ソフトウェア工学 105－3
（1995．9．8）

アクセスコストを最小化するハイパーテキストの構成法

高田喜朗　辻野嘉宏　都倉信樹
y-takata@ics.es.osaka-u.ac.jp
大阪大学基礎工学部情報工学科
〒560 大阪府豊中市待兼山町 1-3

ヘルプシステムや WWW(World Wide Web) など，計算機アプリケーションのいくつかの分野でハイパーテキストを利用したシステムがよく見られるようになってきている．ここでは，大量の情報（文書）が個別に提供され，それらをユーザが効率よくアクセスできるように検索のためのメニューとその間のリンクを構築する問題を考える．
ユーザが効率よくアクセスするためには，操作の手間が平均的に小さいことが必要になる．また，キーワードとそれからアクセスできるページの集合（カテゴリ）は意味的に対応づけられ，ユーザが目的のページを検索するのに途中のメニューで迷わないようにしなければならない．
本稿では，与えられたキーワードとカテゴリの関係を保つリンク構造すべての中から平均アクセス時間が最小なリンク構造を求める効率のよいアルゴリズムを示す．

A Construction Method of a Hypertext Which Minimizes the Access Cost.

Yoshiaki TAKATA, Yoshihiro TSUJINO and Nobuki TOKURA

Department of Information and Computer Sciences,
Faculty of Engineering Science, Osaka University
Toyonaka, Osaka, 560 Japan

Computer applications using hypertext becomes popular, e.g. an online help system and WWW (World Wide Web). The problem we concern is to construct a link structure when many documents are supplied independently, as in WWW. A link structure consists menus and links.
A link structure is useful if it minimizes average access cost, while preserving the semantical relation between each keyword and set of pages (category) accessed from the keyword.
In this paper, we present an efficient algorithm to find the optimal link structure which minimizes the expected access time among all structures which preserve a given semantical relation between keywords and categories.

1 まえがき

計算機アプリケーションのいくつかの分野でハイパーテキストを利用したシステムがよく見られるようになってきている．アプリケーションのヘルプシステムや CD-ROM などで配布される大容量の情報提示システム，分散データベースの一種である WWW(World Wide Web) などがその例である．

ハイパーテキストとは，各頂点が文書や情報であるような有向グラフである．典型的なハイパーテキストシステムでは，頂点の中に，出次するリンクに対応するアンカーと呼ばれるものが置かれる．ユーザはアンカーを活性化させることで対応するリンクが指す頂点にアクセスすることができる [4]．ハイパーテキストを用いることにより，ユーザは大量の情報の中を目的に応じたアンカーを選びながら眺め歩くように検索でき，ユーザがより快適に情報検索できるようにすることができる．

より検索しやすいハイパーテキストを構築するために，モデル，読者支援，著者支援，有効性評価，標準化などについてさまざまな研究がなされている [3]．ここでは著者支援に当たる，ユーザに提供する情報のハイパーテキストの構成法について考える．ここで考えるのは，大量の情報（文書）が個別に提供され，それらをユーザが効率よくアクセスできるように検索するためのメニューとその間のリンクを構築する問題である．ここでいうメニューとは，検索のために設けられたアンカーだけからなる頂点のことである．WWW などのようにネットワークの各地からさまざまな情報が個別に提供されている状況で，それらを検索しやすいように索引を作っていくことなどがこの場合に当たる．ハイパーテキストのユーザの典型的な使い方として，情報が書かれているページにアクセスした後その中のキーワードの中から興味のあるものをさらにアクセスすることがある．キーワードは，そのキーワードに関連する複数のページからなる集合（カテゴリ）に対応する．カテゴリが多数のページにまたがるような抽象度の高いものであった場合，それらのページへのアクセスを支援するためにメニューを配置する必要がある．カテゴリはその部分集合として別のカテゴリを含むことがあるので，メニューの項目は別のメニューを指すことがある（図 1）．また複数のカテゴリに含まれるような部分カテゴリも存在し得ることから，メニューが作るハイパーテキストの部分構造は一般に DAG（有向無閉路グラフ）となる．

