1613–1622, ACM (2008)
- [Wolfe 89] Wolfe, J. M., Cave, K. R., and Franzel, S. L.: Guided Search: An Alternative to the Feature Integration Model for Visual Search, *Journal of Experimental Psychology*, Vol. 15, No. 3, pp. 419–433 (1989)
- [Wolfe 02] Wolfe, J. M.: Visual Search (2002), <http://search.bwh.harvard.edu/pdf/THE%20REVIEW.pdf>
- [山田 07] 山田 誠二 (監・著): 人とロボットの <間> をデザインする, 東京電機大学出版局 (2007)
- [Yu 02] Yu, B.-M. and Roh, S.-Z.: The Effects of Menu Design on Information-Seeking Performance and User's Attitude on the World Wide Web, *Journal of the American Society for Information Science and Technology*, Vol. 53, No. 11, pp. 923–933 (2002)
- [ユー 07] 「ユーザビリティハンドブック」編集委員会 (編): ユーザビリティハンドブック, 共立出版 (2007)
- [Zaphiris 00] Zaphiris, P. G.: Depth vs Breadth in the Arrangement of Web Links, in *Proceedings of Human Factors and Ergonomics Society Annual Meeting*, pp. 453–456 (2000)
- [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 *Proceedings of User Interfaces for All*, pp. 23–42 (2003)
- [Ziefle 04] Ziefle, M. and Bay, S.: Mental Models of a Cellular Phone Menu. Comparing Older and Younger Novice Users, in *Proceedings of Mobile Human-Computer Interaction - MobileCHI 2004*, pp. 25–37, Springer-Verlag (2004)
- [Ziefle 06] Ziefle, M. and Bay, S.: How to Overcome Disorientation in Mobile Phone Menus: A Comparison of Two Different Types of Navigation Aids, *Human-Computer Interaction*, Vol. 21, No. 4, pp. 393–433 (2006)



## 研究業績

### 論文

1. 松井 正一, 山田 誠二：遺伝的アルゴリズムによる階層メニューの最適化, 人工知能学会論文誌, Vol. 23, No. 6B, pp. 494–504 (2008)

### 国際会議（全文査読有り）

1. Matsui, S. and Yamada, S.: Genetic Algorithm Can Optimize Hierarchical Menus, in *CHI '08: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Note*, pp. 1385–1388, ACM (2008)
2. Matsui, S. and Yamada, S.: A Genetic Algorithm for Optimizing Hierarchical Menus, in *WCCI '08: Proceedings of 2008 IEEE World Congress on Computational Intelligence*, pp. 2856–2863, IEEE (2008)
3. Matsui, S. and Yamada, S.: Optimizing Hierarchical Menus by Genetic Algorithm and Simulated Annealing, in *GECCO '08: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*, pp. 1587–1594, ACM (2008)

### 国際会議（全文査読無し）

1. Matsui, S. and Yamada, S.: Optimization of Hierarchical Menu by Genetic Algorithm, in *Proc. of 8th International Symposium on Advanced Intelligent Systems (ISIS 2007)*, pp. 147–152 (2007)

### 国内会議

1. 松井 正一, 山田 誠二：遺伝的アルゴリズムによる階層メニューの最適化, 第 22 回 人工知能学会全国大会, No. 2F2-3 (2008)
2. 松井 正一, 山田 誠二：階層メニューの最適化手法の提案, 第 11 回 Web インテリジェンスとインタラクション研究会, pp. 65–70 (2008)



## 付 録 A 被験者実験で用いたデータ

表 A-1 に PDA を用いた被験者実験で使ったメニューの構造（オリジナル）と各項目の利用頻度（%）を示す。

表 A-1 で頻度が 0% のものが存在するのは、オリジナルのメニュー構造での項目の位置をそのまま再現するために残したためである。

表 A-1 PDA による被験者実験で用いたメニューの構造と項目の利用頻度

レベル 0	レベル 1	レベル 2	レベル 3	レベル 4	利用頻度 (%)
Start	Email	CreateNew			0
		CreNewDeco			0
		Inbox	Private		1
			Job		47
			Family		8
			Misc		2
			ML		7
			Fol6		0
			Fol7		0
			Fol8		0
			Fol9		0
			Junk		3
		Outbox			0
		CheckNewMail			3
	Ezweb	Main			
		Favorite Site	Daily	Microsoft	0
				Yahoo	1
				Google	4
				EkiTan	2
				Infoseek	1
				PC-Watch	5
				I-Watch	1
				EC-Navi	1
				Twotop	1
				JAL	1
				ANA	1
				UA	1
				CRIEPI	1
			Reference		0
			Misc		0
	Appli				0
	SC				0



