

## パラメータ化された連結成分分解を用いた Web 情報の有効利用

正田 備也<sup>†</sup> 高須 淳宏<sup>‡</sup> 安達 淳<sup>‡</sup>

あらまし 本研究では、パラメータ化された連結成分分解を、Web 上の情報の有効利用に役立てる手法を提案する。パラメータ化された連結成分 (PCC: parameterized connected component) への分解は、有向グラフの基本概念である強連結成分の細分化としての Web ページのグループ化を与える。また、閾値パラメータと呼ばれるパラメータを調節することで、グループの粒度を制御できる。本研究では、Web 情報の利用方法としてネットサーフィンと Web 検索とを想定する。前者については PCC 分解の寄与を評価する実験の結果を示し、後者については金澤らによって提案された情報検索システムに PCC 分解としての Web ページのグループ化を組み込む手法を提案する。

キーワード WWW, 強連結成分, Web ページのグループ化, Web 検索, ネットサーフィン

## Efficient Web Information Utilization with Decompositions into Parameterized Connected Components

Tomonari MASADA<sup>†</sup> Atsuhiro TAKASU<sup>‡</sup> Jun ADACHI<sup>‡</sup>

**Abstract** In this paper, we propose a method for applying decompositions into parameterized connected components (PCCs) to efficient Web information utilization. PCC decompositions are subdivisions of decomposition into strongly connected components. Furthermore, by adjusting a parameter called threshold parameter, we can control the granularity of Web page groups. In this paper, we assume the netsurfing and the Web search as two main Web information utilizations. As for the former, we provide the results of evaluation experiment which testifies the contribution of PCC decompositions. As for the Web search, we explain how PCC decompositions can be incorporated into the Web search system proposed by Kanazawa et al.

**Keyword** WWW, Strongly connected components, Web page grouping, Web search, Netsurfing

### 1. はじめに～Web ページのグループ化 の目的

本論文では、パラメータ化された連結成分への分解としての Web ページのグループ化を、Web 上の情報の有効利用に役立てる手法を提案する。パラメータ化された連結成分 (parameterized connected component. 以下 PCC と略す) とは、様々なアプリケーション

に適した粒度で、しかもリンク情報だけを利用して Web ページのグループ化を得るために、筆者らが提案した概念である[1]。WWW は、Web ページを頂点、ハイパーリンクを有向枝と見なすことで、巨大な有向グラフとして抽象化できる。そのため、有向グラフの頂点を、グラフ構造から得られる情報のみを利用してグループ化する手法は、そのまま、

<sup>†</sup> 東京大学情報理工学系研究科 Graduate School of Information Science and Technology, University of Tokyo.

<sup>‡</sup> 国立情報学研究所 National Institute of Informatics.

Web ページをリンク情報のみに基づいてグループ化する手法となる。PCCへの分解としてのグループ化は、このような手法の一つとして、強連結成分分解を一般化することによって提案された。

本論文では、グループ化の利用方法として次の二つを想定する。

- a) ネットサーファーのナビゲーション
- b) テキスト・ベースの Web 検索の性能向上

前者のねらいは、ネットサーファーが短時間に多数のページを閲覧できるようにすることである。そこで、クリックすべきリンクを選択する手がかりとして、グループ化の結果をサーファーに提示する。具体的には、異なるグループ間を結ぶリンクだけを、例えば色を変えるなどして区別して提示する。そして、サーファーがそれを優先的にクリックすると、同一ページに繰り返し戻ってくることの少ないネットサーフィンが実現されるようになります。うまくグループ化を行っておく。この目的a)については、評価実験の結果を第4節で示す。目的b)のねらいは、Web 検索の性能向上である。これについては、本論文では手法の提案にとどめる。具体的には、金澤らの提案する RS モデルに基づく検索システム[2]に PCC 分解の結果をどのように組み込むかを第5節で述べる。評価については他日を期したい。