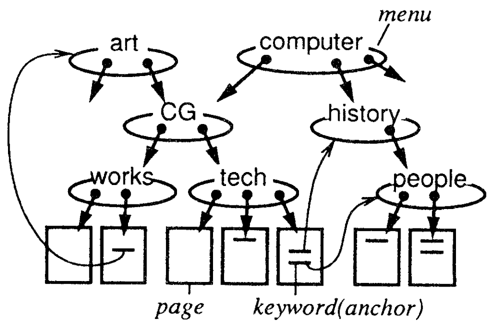


図 1: ハイパーテキスト構造

図 1で，実際の文書のページは長方形で，その検索を助けるメニューは楕円形で表している．左から2つ目のページを開いているユーザがその中の「art」というアンカーをクリックすると，art というメニューが開き，関連するページや別のメニューへのアクセスができる．この例では「CG」という項目を選ぶと別のメニューが現れる．もちろん，メニュー「art」の中にメニュー「CG」の全項目を含めればメニュー「CG」はなくてもいいかも知れないが，その場合メニューが大きくなり過ぎて，そこでの検索がしにくいこともあり得る．

ユーザが効率よくアクセスするためには，操作の手間が平均的に小さいことが必要になる．その尺度として平均アクセス時間を考える．これは適切な選択時間モデル上での，ページへのアクセス 1 回に必要な平均時間である．メニュー内のアンカーの数が多すぎても，逆に各メニューのアンカーが少なくて階層構造が深くても，どちらも平均アクセス時間が大きくなる．すなわち，前者の場合はメニューの中から必要なアンカーを探すのに，後者の場合はアンカーの選択を多数行わなければならないために時間がかかる．

また，キーワードとそれからアクセスできるページの集合（カテゴリ）は意味的に対応づけられ，ユーザが目的のページを検索するのに途中のメニューで迷わないようにしなければならない．

ハイパーテキスト構造の構築の際，設計者は実験や分析により，各ページの使用頻度と，キーワードの集合，各キーワードに対するカテゴリを与えることができると考えられる．そこでこれらの情報から，キーワードからアクセスできるページの集合とカテゴリとの対

応を保ち，平均アクセス時間が小さいハイパーテキスト構造を求める効率のよいアルゴリズムが構成できれば，設計者の負担を軽減できると期待できる．

本稿では，与えられたキーワードとカテゴリの関係を保つリンク構造すべての中から平均アクセス時間が最小なリンク構造を求める多項式時間のアルゴリズムを示す．

2 モデル

2.1 リンク構造

ここで扱うハイパーテキストの構成要素を以下に定義する．

定義 1 ユーザに提供される情報を収めた最小単位をページと呼ぶ． □

ページは 0 個以上のキーワードを含む．キーワードは，そのページをアクセスしたユーザが他のページをアクセスする際の始点となるもの（すなわちアンカー）である．ページはそれに収められた情報に関連したキーワードを含み，ユーザはそのキーワードを選択操作することで関連した情報にアクセスすることができる．

定義 2 ページ p に含まれるキーワードの集合を $key[p]$ と表す． □

複数のページの中から目的のページを選んでアクセスできるようにするために，メニューを用いる．メニューは，ページまたは他のメニューへのリンクを並べたものである．ユーザはメニュー内のリンクから適するものを選択操作することを繰り返して目的のページにアクセスする．

メニューはキーワードを名前に持つ．ページ内のキーワードを選択操作した場合はそれに対応するメニューがアクセスされ，そこからまたリンクの選択操作を繰り返す．ページ内のキーワードを名前に持つメニューを入り口と呼ぶ．

定義 3 ハイパーテキスト構造から，アンカーからメニューへのリンクを除いたものをリンク構造と呼ぶ．リンク構造は次の条件を満たす有向グラフである．

- *DAG*（有向無閉路グラフ）である．
- 頂点はメニューまたはページである．
- 辺の始点はメニューである． □

あるキーワードを選択操作して他のページにアクセスする場合，探索の範囲をそのキーワードが表す意味に関連したページの集合に限るような仕組みが当然望まれる．

意味的にグループ化されたページの集合をカテゴリと呼ぶ．キーワードはカテゴリに一対一に対応しているとする．ここでは次のような性質を満たすリンク構造を対象とする．

定義 4 リンク構造 G の任意のメニュー M から辿れるページの集合とカテゴリ M が一致するとき，G は *semantically well formed*（以下 *SWF* と略す）であると言う． □