## 付 録 B シミュレーション実験で用いたデータ

### B.1 想定した携帯電話での困難度の指標 ( $ID$ )

表 B-1 ～ 表 B-7 にシミュレーション実験で想定した携帯電話のキー配置 (図 B-1) での困難度の指標を示す．表 B-1 ～ 表 B-7 で， $A_i$  は from から to までの距離 (mm) で， $W_i$  は to の高さ (from から to までの移動が垂直的な場合) あるいは幅 (from から to までの移動が水平的な場合 (mm) である．表 B-1 ～ 表 B-7 では，図 B-1 に示したボタンを，上から順に，左から右の順で“App”，“Up”，“Addr”，“Left”，“Center”，“Right”，“Mail”，“Down”，“Web”，“Call”，“Clear”，“Power”，“1”，“2”，“3”，“4”，“5”，“6”，“7”，“8”，“9”，“\*”，“0”，“#” と表記している．

$A_i$  は端末のボタンの中心間の距離を測定して求めた値であり， $W_i$  はボタンの高さと幅を測定して求めた値である．測定精度は小数点以下 1 桁と仮定した．このため， $ID$  の値も小数点以下 1 桁に丸めた形で示してある．



図 B-1 対象とした携帯電話のキー配列 (図 6-1 の再掲)

表 B-1 想定した携帯電話での困難度の指標 ( $ID$ ) (その 1)

$ID$	from	to	$A_i$	$W_i$
3.7	*	Up	58.2	5.0
3.7	#	Up	58.2	5.0
3.6	0	Up	56.8	5.0
3.6	*	R	54.5	5.0
3.6	#	Left	54.5	5.0
3.5	9	Up	49.9	5.0
3.5	7	Up	49.9	5.0
3.5	0	Right	51.7	5.0
3.5	0	Left	51.7	5.0
3.5	*	Left	51.9	5.0
3.5	#	Right	51.9	5.0
3.4	App	#	63.5	6.5
3.4	Addr	*	63.5	6.5
3.4	9	Left	46.5	5.0
3.4	8	Up	48.2	5.0
3.4	7	Right	46.5	5.0
3.4	0	Down	46.1	5.0
3.4	*	Down	47.8	5.0
3.4	#	Down	47.8	5.0
3.3	Up	0	56.8	6.5
3.3	Up	*	58.2	6.5
3.3	Up	#	58.2	6.5
3.3	App	9	55.9	6.5
3.3	App	0	58.9	6.5
3.3	App	*	56.9	6.5
3.3	Addr	7	55.9	6.5
3.3	Addr	0	58.9	6.5
3.3	Addr	#	56.9	6.5
3.3	9	Right	43.5	5.0
3.3	8	Right	43.2	5.0
3.3	8	Left	43.2	5.0
3.3	7	Left	43.5	5.0
3.3	*	Power	42.8	5.0
3.3	*	Addr	63.5	7.0
3.3	#	Call	42.8	5.0
3.3	#	App	63.5	7.0
3.2	Right	0	51.7	6.5
3.2	Right	*	54.5	6.5
3.2	Right	#	51.9	6.5
3.2	Left	0	51.7	6.5
3.2	Left	*	51.9	6.5

$ID$	from	to	$A_i$	$W_i$
3.2	Left	#	54.5	6.5
3.2	Web	*	54.1	6.5
3.2	Mail	#	54.1	6.5
3.2	Center	0	51.4	6.5
3.2	Center	*	53.0	6.5
3.2	Center	#	53.0	6.5
3.2	9	Down	39.6	5.0
3.2	9	App	55.9	7.0
3.2	7	Down	39.6	5.0
3.2	7	Addr	55.9	7.0
3.2	6	Up	41.6	5.0
3.2	5	Up	39.6	5.0
3.2	4	Up	41.6	5.0
3.2	0	App	58.9	7.0
3.2	0	Addr	58.9	7.0
3.2	*	App	56.9	7.0
3.2	#	Addr	56.9	7.0
3.1	Web	0	48.6	6.5
3.1	Up	9	49.9	6.5
3.1	Up	8	48.2	6.5
3.1	Up	7	49.9	6.5
3.1	Mail	0	48.6	6.5
3.1	Down	*	47.8	6.5
3.1	Down	#	47.8	6.5
3.1	App	8	50.7	6.5
3.1	App	7	48.3	6.5
3.1	App	6	48.7	6.5
3.1	Addr	9	48.3	6.5
3.1	Addr	8	50.7	6.5
3.1	Addr	4	48.7	6.5
3.1	8	Down	37.5	5.0
3.1	6	Left	38.7	5.0
3.1	4	Right	38.7	5.0
3.1	0	Power	36.7	5.0
3.1	0	Center	51.4	7.0
3.1	0	Call	36.7	5.0
3.1	*	Clear	36.7	5.0
3.1	*	Web	54.1	7.0
3.1	*	Center	53.0	7.0
3.1	#	Clear	36.7	5.0
3.1	#	Mail	54.1	7.0