## 2. 新しい連結性概念～PCC

有向グラフの頂点のグループ化を与えるものとして、強連結成分分解が良く知られている。実際、Broder らは Web のリンク構造がなす有向グラフに対して、強連結成分分解の大規模な実験を行い、成分の大きさの分布を調べてもいる[3]。しかし、彼らの示す結果から、Web グラフ上では強連結成分分解が相当大きなグループを多数与えてしまうことが分かる。さらに最近では、Cooper らが、Web グラフの最大の強連結成分が Web 全体の規模に匹敵することを数学的に示した[4]。このように大きなグループが構成されてしまうと、雑多なページを含むことが予想されるため、Web 検索への応用には不利である。

また、ネットサーファーのナビゲーションにおいても、異なるグループへつながるリンクに出会う頻度が低下するため、やはり不利である。そこで、筆者らは、強連結成分分解の細分化としての粒度可変のグループ化を与える手法を提案した。これが、PCCへの分解としてのグループ化である。

PCC は、以下のように、強連結成分の一般化として提案されている。有向グラフ  $G = (V, E)$  において、頂点  $v \in V$  を含む強連結成分は、その頂点を通過するあらゆる有向閉路を列挙し、それらの上にある頂点をグループ化することで得られる。ただし、有向閉路は、同じ頂点を何度も通過するものも含めて列挙する。PCC は、強連結成分のこの構成法において、列挙されるべき有向閉路の長さを、予め定められたパラメータ  $\tau$  以下に制限することで得られる。このパラメータ  $\tau$  を閾値パラメータ (**threshold parameter**) と呼ぶ。下に PCC 分解のアルゴリズムを示す。

1. 任意に一つの頂点  $v$  を選ぶ。
2. 頂点  $v$  を含み、長さが  $\tau$  以下の有向閉路を列挙する。これらの上にある頂点を、 $v$  と同じ PCC に属するものと定める。
3. すべての頂点がいずれかの PCC に属するまで 1 から 1 を繰り返す。

なお、閾値パラメータが十分に大きい場合、ステップ1で頂点  $v$  を含むすべての閉路が列挙されることになる。このとき、PCC 分解は強連結成分分解に一致する。つまり、強連結成分分解は、PCC 分解の特殊例と言える。また、PCC 分解の時間計算量は、Web ページの総数を  $n$  として  $O(n^2 \log n)$  であることが示せる。アルゴリズムの詳細については、論文[1]を参照されたい。

ところで、有向閉路の長さは、通常、それに含まれる有向枝の数として定義される。しかし、本論文では、有向閉路の長さを、それが通過する頂点の出次数の総和として定義する。例えば、図 1 の有向閉路の場合、通常の意味での長さは 5 であるが、出次数の総和による定義では長さは 18 となる。このように、通常の閉路長を使わない理由は、Web のリンク構造上でこの定義に基づいて PCC 分解を行うと、きめ細かなグループ化が実現されなかった点にある。実際、通常の定義で

は同じ距離とみなされる閉路のなかに、新しい定義では長短の微妙なグラデーションが導入される。これによって、よりきめの細かいグループ化を実現できる。また、出次数の和を閉路の長さと定めることで、出次数の多いWebページが同じグループに属しにくくなる。よって、局所的なディレクトリ構造、つまり、出次数の多い親ページに適度な数の子ページがぶら下がる構造を抽出できるという利点もある。このような局所構造には、意味上のまとまりも期待できるため、Web検索への応用に際しても有利となることが予想できる。

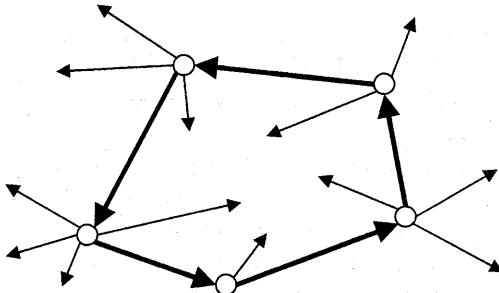


図1. 有向閉路の例。

### 3. PCC 分解の性質

本節では、クローリングによって得たWebページ集合についてPCC分解を行った結果を示す。これによって、PCC分解の性質を明らかにする。

まず、Webページの数や閾値パラメータと、PCC分解に要する時間との関係をみるために、500,000件および1,000,000件のページからなる集合について閾値パラメータ $\tau$ が5, 10, 20, 50, 100, 150, 200の場合のPCC分解を行い、実行時間を測定した結果を図2に示す。この実験は、750MHzと900MHzの二つのUltra SPARC IIIを搭載し、主記憶容量が8GByteのSun Blade 1000上で行った。OSはSolaris 8である。閾値パラメータが大きいほど実行時間は少ない。これは、閾値パラメータの増大