2.2 平均アクセス時間

ここでは，あるリンク構造のページにアクセスを行うときに必要な平均的な操作時間を考える．

定義 5 1 個のメニュー M をアクセスするのに必要な平均時間を次のように定義する．

$t(M) = c(|M| + 1)/2 + s + r,$

c: 出次リンク 1 個あたりの選択判定時間，

s: 選択操作 (クリック，打鍵) 時間，

r: 計算機の反応時間．

ただし，$|M|$ は M から出るリンクの数，c, s, r はそれぞれ定数である． □

これは，メニュー内の各リンクを順に 1 個ずつ目的のものかどうか判定し目的のリンクに到達するまでの平均時間が $c(|M| + 1)/2$，その後ユーザがマウスやキーボードを操作してそのリンクを選択するのに s，計算機がその操作を受け取り次のページやメニューの表示などを行うのに r だけ時間がかかるというモデルである．これは，このリンク構造について経験の浅いユーザの選択時間の第 1 次近似モデルとして妥当であると考えられる [1, 5]．

定義 6 リンク構造 G とその頂点 M について，M と M から到達可能な頂点を頂点集合とする G の誘導部分グラフ (*induced subgraph*)[1] を $G_{<M>}$ と表し，M を根とする G の部分リンク構造という． □

定義 7 $G_{<M>}$ の平均アクセス時間 $ET[G_{<M>}]$ を次のように再帰的に定義する．

$$ET[G_{<M>}] = t(M) + \sum_{M' \in child[M]} ET[G_{<M'>}]\,P(M'|M).$$

[1] グラフ $G = (V, E)$ と頂点集合 $V' \subseteq V$ について，V' を頂点集合とする G の極大な部分グラフ $G' = (V', E')$ を，V' を頂点集合とす G の誘導部分グラフという [2]．すなわち $E' = \{(u, v) \in E | u, v \in V'\}$．

$child[M]$ は M の子である頂点の集合を表す. $P(M'|M)$ はメニュー M がアクセスされたときにその子 M' がアクセスされる割合を表す重みである. □

アクセスの重みについて次節で述べる.

定義 7は, 根から必要なアンカーを選ぶのに $t(M)$ 時間かかり, その後全部分構造の平均アクセス時間の平均（アクセスされる割合で重みづけした和）の時間がかかるというモデルである. これは葉（すなわちページ）にアクセスするまでにかかる時間の期待値となる.

定義 8 リンク構造 G の平均アクセス時間 $ET[G]$ を次のように定義する.

$$ET[G] = \sum_{q \in \mathrm{PAGE}[G]} \frac{1}{|key[q]|} \sum_{M \in key[q]} ET[G_{<M>}]\, P(q).$$

ただし, $\mathrm{PAGE}[G]$ は G 中のページの集合, $P(q)$ はページ q のアクセス確率である. □

アクセス確率については次節で述べる.

定義 8は, リンク構造の平均アクセス時間を, 各キーワードに対応するメニューを根とする部分リンク構造の平均アクセス時間を重みづけして足し合わせたものとしている. ただし, 部分リンク構造の重みはキーワードがアクセスされる重みとし, これはキーワードを含むページのアクセス確率をそのページのキーワードの数で均等分して足し合わせたものとしている. よくアクセスされるページに少数だけあるキーワードはより多くアクセスされると考えられる. このキーワードのアクセス重みで重みづけした部分リンク構造の平均アクセス時間の和は, リンク構造の平均アクセス時間の定義として妥当と考えられる.

2.3 アクセス確率

リンク構造の平均的なアクセス時間を決めるために, 各ページやメニューがアクセスされる割合を表す重みを決める必要がある. 各ページがどのような頻度でアクセスされるかというアクセス確率は, システム設計者が実験や分析によって与えることができると考えられる. そこで, 与えられたページのアクセス確率を基にメニューのアクセスの重みを定義する.

ページ q のアクセス確率を $P(q)$ とする. これはシステム設計者が与える. 確率の性質として, $P(q)$ は非負で, $\sum_{q \in \mathrm{PAGE}[G]} P(q) = 1$ を満たすように与えられるとする.

親メニュー M がアクセスされた場合に子孫の頂点 M' がアクセスされる重み $P(M'|M)$ については次の2通りの定義が考えられる.