表 B-2 想定した携帯電話での困難度の指標 (ID) (その2)

ID	from	to	$A_i$	$W_i$
3.1	#	Center	53.0	7.0
3.0	Right	7	46.5	6.5
3.0	Left	9	46.5	6.5
3.0	Web	7	46.9	6.5
3.0	Web	#	46.1	6.5
3.0	Mail	9	46.9	6.5
3.0	Mail	*	46.1	6.5
3.0	Down	0	46.1	6.5
3.0	Center	9	44.7	6.5
3.0	Center	7	44.7	6.5
3.0	App	Power	36.1	5.0
3.0	Addr	Call	36.1	5.0
3.0	9	Call	36.2	5.0
3.0	9	Addr	48.3	7.0
3.0	8	App	50.7	7.0
3.0	8	Addr	50.7	7.0
3.0	7	Power	36.2	5.0
3.0	7	App	48.3	7.0
3.0	6	Right	35.0	5.0
3.0	6	App	48.7	7.0
3.0	5	Right	34.6	5.0
3.0	5	Left	34.6	5.0
3.0	4	Left	35.0	5.0
3.0	4	Addr	48.7	7.0
3.0	0	Clear	34.4	5.0
3.0	0	Web	48.6	7.0
3.0	0	Mail	48.6	7.0
3.0	*	Call	34.4	5.0
3.0	#	Power	34.4	5.0
2.9	Right	9	43.5	6.5
2.9	Right	8	43.2	6.5
2.9	Power	*	42.8	6.5
2.9	Left	8	43.2	6.5
2.9	Left	7	43.5	6.5
2.9	Web	8	40.6	6.5
2.9	Web	4	40.4	6.5
2.9	Up	6	41.6	6.5
2.9	Up	4	41.6	6.5
2.9	Mail	8	40.6	6.5
2.9	Mail	6	40.4	6.5
2.9	Center	8	42.8	6.5
2.9	Call	#	42.8	6.5

ID	from	to	$A_i$	$W_i$
2.9	App	5	42.6	6.5
2.9	App	3	42.0	6.5
2.9	Addr	5	42.6	6.5
2.9	Addr	1	42.0	6.5
2.9	9	Mail	46.9	7.0
2.9	9	Center	44.7	7.0
2.9	7	Web	46.9	7.0
2.9	7	Center	44.7	7.0
2.9	6	Down	31.5	5.0
2.9	4	Down	31.5	5.0
2.9	3	Left	31.4	5.0
2.9	3	Up	33.5	5.0
2.9	1	Right	31.4	5.0
2.9	1	Up	33.5	5.0
2.9	*	Mail	46.1	7.0
2.9	#	Web	46.1	7.0
2.8	Right	4	38.7	6.5
2.8	Left	6	38.7	6.5
2.8	Web	Call	30.6	5.0
2.8	Web	9	37.6	6.5
2.8	Up	5	39.6	6.5
2.8	Mail	Power	30.6	5.0
2.8	Mail	7	37.6	6.5
2.8	Down	9	39.6	6.5
2.8	Down	8	37.5	6.5
2.8	Down	7	39.6	6.5
2.8	App	4	39.7	6.5
2.8	Addr	6	39.7	6.5
2.8	9	Clear	28.8	5.0
2.8	8	Power	28.8	5.0
2.8	8	Web	40.6	7.0
2.8	8	Mail	40.6	7.0
2.8	8	Center	42.8	7.0
2.8	8	Call	28.8	5.0
2.8	7	Clear	28.8	5.0
2.8	6	Mail	40.4	7.0
2.8	6	Call	30.7	5.0
2.8	5	Down	28.9	5.0
2.8	5	App	42.6	7.0
2.8	5	Addr	42.6	7.0
2.8	4	Power	30.7	5.0
2.7	3	*	36.2	6.5

表 B-3 想定した携帯電話での困難度の指標 ( $ID$ ) (その 3)