に伴ってPCC分解のアルゴリズムのあるましいが強連結成分分解に近づくためである。なお、強連結成分分解の時間計算量は、Webページ数およびハイパーリンク数の線形オーダーである[5]。

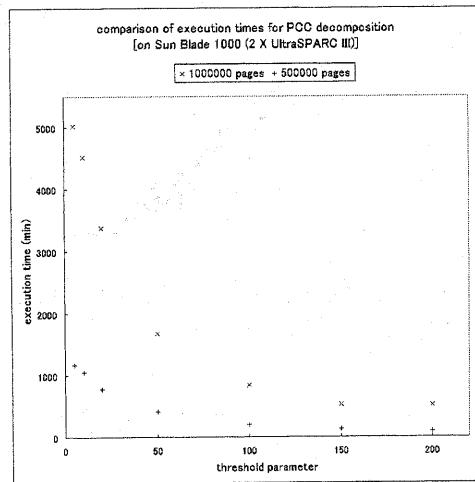


図2. PCC分解の実行時間。Webページの総数が500,000の場合と1,000,000の場合とのデータを含む。

次に、より大規模な5,000,000件のWebページについてPCC分解を行った場合のPCCのサイズの分布を図3に示す。この実験では、Xeon 2.6GHzプロセッサを二つ搭載し、主記憶容量が2GByteのPCを8台使った。OSはSolaris 8である。PCC分解のアルゴリズムは、マルチキャストを利用して並列化した。閾値パラメータは20, 50, 100, 200と変化させた。それぞれの場合の実行時間は7380, 3120, 1650, 1220分だった。横軸がPCCのサイズ、縦軸がPCCの個数である。閾値パラメータが小さいほど、サイズの小さなPCCが増え、サイズの大きなPCCが減っている。閾値パラメータによってグループの粒度が制御されていると分かる。

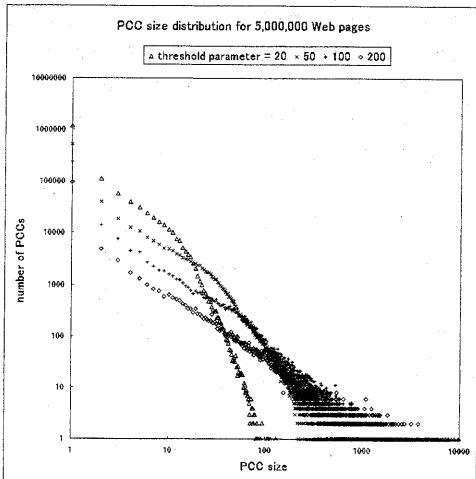


図3. 5,000,000のWebページについてPCC分解を行った場合のPCCのサイズの分布。

#### 4. ネットサーファーのナビゲーション

PCC分解がネットサーファーのナビゲーションにどのように寄与しうるかを調べるために、5,000,000件のWebページの集合について求めたPCC分解に対して評価実験を行った。本節ではその結果を示す。

ネットサーフィンは、ページ上のリンクのうち、未訪問のページへつながるリンクから等確率でクリックすべきリンクを選ぶランダムウォークによって、シミュレートした。グループ化の効果を調べるために、次の二種類のランダムウォークを比較した。