(1) 子のアクセス確率を複数の親に分配. 子メニューまたはページ M' がアクセスされた場合に親メニュー M がアクセスされている事後確率 $P(M|M')$ を考え, これを

$$P(M|M') = 1/|parent[M']|$$

とする. ただし, $parent[M']$ は M' の親の集合である. つまり, 各親から均等な確率でアクセスされるとする.
均等分ではなく入力として与えることも可能である. この場合, 事後確率は複数の親カテゴリにまたがる子カテゴリの, 各親への関連度と言える.
メニューのアクセス確率 $P(M)$ を

$$P(M) = \sum_{M' \in child[M]} P(M|M')\, P(M')$$

とする. 親メニュー M がアクセスされた場合に子メニュー M' がアクセスされる遷移確率 $P(M'|M)$ を次のようにする.

$$P(M'|M) = P(M|M')\, P(M')/P(M).$$

このとき $\sum_{M' \in child[M]} P(M'|M) = 1$.

(2) アクセス確率を分配せず, 同じ値を親に渡す. メニューのアクセス重み $P(M)$ をそのメニューからたどれるページのアクセス確率の和と定義する. このとき

- $0 \leq P(M) \leq 1$
- M が M' の祖先の場合 $P(M) \geq P(M')$

が成り立つ. これはカテゴリのアクセス重みといえる.
親メニュー M がアクセスされた場合に子孫 M' がアクセスされる重み $P(M'|M)$ を次のようにする.

$$P(M'|M) = P(M')/P(M).$$

このとき $\sum_{M' \in child[M]} P(M'|M) \geq 1$.

2.4 縮約

ここで考える問題は，SWF であるリンク構造のうち平均アクセス時間が最小のリンク構造を求めることである．しかし，どのようなリンク構造が平均アクセス時間を最小にするかは自明ではない．たとえば，$B \subseteq A$ であるような 2 カテゴリ A, Bが与えられたとき，

- メニュー A, Bから別々にそれぞれのカテゴリの要素であるページへのリンクを置いたリンク構造，
- カテゴリ Bの要素に対しては A からのリンクを置かず代わりに A から Bへのリンクを置いたリンク構造，

ともに SWF であるが，どちらが平均アクセス時間が小さいかはカテゴリの要素数，アクセス重み，定数 c, s, rによって異なり一般的にいえない．

定義 9 与えられたカテゴリの集合に対し，次のように構成されたリンク構造を *seed* リンク構造と呼ぶ．

- 任意のカテゴリ A とその要素 p について，$p \in B$ かつ $B \subseteq A$ であるような Bが存在しないとき，メニュー A から p へのリンクを置く．
- $B \subseteq A$ であるような任意のカテゴリ A, Bについて，$B \subseteq C \subseteq A$ であるような Cが存在しないとき，メニュー A から Bへのリンクを置く．
- 上記以外のリンクを置かない． □

このように構成された seed リンク構造は，SWF でありかつ各メニュー内のアンカーが最も少ない（すなわち階層が深い）リンク構造になっている．seed リンク構造は，与えられたカテゴリの集合と一対一に対応する．

ここで，seed リンク構造に次のような制限を置く．

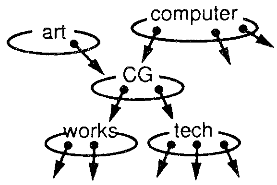
定義 10 *seed* 階層構造は「枝分かれ合流」のない *DAG* とする．「枝分かれ合流」とは，任意のメニュー Mとその子孫 M'の間に複数のパスがあることをいう． □

この制限については 4.1節で議論する．

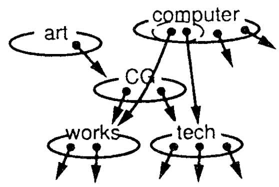
定義 11 次の操作を縮約という．

> メニューから出る 1 個のリンクを，そのリンクの先のメニューから出るリンク全てに置き換える操作． □

図 2は縮約の例である．図 2(a) のリンク構造でメニュー「computer」から「CG」へのリンクを置き換えると（「computer」に「CG」を縮約すると）図 2(b) になる．



(a)



(b)

図 2: 縮約

定義 12 *seed* リンク構造に *0* 回以上縮約を行ったリンク構造を縮約リンク構造という． □

補題 1 与えられたカテゴリの集合に対し，*SWF* であるリンク構造の集合と縮約リンク構造の集合は一致する．

(証明) 縮約リンク構造が SWF であるのは自明．

SWF であるリンク構造は，与えられたカテゴリの集合からいくつかのカテゴリを除いたものに対する seed リンク構造となっている．これは与えられた seed リンク構造に対する縮約によって得られる． □