$ID$	from	to	$A_i$	$W_i$
2.7	1	Power	26.9	5.0
2.7	1	Left	26.7	5.0
2.7	1	#	36.2	6.5
2.7	*	3	36.2	6.5
2.7	#	1	36.2	6.5
2.6	Right	Call	24.8	5.0
2.6	Power	Left	24.8	5.0
2.8	4	Web	40.4	7.0
2.8	3	App	42.0	7.0
2.8	2	Up	31.0	5.0
2.8	1	Addr	42.0	7.0
2.7	Right	6	35.0	6.5
2.7	Right	5	34.6	6.5
2.7	Power	7	36.2	6.5
2.7	Power	0	36.7	6.5
2.7	Power	#	34.4	6.5
2.7	Left	5	34.6	6.5
2.7	Left	4	35.0	6.5
2.7	Clear	0	34.4	6.5
2.7	Clear	*	36.7	6.5
2.7	Clear	#	36.7	6.5
2.7	Web	1	34.8	6.5
2.7	Mail	3	34.8	6.5
2.7	Center	6	36.5	6.5
2.7	Center	4	36.5	6.5
2.7	Call	9	36.2	6.5
2.7	Call	0	36.7	6.5
2.7	Call	*	34.4	6.5
2.7	App	Clear	27.3	5.0
2.7	App	2	34.7	6.5
2.7	Addr	Clear	27.3	5.0
2.7	Addr	2	34.7	6.5
2.7	9	Web	37.6	7.0
2.7	7	Mail	37.6	7.0
2.7	6	Addr	39.7	7.0
2.7	4	App	39.7	7.0
2.7	3	Right	26.7	5.0
2.7	3	Call	26.9	5.0
2.6	Power	Up	25.8	5.0
2.6	Power	Call	25.4	5.0
2.6	Power	App	36.1	7.0

$ID$	from	to	$A_i$	$W_i$
2.6	Left	Power	24.8	5.0
2.6	Web	5	32.8	6.5
2.6	Up	Power	25.8	5.0
2.6	Up	Call	25.8	5.0
2.6	Up	3	33.5	6.5
2.6	Up	1	33.5	6.5
2.6	Mail	5	32.8	6.5
2.6	Center	5	34.2	6.5
2.6	Call	Right	24.8	5.0
2.6	Call	Power	25.4	5.0
2.6	Call	Up	25.8	5.0
2.6	Call	Addr	36.1	7.0
2.6	9	Power	25.8	5.0
2.6	8	Clear	25.8	5.0
2.6	7	Call	25.8	5.0
2.6	6	Center	36.5	7.0
2.6	5	Center	34.2	7.0
2.6	4	Center	36.5	7.0
2.6	3	Mail	34.8	7.0
2.6	2	Right	26.2	5.0
2.6	2	Left	26.2	5.0
2.6	2	App	34.7	7.0
2.6	2	Addr	34.7	7.0
2.6	1	Web	34.8	7.0
2.5	Right	1	31.4	6.5
2.5	Power	4	30.7	6.5
2.5	Left	3	31.4	6.5
2.5	Clear	Up	22.4	5.0
2.5	Web	App	33.0	7.0
2.5	Up	Clear	22.4	5.0
2.5	Up	2	31.0	6.5
2.5	Mail	Addr	33.0	7.0
2.5	Down	6	31.5	6.5
2.5	Down	4	31.5	6.5
2.5	Call	6	30.7	6.5
2.5	App	Web	33.0	7.0
2.5	App	Call	22.6	5.0
2.5	App	1	31.1	6.5
2.5	Addr	Power	22.6	5.0
2.5	Addr	Mail	33.0	7.0
2.5	Addr	3	31.1	6.5



表 B-4 想定した携帯電話での困難度の指標 ( $ID$ ) (その4)