- ・ガイドなしランダムウォーク 未訪問のページへつながるリンクから、クリックすべきリンクを等確率でランダムに選ぶ。
- ・ガイドありランダムウォーク 未訪問のPCCへつながるリンクがあれば、それらの中からクリックすべきリンクを等確率でランダムに選ぶ。そのようなリンクがなければ、未訪問のページへつながるリンクからクリックすべきリンクを等確率でランダムに選ぶ。

いずれのランダムウォークも、訪問済みのページへつながるリンクしかないページに行き着くまで、続けられる。よって、ランダムウォーク中に訪れるページはすべて相異なる

ページであり、その数はランダムウォークの長さに一致する。なお、ランダムウォークの長さは、訪問したページ数と定義する。評価実験の手続きの詳細は以下のとおりである。この実験手続きが $k$ というパラメータを持っていることに留意されたい。

1. ランダムウォークの出発点となるWebページを5,000,000件のページからランダムに選ぶ。
2. ステップ1で選ばれたページから、ガイドありランダムウォークを行う。
3. ステップ2と同じページを出発点として、ガイドなしランダムウォークを行う。
4. ガイドなしランダムウォークが、少なくとも $k$ 個の相異なるPCCを訪問できなかつた場合、ステップ1から3までをやり直す。
5. ガイドありランダムウォークの長さから、ガイドなしランダムウォークの長さを引き算し、その結果を出力する。
6. 合計20,000個の結果が出力されるまでステップ1から4を繰り返す。

図4は、ガイドありランダムウォークと、ガイドなしランダムウォークとの長さの差について、合計20,000個のデータを得るために、ステップ1から3までの試行が何回必要だったかを示している。

|       | $\tau=20$ | $\tau=50$ | $\tau=100$ | $\tau=200$ |
|-------|-----------|-----------|------------|------------|
| $k=2$ | 23342     | 28106     | 36191      | 50736      |
| $k=4$ | 36355     | 47837     | 70516      | 110389     |
| $k=6$ | 51744     | 72264     | 112050     | 182533     |

図4. 二種類のランダムウォークの長さの差について20,000件のデータを得るために必要とされた試行の数。 $\tau$ は閾値パラメータ。

パラメータ $k$ が増えるほど、必要な試行の数は増えている。これは、多くの相異なるPCCを訪問するランダムウォークほど、その生起の頻度が低いからである。また、閾値パラメータが増えて、やはり必要な試行の数が増えている。これは、閾値パラメータが増えると、大きいPCCの個数が増え、異なるPCCへつながるリンクに出会う頻度が低下するためである。

図5は、 $k=2$ の場合の二種類のランダムウォークの長さの差の分布である。データの総数は20,000件である。 $k=2$ より、ガイドありランダムウォークが異なる二つのPCCさえ訪問すれば、成功した試行として認められる。よって、ガイドなしのランダムウォークとあまり大きな差がない。実際、どの閾値パラメータの値についても、差がゼロのケースが最も多い。

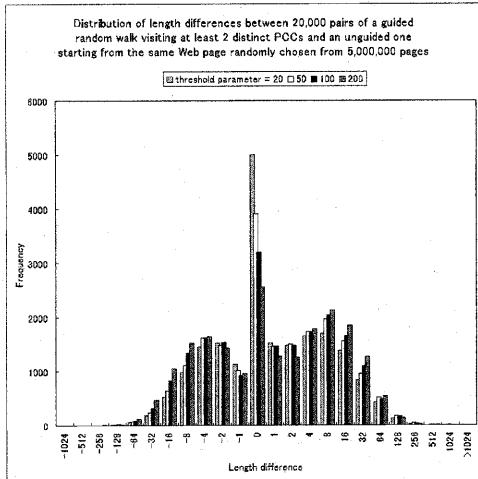


図5. 少なくとも二つのPCCを訪問したガイドありランダムウォークが、ガイドなしランダムウォークに比べて、どれだけ多くのWebページを訪問できたかを示すグラフ

図6は、 $k=4$ の場合の二種類のランダムウォークの長さの差の分布である。相異なる四つのPCCを訪問してはじめて、ガイドありランダムウォークが成功したと認められるため、 $k=2$ の場合よりは、ガイドなしのものとの差が大きく出ている。つまり、明らかに正の側に分布が偏っている。しかし、閾値パラメータが20程度であると、差がゼロの場合もかなり多く、はっきり差が出ているとは言いがたい。その一方で、閾値パラメータがあまり大きくても、確かに正の側の頻度も増えるが、負の側の頻度も同時に増えているため、より良いナビゲーションが実現できるとは言えない。