以上より，ここで考える問題は，縮約リンク構造のうち平均アクセス時間が最小のものを求めることになる．

定義 13 与えられた *seed* リンク構造に対する縮約リンク構造のうち，平均アクセス時間が最小のものを最適リンク構造という． □

3 最適リンク構造問題

3.1 問題の定義

定義 14 最適リンク構造問題は次のように定義される．

入力
- *seed* リンク構造，
- ページのアクセス確率，
- ページ中のキーワードの集合．

出力 最適リンク構造． □

カテゴリの集合と seed リンク構造は一対一に対応するので，seed リンク構造の形で入力を与えることにする．

seed リンク構造中のメニューからメニューへのリンクそれぞれについて縮約するかしないか選べるので，縮約リンク構造の数はリンクの数に対して指数的になる．この中から最適リンク構造を効率よく求めたい．

3.2 縮約リンク構造の性質

定理 2 *seed* リンク構造 G に対する最適リンク構造は，各入り口に対する部分リンク構造をそれに対する最適リンク構造に置き換えたものに等しい．

(証明) メニュー A から B へのリンクを縮約した場合，B や B を指す他のリンクには影響しない．すなわち A を含まないような部分リンク構造の平均アクセス時間には影響しない．

また，ある部分リンク構造 $G_{<M>}$ を含むどの部分リンク構造についても，$G_{<M>}$ を $G_{<M>}$ に対する最適リンク構造に置き換えることによって平均アクセス時間が大きくならない． □

補題 3 リンク構造 G の任意のメニュー M_1 からメニュー M_2 へのリンクを縮約したとき，G の平均アクセス時間の変化は，M_1 から M_2 以外へのリンクを縮約するかどうかによらない．

(証明) M_1 から M_2 へのリンクを縮約したとき，$G_{<M_1>}$ を含まない部分リンク構造の平均アクセス時間は変化しない．$G_{<M_1>}$ を含む部分リンク構造（$G_{<M_r>}$ とする）の平均アクセス時間は，$ET[G_{<M_1>}]$ の変化を $P(M_1|M_r)$ 倍したものになる．

$ET[G_{<M_1>}]$ の変化は定義から計算すると

$$\frac{c}{2}(|M_2|-1)(1-P(M_2|M_1))-(c+s+r)P(M_2|M_1)$$

となる．これは M_1 から M_2 以外へのリンクを縮約するかどうかによらない． □

3.3 アルゴリズム

以下，部分リンク構造 G の根を $root[G]$，G に対する最適リンク構造を $\langle\langle G\rangle\rangle$ と表す．

最適リンク構造を求めるアルゴリズム OPTLS の動作を説明する．リンク構造 G とメニュー M に対し，$\langle\langle G_{<M>}\rangle\rangle$ の根に縮約されるメニューの集合を $Rdc[M]$ とする．OPTLS は，各入り口 M について，$G_{<M>}$ に対し手続き OPTSUB を行う．OPTSUB は $G_{<M>}$ の全メニューについて Rdc を求める．その後，手続き RESTR によって，Rdc を基に最適リンク構造を構成する．

OPTSUB は，再帰呼出しにより部分リンク構造中の根以外の全メニューについて Rdc を求めた後，手続き FINDR の計算結果を用いて根の Rdc を求める．同時に，部分リンク構造の全メニュー M について $g[M]=ET[\langle\langle G_{<M>}\rangle\rangle]-ET[G_{<M>}]$ を計算する．これはグローバル変数に保持し，FINDR が参照する．

FINDR は，部分リンク構造 G について $root[\langle\langle G\rangle\rangle]$ に縮約されるメニューの集合 R を求める．これは，G の各メニュー M について，M を縮約した場合と，縮約せず $G_{<M>}$ を $\langle\langle G_{<M>}\rangle\rangle$ に置き換えた場合とを比較し，前者のほうが平均アクセス時間が小さい場合に M を R に含めることによって行う．ただし，M を縮約する場合には，その子孫 M' についても同様に根に縮約するか縮約せず $\langle\langle G_{<M'>}\rangle\rangle$ に置き換えるか，最も平均アクセス時間を小さくするように選ぶとして計算する．