$ID$	from	to	$A_i$	$W_i$
2.5	9	1	30.7	6.5
2.5	7	3	30.7	6.5
2.5	6	*	30.7	6.5
2.5	5	Web	32.8	7.0
2.5	5	Mail	32.8	7.0
2.5	4	#	30.7	6.5
2.5	3	Down	23.9	5.0
2.5	3	7	30.7	6.5
2.5	1	Down	23.9	5.0
2.5	1	9	30.7	6.5
2.5	*	6	30.7	6.5
2.5	#	4	30.7	6.5
2.4	Right	3	26.7	6.5
2.4	Power	Mail	30.6	7.0
2.4	Power	8	28.8	6.5
2.4	Power	1	26.9	6.5
2.4	Left	1	26.7	6.5
2.4	Clear	9	28.8	6.5
2.4	Clear	7	28.8	6.5
2.4	Web	Left	21.6	5.0
2.4	Web	Mail	31.2	7.0
2.4	Web	6	29.0	6.5
2.4	Mail	Right	21.6	5.0
2.4	Mail	Web	31.2	7.0
2.4	Mail	4	29.0	6.5
2.4	Down	5	28.9	6.5
2.4	Center	Power	21.3	5.0
2.4	Center	Call	21.3	5.0
2.4	Center	3	28.6	6.5
2.4	Center	1	28.6	6.5
2.4	Call	Web	30.6	7.0
2.4	Call	8	28.8	6.5
2.4	Call	3	26.9	6.5
2.4	App	Right	21.6	5.0
2.4	App	Addr	31.2	7.0
2.4	Addr	Left	21.6	5.0
2.4	Addr	App	31.2	7.0
2.4	9	4	26.9	6.5
2.4	9	*	26.9	6.5
2.4	7	6	26.9	6.5
2.4	7	#	26.9	6.5

$ID$	from	to	$A_i$	$W_i$
2.4	6	Clear	21.4	5.0
2.4	6	Web	29.0	7.0
2.4	6	7	26.9	6.5
2.4	6	1	26.9	6.5
2.4	5	Power	21.4	5.0
2.4	5	Call	21.4	5.0
2.4	4	Clear	21.4	5.0
2.4	4	Mail	29.0	7.0
2.4	4	9	26.9	6.5
2.4	4	3	26.9	6.5
2.4	3	Addr	31.1	7.0
2.4	3	4	26.9	6.5
2.4	3	0	28.8	6.5
2.4	2	*	28.8	6.5
2.4	2	#	28.8	6.5
2.4	1	App	31.1	7.0
2.4	1	6	26.9	6.5
2.4	1	0	28.8	6.5
2.4	0	3	28.8	6.5
2.4	0	1	28.8	6.5
2.4	*	9	26.9	6.5
2.4	*	2	28.8	6.5
2.4	#	7	26.9	6.5
2.4	#	2	28.8	6.5
2.3	Right	2	26.2	6.5
2.3	Power	9	25.8	6.5
2.3	Left	2	26.2	6.5
2.3	Clear	App	27.3	7.0
2.3	Clear	Addr	27.3	7.0
2.3	Clear	8	25.8	6.5
2.3	Web	Clear	19.5	5.0
2.3	Web	Up	18.9	5.0
2.3	Web	2	25.6	6.5
2.3	Mail	Clear	19.5	5.0
2.3	Mail	Up	18.9	5.0
2.3	Mail	2	25.6	6.5
2.3	Center	2	25.6	6.5
2.3	Call	7	25.8	6.5
2.3	App	Down	18.9	5.0
2.3	Addr	Down	18.9	5.0
2.3	9	7	25.4	6.5

表 B-5 想定した携帯電話での困難度の指標 ( $ID$ ) (その 5)

$ID$	from	to	$A_i$	$W_i$
2.3	7	9	25.4	6.5
2.3	6	4	25.4	6.5
2.3	4	6	25.4	6.5
2.3	3	Center	28.6	7.0
2.3	3	1	25.4	6.5
2.3	3	#	25.8	6.5
2.3	2	Down	20.3	5.0
2.3	2	0	25.8	6.5
2.3	1	Center	28.6	7.0
2.3	1	3	25.4	6.5
2.3	1	*	25.8	6.5
2.3	0	2	25.8	6.5
2.3	*	1	25.8	6.5
2.3	*	#	25.4	6.5
2.3	#	3	25.8	6.5
2.3	#	*	25.4	6.5
2.2	Right	Power	18.5	5.0
2.2	Right	Clear	17.9	5.0
2.2	Power	Right	18.5	5.0
2.2	Power	Down	17.3	5.0
2.2	Left	Clear	17.9	5.0
2.2	Left	Call	18.5	5.0
2.2	Clear	Right	17.9	5.0
2.2	Clear	Left	17.9	5.0
2.2	Down	Power	17.3	5.0
2.2	Down	Call	17.3	5.0
2.2	Down	3	23.9	6.5
2.2	Down	1	23.9	6.5
2.2	Call	Left	18.5	5.0
2.2	Call	Down	17.3	5.0
2.2	6	Power	17.2	5.0
2.2	5	Clear	17.2	5.0
2.2	4	Call	17.2	5.0
2.2	2	Web	25.6	7.0
2.2	2	Mail	25.6	7.0
2.2	2	Center	25.6	7.0
2.1	Power	Addr	22.6	7.0
2.1	Power	5	21.4	6.5
2.1	Clear	6	21.4	6.5
2.1	Clear	4	21.4	6.5
2.1	Web	3	20.5	6.5