図7は、 $k=6$ の場合の差の分布である。ガイドありランダムウォークの優位がはっきりと確認できる。しかし、図4が示していた

ように、多数の異なるPCCを訪問するランダムウォークはそもそも実現されにくく、ランダムウォークの質と生起頻度との間に、トレード・オフのあることが分かる。

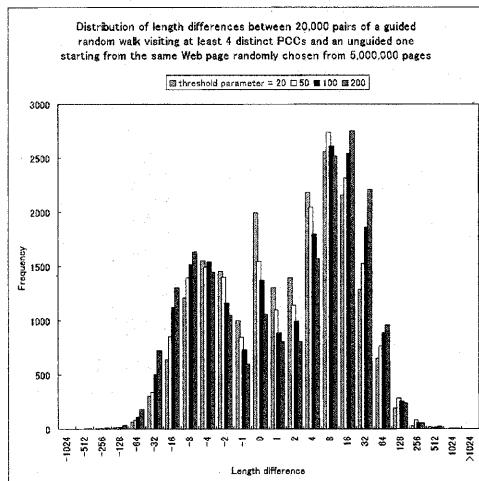


図6. 少なくとも四つのPCCを訪問したガイドありランダムウォークが、ガイドなしランダムウォークに比べて、どれだけ多くのWebページを訪問できたかを示すグラフ。

図8は、以上のデータを使って、二種類のランダムウォークの長さの差の平均についてZ検定を行なった結果である。有意水準は5%とした。図9には各々のケースについてZ検定に用いた分散の値を示した。どの $k$ についても、閾値パラメータが50の場合に最も大きな長さの差を得た。例えば $k=4$ のとき、ガイドありランダムウォークのほうが、ガイドなしのものに比べて、平均で5.5個多くのページを訪問できると、有意水準5%で言える。閾値パラメータが大きすぎても、小さすぎてもいけない理由は、以下のとおりである。閾値パラメータが大きすぎると、サイズの大きなグループ、つまり、多様なページを含み込むグループの個数が増え、異なるPCCをいくつ訪問したかを問わないガイドなしランダムウォークであっても、多様なページを訪問できるようになる。逆に、閾値パラメータが小さすぎると、異なるグループにつながるリンクに出会う頻度が増え、ガイドなしランダムウォークであっても、多数の相異なるPCCを訪問できるようになる。以上の理由に

よって、閾値パラメータが大きすぎても、小さすぎても、ガイドなしランダムウォークとの差が現れにくくなる。

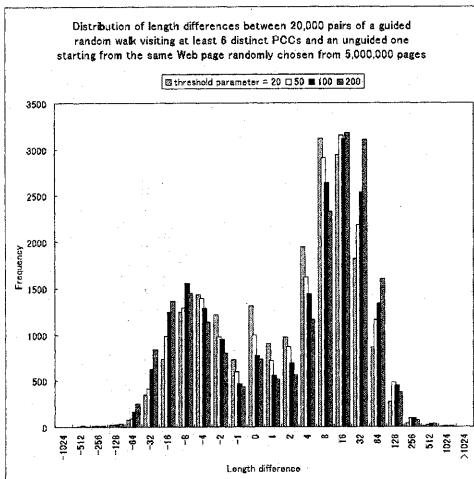


図7. 少なくとも六つのPCCを訪問したガイドありランダムウォークが、ガイドなしランダムウォークに比べて、どれだけ多くのWebページを訪問できたかを示すグラフ。

|       | $\tau = 20$ | $\tau = 50$ | $\tau = 100$ | $\tau = 200$ |
|-------|-------------|-------------|--------------|--------------|
| $k=2$ | 2.4         | 3.4         | 2.7          | 2.0          |
| $k=4$ | 4.2         | 5.5         | 5.0          | 4.5          |
| $k=6$ | 6.0         | 8.4         | 7.9          | 8.1          |