アルゴリズムの詳細を以下に示す．

```
/* グローバル変数 */
Rdc: array[ menu ] of set of vertex;
g: array[ menu ] of real;

OPTLS(inout G: linkStructure)
 /*  (入力)  G:  seed リンク構造,
     (出力)  G:  最適リンク構造. */
 for each M ∈ キーワードの集合
     do OPTSUB(G<M>)
 RESTR(G)

OPTSUB(in G: linkStructure)
 r ← root[G]
 if optmized[r] /* 最適化が済んでいる */
     then return
 optmized[r] ← TRUE
 Rdc[r] ← ∅
 g[r] ← 0
 if G の高さが 0
     then return
 for each M ∈ child[r]
     do OPTSUB(G<M>)
        FINDR(G, M, R, d)
        Rdc[r] ← Rdc[r] ∪ R
        g[r] ← g[r] + d
```

```
FINDR(in G: linkStructure; in M: menu; out R: set
of menu; d: real)
 /*  R:  G<M> のうち 《G》 の根に縮約されるメ
         ニューの集合,
     d:  G<M> を 《G<M>》 に置き換えたときの
         ET の差. */
 1 R1 ← {M}
 2 d1 ← c(|M| − 1)/2 − t(M)P(M|root[G])
   /* M を根に縮約したときの ET の変化量 */
 3 for each M′ ∈ child[M]
 4    do FINDR(G, M′, R′, d′)
 5        R1 ← R1 ∪ R′
 6        d1 ← d1 + d′
   /* M を縮約すると仮定したとき, 《G》 の根に
   縮約されるメニューと, G と 《G》 の ET の差*/
   /* M が葉のとき, 3–6 行目は実行されない */
 7 if d1 < g[M]P(M|root[G])
        /* G<M> を最適化した場合と比較 */
 8    then R ← R1
 9         d ← d1
10    else R ← ∅
11         d ← g[M]P(M|root[G])
```

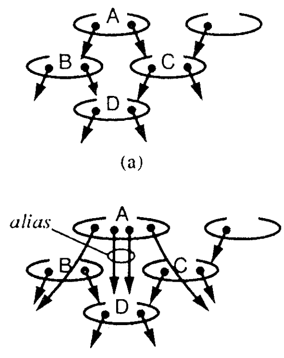
```
RESTR(inout G: linkStructure)
 /*  (入力)  G:  seed リンク構造,
     (出力)  G:  最適リンク構造. */
 1 for each M ∈ メニューの集合
 2    do for each M′ ∈ Rdc[M]
 3          do リンク (M, M′) を削除
 4             for each M″ ∈ child[M′]
 5                do リンク (M, M″) を追加
```

3.4 アルゴリズムの正当性

定理 4 アルゴリズム *OPTLS* は, 与えられた *seed* リンク構造 G から得られる縮約リンク構造のうち平均アクセス時間が最小のリンク構造を返す.

(証明) 定理 2より, OPTSUB が各部分リンク構造に対して正しく *Rdc* を求めれば題意は成り立つ.

補題 3の証明のように, メニュー M を縮約の際の平均アクセス時間の変化は, M 以外のメニューの縮約状況に関係なく FINDR の 2 行目の式となる. また, 定理 2より, 親に縮約しないメニューに対してはそれを根とする部分リンク構造を最適なものに置き換えた場合が, 最も全体の平均アクセス時間が小さい. よってアルゴリズム OPTSUB は正しく *Rdc* を求める. □



(a)

(b) Reduce B and C.

図 3: 枝分かれ合流の問題

3.5 アルゴリズムの評価

seed リンク構造のメニューの数を n とする.

補題 5 *OPTLS* の最悪時間計算量は $O(n^3)$ である.

(証明) 部分リンク構造 $G_{<M>}$ について OPTSUB が呼ばれたとき,

- OPTSUB は再帰的に $G_{<M>}$ のメニュー数だけ呼ばれる. これは $O(n)$ 個ある.
- OPTSUB から FINDR が呼び出されたとき, FINDR は再帰的に $G_{<M'>}$ (M' は M の子) のメニュー数だけ呼ばれる. 各 M' について $G_{<M'>}$ のメニューは重複していないので, OPTSUB が 1 回呼ばれたとき FINDR は $O(n)$ 回呼ばれる.

また, RESTR の計算量は $O(n^3)$ である.