$ID$	from	to	$A_i$	$W_i$
2.1	Mail	1	20.5	6.5
2.1	Center	Clear	17.0	5.0
2.1	Call	App	22.6	7.0
2.1	Call	5	21.4	6.5
2.1	9	2	21.4	6.5
2.1	8	3	21.4	6.5
2.1	8	1	21.4	6.5
2.1	7	2	21.4	6.5
2.1	6	0	21.4	6.5
2.1	5	*	21.4	6.5
2.1	5	#	21.4	6.5
2.1	4	0	21.4	6.5
2.1	3	8	21.4	6.5
2.1	2	9	21.4	6.5
2.1	2	7	21.4	6.5
2.1	1	8	21.4	6.5
2.1	0	6	21.4	6.5
2.1	0	4	21.4	6.5
2.1	*	5	21.4	6.5
2.1	#	5	21.4	6.5
2.0	Right	Mail	21.6	7.0
2.0	Right	App	21.6	7.0
2.0	Power	Center	21.3	7.0
2.0	Left	Web	21.6	7.0
2.0	Left	Addr	21.6	7.0
2.0	Web	Down	15.6	5.0
2.0	Mail	Down	15.6	5.0
2.0	Down	2	20.3	6.5
2.0	Call	Center	21.3	7.0
2.0	App	Up	15.6	5.0
2.0	Addr	Up	15.6	5.0
2.0	3	Clear	15.4	5.0
2.0	3	Web	20.5	7.0
2.0	2	Power	15.4	5.0
2.0	2	Call	15.4	5.0
2.0	1	Clear	15.4	5.0
2.0	1	Mail	20.5	7.0
1.9	Power	6	17.2	6.5
1.9	Clear	Web	19.5	7.0
1.9	Clear	Mail	19.5	7.0
1.9	Clear	5	17.2	6.5

表 B-6 想定した携帯電話での困難度の指標 ( $ID$ ) (その6)

$ID$	from	to	$A_i$	$W_i$
1.9	Up	Web	18.9	7.0
1.9	Up	Mail	18.9	7.0
1.9	Down	App	18.9	7.0
1.9	Down	Addr	18.9	7.0
1.9	Call	4	17.2	6.5
1.9	9	3	17.2	6.5
1.9	8	2	17.2	6.5
1.9	7	1	17.2	6.5
1.9	6	#	17.2	6.5
1.9	5	0	17.2	6.5
1.9	4	*	17.2	6.5
1.9	3	9	17.2	6.5
1.9	2	8	17.2	6.5
1.9	1	7	17.2	6.5
1.9	0	5	17.2	6.5
1.9	*	4	17.2	6.5
1.9	#	6	17.2	6.5
1.8	Power	Clear	12.7	5.0
1.8	Clear	Power	12.7	5.0
1.8	Clear	Center	17.0	7.0
1.8	Clear	Call	12.7	5.0
1.8	Web	Power	12.0	5.0
1.8	Mail	Call	12.0	5.0
1.8	Call	Clear	12.7	5.0
1.7	Right	Left	10.8	5.0
1.7	Power	2	15.4	6.5
1.7	Left	Right	10.8	5.0
1.7	Clear	Down	11.7	5.0
1.7	Clear	3	15.4	6.5
1.7	Clear	1	15.4	6.5
1.7	Web	Right	11.5	5.0
1.7	Web	Center	16.5	7.0
1.7	Up	Down	10.8	5.0
1.7	Up	App	15.6	7.0
1.7	Up	Addr	15.6	7.0
1.7	Mail	Left	11.5	5.0
1.7	Mail	Center	16.5	7.0
1.7	Down	Clear	11.7	5.0
1.7	Down	Web	15.6	7.0
1.7	Down	Up	10.8	5.0
1.7	Down	Mail	15.6	7.0