図8. 二種類のランダムウォークの長さの差の平均についてのZ検定の結果。有意水準は5%とした。 $\tau$ は閾値パラメータを表す。例えば、 $k=6$ 、 $\tau=100$ のときの7.9という値は、この設定の下でのガイドありランダムウォークが、ガイドなしランダムウォークよりも、20,000個のデータについてとった平均値で、7.9だけ長いと、有意水準5%で言えることを意味する。

## 5. テキスト・ベースのWeb検索への応用

本節では、PCC分解としてのWebページのグループ化を、テキスト・ベースのWeb検索システムに組み込む手法を提案する。

|       | $\tau = 20$ | $\tau = 50$ | $\tau = 100$ | $\tau = 200$ |
|-------|-------------|-------------|--------------|--------------|
| $k=2$ | 441.3       | 688.9       | 840.5        | 1057.6       |
|       | 462.9       | 1084.8      | 1060.7       | 1135.9       |
| $k=4$ | 638.4       | 977.2       | 1496.8       | 1925.5       |
|       | 757.7       | 1332.6      | 1777.9       | 2143.9       |
| $k=6$ | 1370.3      | 1564.2      | 1917.3       | 2877.2       |
|       | 1343.5      | 1967.2      | 2322.1       | 3401.3       |

図9. 図8のZ検定に用いた、二種類のランダムウォークの長さの差の分散。 $\tau$ は閾値パラメータを表す。

本論文では、ベクトルモデル[6]に基づく情報検索システムを想定する。ベクトルモデルを利用した検索システムにおいては、各々の文書に文書ベクトル(document vector)と呼ばれる高次元のベクトルが対応づけられる。文書ベクトルは、索引語の数だけの次元を持ち、TF-IDFをエントリとする。検索者が与えた質問にも質問ベクトル(query vector)と呼ばれるベクトルが対応づけられ、質問と各文書との類似度が、ベクトルとしての類似度によって評価される。ベクトルとしての類似度は内積やコサインによって求められる。

Webページのグループ化は、文書ベクトルのグループ化を引き起こす。そこで、金澤らの提案するRSモデル[2]を利用して、PCC分解としてのグループ化の影響を、以下のようにして文書ベクトルの各エントリに反映させる。まず、各文書グループについて代表ベクトル(representative vector)と呼ばれるベクトルを計算する。第*j*番目の文書 $d_j$ に対応する文書ベクトルの、第*i*番目の検索語 $t_i$ に対応するエントリを $v_{j,i}$ とする。このとき、文書グループ*C*の代表ベクトルの、検索語 $t_i$ に対応するエントリは $r_i = \left( \frac{1}{|C|} \sum_{d_j \in C} \sqrt{v_{j,i}} \right)^2$ という式で算出される。そして、当のグループに属する各ベクトルを、上式で求められたエントリをもつ代表ベクトルに向かって微小量だけ移動させる(図10)。

こうして変更された文書ベクトルを、元の

文書ベクトルのかわりに用いれば、検索者が与えた質問との類似度の評価にグループ化の結果を反映させることができる。仮に、各々のグループが意味的にも纏まっているれば、上記の方法による文書ベクトルの変更は、検索性能の向上をもたらすと予想される。よって、PCC 分解としてのグループ化が、Web 検索の性能向上に寄与する種類のグループ化か否かを調べることができる。

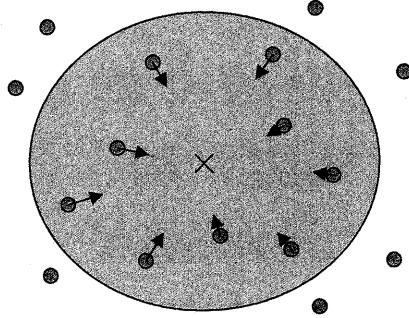


図10. PCC 分解のテキスト検索への利用方法。濃いグレーの丸印は文書ベクトルを表す。薄いグレーで示された領域は、文書のグループを表す。短い矢印は、同じグループに属するベクトルが、 $\times$ 印で表された代表ベクトルへ向かって移動させられている様を示す。