よって, OPTLS の最悪時間計算量は $O(n^3)$ となる. □

定理 6 最適リンク構造問題は $O(n^3)$ で計算できる. □

4 議論

4.1 枝分かれ合流のあるリンク構造

以上では seed リンク構造として枝分かれ合流のないものに制限した. もしこの制限がないと次のような問題が生じる.

(1) 条件付き重み $P(D|A)$ の問題.
  A がアクセスされた場合に孫メニュー D がアクセスされる確率が, 途中 B を通るか C を通るかで変わる (図 3(a)).

縮約したときのコスト変化がパスによって変わる．よって縮約するかどうか，パスの数だけ調べないといけない．このときパスの数は一般に指数オーダとなる．

この問題は，アクセス確率の定義 (2) を使えば解消できる．

補題 7 アクセス確率の定義 *(2)* を用いた場合，メニュー M を根に縮約したときのコストの変化は，根から M までのパス上のどのメニューが根に縮約されているかによらない．

ただし，パス上のメニューがすべて縮約されているようなパスが 1 個以上必要である（縮約可能であるための条件）．

(証明) 平均アクセス時間の定義から，M を縮約したときのコストの変化は $|M|$ と $P(M|root)$ にのみ依存． □

(2) エイリアスの問題．
縮約の結果発生した，同じ子メニューへの複数のリンクをエイリアスという（図 3(b)）．
エイリアスを除去する場合，子メニューを縮約したときのコスト変化が，親メニューの縮約状況に依存する．よって補題が成り立たなくなりアルゴリズム OHTLS は正しくなくなる．逆にエイリアスを除去しない場合，指数個のリンクが発生する．

この制限を緩めるため，リンク構造中の枝分かれ合流の数を定数以下に制限するなどの方法が考えられる．定数以下にした場合にはアルゴリズムの計算量は多項式時間に収まる．

4.2 通信コスト

WWW のように，各ページやメニューがネットワーク上に分散しておりそれぞれに対する通信コストにばらつきがある場合がある．この場合，メニューに対するアクセスコストのうち計算機の反応時間 r をメニューによって決まる関数 $r(M)$ とすることが考えられる．すなわち

$$t(M) = c(|M| + 1)/2 + s + r(M)$$

と定義する．

この場合でも補題 3 の証明の $ET[G_{<M_1>}]$ の変化は

$$\frac{c}{2}(|M_2|-1)(1-P(M_2|M_1))-(c+s+r(M_2))P(M_2|M_1)$$

となって M_1 と M_2 によってのみ決まるので補題 3 は成り立つ．よってこれまでの議論がそのまま有効となる．

5 あとがき

本稿では，与えられたリンク構造について，キーワードからアクセスできるページとキーワードに対するカテゴリの対応を保つようなリンク構造すべての中から平均アクセス時間が最小なリンク構造を求める効率のよいアルゴリズムを示した．

なお，本稿の最適リンク構造の定義は，要素が 1 個しかないカテゴリに対しても必ずメニュー頂点をアクセスしてからページにアクセスするような構造を構成する．このようなカテゴリについては，本稿のアルゴリズムで求めた最適リンク構造に対しアンカーからのリンクを置き換えてそのようなメニューを除去することで，より操作しやすいハイパーテキスト構造が構築できる．この操作を行ってもそのアンカーに対するアクセス以外に影響はなく，リンク構造の最適性は失われない．

今後の課題として，ハイパーテキストの別のモデルやより一般的なモデルに対する構成法を考えることがある．

参考文献

[1] Fisher, D. L., Yungkurth, E. J. and Moss, S. M.: "Optimal Menu Hierarchy Design: Syntax and Semantics", *Human Factors*, Vol.32, No.6, pp.665–683 (1990).

[2] ハラリイ, F.: "グラフ理論", 池田貞雄訳, 共立出版, p.15 (1971).

[3] 金子: "ハイパーメディアの研究動向", 情報処理学会誌, Vol.34, No.1, pp.60–71 (1993-01).

[4] Nielsen, J.: "The Art of Navigating through Hypertext", *Communications of the ACM*, Vol.33, No.3, pp.296–310 (March 1990).

[5] 高田, 辻野, 都倉: "最適メニュー階層構造を求めるアルゴリズムについて", 情報処理学会論文誌, Vol.36, No.2, pp.415–422 (1995-02).