$ID$	from	to	$A_i$	$W_i$
1.7	Center	Web	16.5	7.0
1.7	Center	Mail	16.5	7.0
1.7	Center	App	16.5	7.0
1.7	Center	Addr	16.5	7.0
1.7	Call	2	15.4	6.5
1.7	App	Left	11.5	5.0
1.7	App	Center	16.5	7.0
1.7	Addr	Right	11.5	5.0
1.7	Addr	Center	16.5	7.0
1.7	9	5	15.4	6.5
1.7	9	0	15.4	6.5
1.7	8	6	15.4	6.5
1.7	8	4	15.4	6.5
1.7	8	*	15.4	6.5
1.7	8	#	15.4	6.5
1.7	7	5	15.4	6.5
1.7	7	0	15.4	6.5
1.7	6	8	15.4	6.5
1.7	6	2	15.4	6.5
1.7	5	9	15.4	6.5
1.7	5	7	15.4	6.5
1.7	5	3	15.4	6.5
1.7	5	1	15.4	6.5
1.7	4	8	15.4	6.5
1.7	4	2	15.4	6.5
1.7	3	5	15.4	6.5
1.7	2	6	15.4	6.5
1.7	2	4	15.4	6.5
1.7	1	5	15.4	6.5
1.7	0	9	15.4	6.5
1.7	0	7	15.4	6.5
1.7	*	8	15.4	6.5
1.7	#	8	15.4	6.5
1.6	9	8	12.7	6.5
1.6	8	9	12.7	6.5
1.6	8	7	12.7	6.5
1.6	7	8	12.7	6.5
1.6	6	5	12.7	6.5
1.6	5	6	12.7	6.5
1.6	5	4	12.7	6.5
1.6	4	5	12.7	6.5

表 B-7 想定した携帯電話での困難度の指標 ( $ID$ ) (その 7)

$ID$	from	to	$A_i$	$W_i$
1.6	3	2	12.7	6.5
1.6	2	3	12.7	6.5
1.6	2	1	12.7	6.5
1.6	1	2	12.7	6.5
1.6	0	*	12.7	6.5
1.6	0	#	12.7	6.5
1.6	*	0	12.7	6.5
1.6	#	0	12.7	6.5
1.4	Right	Web	11.5	7.0
1.4	Right	Addr	11.5	7.0
1.4	Power	Web	12.0	7.0
1.4	Left	Mail	11.5	7.0
1.4	Left	App	11.5	7.0
1.4	Call	Mail	12.0	7.0
1.4	3	Power	8.6	5.0
1.4	2	Clear	8.6	5.0
1.4	1	Call	8.6	5.0
1.3	Right	Up	7.6	5.0
1.3	Right	Down	7.6	5.0
1.3	Left	Up	7.6	5.0
1.3	Left	Down	7.6	5.0
1.3	Web	Addr	10.8	7.0
1.3	Up	Right	7.6	5.0
1.3	Up	Left	7.6	5.0
1.3	Mail	App	10.8	7.0
1.3	Down	Right	7.6	5.0
1.3	Down	Left	7.6	5.0
1.3	App	Mail	10.8	7.0
1.3	Addr	Web	10.8	7.0
1.2	Power	3	8.6	6.5
1.2	Clear	2	8.6	6.5
1.2	Call	1	8.6	6.5
1.2	9	6	8.6	6.5
1.2	9	#	8.6	6.5
1.2	8	5	8.6	6.5
1.2	8	0	8.6	6.5
1.2	7	4	8.6	6.5
1.2	7	*	8.6	6.5
1.2	6	9	8.6	6.5
1.2	6	3	8.6	6.5
1.2	5	8	8.6	6.5

$ID$	from	to	$A_i$	$W_i$
1.2	5	2	8.6	6.5
1.2	4	7	8.6	6.5
1.2	4	1	8.6	6.5
1.2	3	6	8.6	6.5
1.2	2	5	8.6	6.5
1.2	1	4	8.6	6.5
1.2	*	7	8.6	6.5
1.2	0	8	8.6	6.5
1.2	#	9	8.6	6.5
1.1	Center	Right	5.4	5.0
1.1	Center	Left	5.4	5.0
1.1	Center	Up	5.4	5.0
1.1	Center	Down	5.4	5.0
0.8	Right	Center	5.4	7.0
0.8	Left	Center	5.4	7.0
0.8	Up	Center	5.4	7.0
0.8	Down	Center	5.4	7.0
0.0	Right	Right	0.0	5.0
0.0	Power	Power	0.0	5.0
0.0	Left	Left	0.0	5.0
0.0	Clear	Clear	0.0	5.0
0.0	Web	Web	0.0	7.0
0.0	Up	Up	0.0	5.0
0.0	Mail	Mail	0.0	7.0
0.0	Down	Down	0.0	5.0
0.0	Center	Center	0.0	7.0
0.0	Call	Call	0.0	5.0
0.0	App	App	0.0	7.0
0.0	Addr	Addr	0.0	7.0
0.0	9	9	0.0	6.5
0.0	8	8	0.0	6.5
0.0	7	7	0.0	6.5
0.0	6	6	0.0	6.5
0.0	5	5	0.0	6.5
0.0	4	4	0.0	6.5
0.0	3	3	0.0	6.5
0.0	2	2	0.0	6.5
0.0	1	1	0.0	6.5
0.0	0	0	0.0	6.5
0.0	*	*	0.0	6.5
0.0	#	#	0.0	6.5