## 6.まとめ～今後の課題

本論文では、第1節で Web ページをグループ化する目的について述べ、筆者らが既に提案した PCC 分解としてのグループ化を第2節で簡潔に紹介した。その性質は、第0節に示したとおりであり、粒度の制御を伴ったグループ化を、妥当な計算時間内で実現できている。そして、第4節ではネットサーファーのナビゲーションにおいて、どのようなパラメータ設定の下で PCC 分解が良いパフォーマンスを発揮するのかに関する調査結果を示した。

現在、第5節に記した手法で PCC 分解の Web 検索への寄与を評価するための準備を行っている。予定としては、第4節で示した実験で用いたものよりも大規模な 10,000,000 件の Web ページ集合を用意し、PCC 分解と、

ベクトルモデルに基づく検索処理のための文書ベクトルの計算とを行う。実験環境は、Xeon 2.8GHz を二つ搭載した 13 台の PC で構成されている。OS は RedHat Linux である。10,000,000 件の Web ページに対する PCC 分解はすでに何度か試みており、閾値パラメータが 50 の場合に 23.5 時間、また 20 の場合に 47 時間を要することが分かっている。今後は、代表ベクトルの生成の部分と、グループ化の結果を利用した文書ベクトルの変更の部分との実装に着手する。本システムにおける処理の流れを図11に示す。PCC 分解は、テキスト情報を利用しないため、文書ベクトルを作成する処理と同時に行うことができる点に注意されたい。文書ベクトルの作成は、形態素解析や検索語の抽出など、コストのかかる処理を必要とする。よって、こうしたテキスト処理と並行して Web ページをグループ化できるという事実は、システムにスケーラビリティを確保するという観点からは重要である。

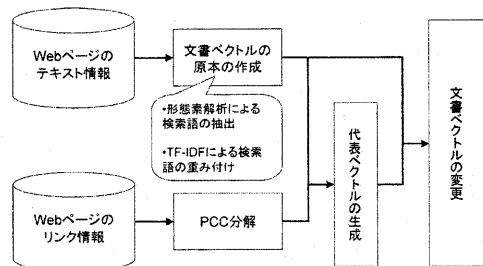


図11. PCC 分解を利用した Web 検索システムにおける処理の流れ。

## 謝辞

本研究は、文部科学省科学研究費補助金特定領域研究「情報学」(課題番号 13224087) の助成のもとに行われています。また、本研究での計算機実験は、国立情報学研究所の大山敬三教授のご協力なしには実現できませんでした。この場を借りて感謝の意を表します。

## 文 献

- [1] 正田備也, 高須淳宏, 安達淳, “新しい連結性概念と Web ページのグループ化への応用,”日本データベース学会 Letters, vol.2, no.1, pp.3-6, May. 2003.
- [2] T. Kanazawa, A. Aizawa, A. Takasu, and J. Adachi, “The Effects of the Relevance-Based Superimposition Model in Cross-Language Information Retrieval,” ECDL 2001, LNCS 2163, pp.312-324, Darmstadt, Sep. 2001.
- [3] A. Z. Broder, S. R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, “Graph Structure in the Web,” Proc. 9th WWW Conf., pp.309-320, Amsterdam, May. 2000.
- [4] C. Cooper and A. Frieze, “The Size of the Largest Strongly Connected Component of a Random Digraph with a Given Degree Sequence,” pre-print, available at <http://www.math.cmu.edu/~af1p/papers.html>, Mar. 2002.
- [5] E. Nuutila and E. Soisalon-Soininen, “On Finding the Strongly Connected Components in a Directed Graph,” Information Processing Letters, vol.49, issue 1, pp.9-14, Jan. 1994.
- [6] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, pp.27-30, ACM Press, New York, 1998.