## B.2 メニュー構造と利用頻度

表 B-8～表 B-10 にシミュレーション実験で用いたオリジナルのメニューの構造と、各項目の利用頻度を示す。

表 B-8 携帯電話のメニューの構造と項目の利用頻度（その 1）

レベル 0	レベル 1	レベル 2	レベル 3	レベル 4	利用頻度
Root	E-Mail	Create New Mail			10
		Create New Decoration Mail			1
		InBox	Folder1		100
			Folder2		2500
			Folder3		500
			Folder4		100
			Folder5		500
			Folder6		50
			Folder7		50
			Folder8		50
			Folder9		50
			Folder0		200
		Outbox			50
		Check New Mail			200
		Mail Recipe			1
		Company Box			5
		InfoBox			5
		Check Backup Mail			1
		E-Mail Setup	Receive Display	Auto Receive	1
				Receiving Order	1
				Receive in BG	1
				Attach Auto Rcv	1
				InBox Setup	1
			Send/Create	Add Reply Ahead	1
				Addressor Name	1
				Opening Sentence	1
				Signature	1
				Reply Message	1
				Auto Chg Address	1
				Add Num Rstr	1
			Attached Play	Inline Playing	1
				Music Playing	1
			Scroll Setup	Line Scroll	1
				Page Scroll	1
			Font Size Setup	Large	1
				Middle	1
				Small	1
				Minimum	1
			Mail Filter		1
			Other Setup		1
			Chk E-Mail Setup		1

表 B-9 携帯電話のメニューの構造と項目の利用頻度（その 2）

レベル 0	レベル 1	レベル 2	レベル 3	レベル 4	利用頻度
	EZWeb	Top Menu			25
		Favorite Site	Folder 1	URL1-1	50
				URL1-2	100
				URL1-3	300
				URL1-4	200
				URL1-5	100
				URL1-6	400
				URL1-7	100
				URL1-8	100
				URL1-9	100
				URL1-10	100
				URL1-11	100
				URL1-12	100
				URL1-13	100
			Folder2	URL2-1	50
				URL2-2	5
				URL2-3	50
				URL2-4	1
				URL2-5	1
				URL2-6	1
				URL2-7	1
				URL2-8	1
			Folder3	URL3-1	30
				URL3-2	30
				URL3-3	5
				URL3-4	5
				URL3-5	5
				URL3-6	5
				URL3-7	5
				URL3-8	5
				URL3-9	5
				URL3-10	5
				URL3-11	5
				URL3-12	5
				URL3-13	5
				URL3-14	5
				URL3-15	5
				URL3-16	5
			Folder4	URL4-1	30
				URL4-2	1
				URL4-3	1
				URL4-4	1
				URL4-5	1
				URL4-6	1
				URL4-7	1
				URL4-8	1
				URL4-9	1

表 B-10 携帯電話のメニューの構造と項目の利用頻度（その3）

レベル0	レベル1	レベル2	レベル3	レベル4	利用頻度
	EZWeb	Favorite Site	Folder4	URL4-10	1
				URL4-11	1
				URL4-12	1
				URL4-13	1
				URL4-14	1
				URL4-15	1
				URL4-16	1
			Folder5	URL5-1	1
				URL5-2	1
				URL5-3	1
				URL5-4	1
				URL4-5	1
			Folder6	URL6-1	1
				URL6-2	1
				URL6-3	1
				URL6-4	1
				URL6-5	1
			Folder7	URL7-1	2
				URL7-2	2
				URL7-3	2
			Folder8	URL8-1	1
			Folder9	URL9-1	1
			Folder0	URL0-1	1
		Screen Memo	memo1		5
			memo2		5
			memo3		5
			memo4		5
			memo5		5
			memo6		5
		Pay service Info			1
		URL Input			2
		Clear Browser Record			3
		EZWeb Setup			1
		Previous Display			1
	Appli	Application1			5
		Application2			5
		Application3			10
		Application4			5
		Application5			5
		Application6			5
		Application7			5