2004－AL－95　(8)
2004／5／21

サイト間グラフの最小カットを用いたウェブ上のコミュニティ発見法

浅野泰仁* 吉田雄介* 西関隆夫* 豊田正史† 喜連川優†

概要

近年, ウェブのハイパーリンク構造を利用した情報発見手法の研究が進められているが, これに対してサイトを用いた新しい枠組みが提案され, trawling などを用いてこの枠組みの有効性が示されている. 一方, ユーザーによって指定されたページに関連するページの集合, すなわちコミュニティ, を求める手法として, Flake らによって最小カット法が提案されている. しかし, この手法はリンクのグラフ構造に対して最小カットを見つけて, コミュニティを求めており, サイトを用いた枠組みが最小カット法にどれだけ有用であるかはわかっていない. 本論文では, 浅野らによって提案されたサイト識別の方法と豊田, 喜連川らによって収集, 作成された 2003 年 2 月の日本語ページからなる URL とリンクのデータを用いて, 既存のページを単位とした枠組みとサイトを用いた枠組み双方で最小カット法によって様々なトピックに関してコミュニティを得る実験をすることによって, 双方の性能を評価した. 結果として, サイトを用いた枠組みを用いた最小カット法は既存の枠組みと比べて精度をほとんど落とさずに, 平均 2.5 倍程度の大きさのコミュニティを得られることがわかった.

Finding Communities on the Web by a Min-Cut of an Inter-Site Graph

Yasuhito ASANO* Yusuke YOSHIDA* Takao NISHIZEKI*
Masashi TOYODA† Masaru KITSUREGAWA†

Abstract

Information retrieval methods from the Web using hyperlinks have been widely researched. Very recently, Asano et al. have proposed a new site-oriented framework for these methods and verified usefulness of this framework for information retrieval from the Web by using several methods including a trawling method. Flake et al. have proposed a min-cut based method of finding communities (a set of pages related to pages given by a user). While this method finds a community by using a min-cut on a graph structure of links, it is not known how the site-oriented framework is useful for this method. In this paper, we compare the min-cut method using the site-oriented framework with the existing page-oriented framework by computational experiments for several topics, by using a method of identifying sites proposed by Asano et al. and data of URLs and links collected in February 2003 by Toyoda and Kitsuregawa. As a result, the min-cut method using the site-oriented framework has obtained communities with size 2.5 times on average as large as the min-cut method using the existing page-oriented framework, without decreasing the precision much.

*東北大学大学院情報科学研究科, 980-8579 仙台市青葉区荒巻字青葉 05
†東京大学生産技術研究所, 153-8505 東京都目黒区駒場 4-6-1

1 はじめに

近年，急速に発展しているウェブ (World Wide Web) は, 情報を得る手段として日常に欠かせないものとなっている．ウェブから必要な情報を発見する手段としては, Google, Yahoo!, Goo, Infoseek などの検索エンジンが代表的なものである．これらの検索エンジンは, ユーザーの入力キーワードを含んでいるページを見つけるテキストに基づいた手法を基本として用いている．

これに対して，ウェブのハイパーリンク構造を利用した情報発見手法の研究も進められている．このリンクを用いた手法は, テキストを用いた手法のように同綴異義語やダミーキーワードなどの言語的問題に惑わされないという長所がある．リンクを情報発見に用いるための基本的な考えは, あるページ u からリンクされているページ v は, u の作者にとって価値のある情報を含んでいる可能性が高いという仮定であり, これらの研究のほとんどは，以下のような枠組みに基づいている．(1) ウェブ中のページ群の URL とリンクのデータを収集する．(2) このデータからウェブグラフを作成する．(3) ウェブグラフ中の特徴的な部分グラフ構造を互いに関連しているページまたはサイトの集合である**コミュニティ**として抽出する．なお, 点集合 V がウェブページの集合を表し, 辺集合 E が V 中のページ間のリンクを表す有向グラフ $G = (V, E)$ を, ウェブグラフと呼ぶ.

リンク構造を用いてコミュニティを求める手法の研究としてよく知られたものに，Kleinberg の提案した HITS([1])，Kumar らの trawling ([2]), Flake らによる最小カット法 ([3]) などがある．とくに, 最小カット法はユーザーが与えたページ群 (以下では**シードページ群**と呼ぶ) に関連するコミュニティを求めることができる手法である．

一方, 近年浅野らがリンク構造を用いてコミュニティを求める手法のための, サイトを用いた新しい枠組みを提案し, trawling などを用いてこの枠組みの有効性を示した ([4], [5]). この枠組みは, 次のようになっている．(1) ウェブ中のページ群の URL とリンクのデータを収集する．(2) このデータから**フィルタ**手法を用いて**ディレクトリベースドサイト**を抽出する．(3) ディレクトリベースドサイトを点とし, ディレクトリベースドサイト間のリンクを辺とする**サイト間グラフ**を作成する．(4) サイト間グラフ中の特徴的な部分グラフ構造を**コミュニティ**として抽出する．

すなわち, 従来の枠組みではページを単位として用いていたのに対して, この枠組みではサイト (正確にはその近似的なモデルであるディレクトリベースドサイト) を単位として用いているといえる．これは, あるサイト A から他のサイト B へのリンクが, そのサイト A の作者にとって価値のある情報への参照を表している可能性が高いのに対して, サイト内のリンクはサイト内のページを見やすく整理するというナビゲーション目的で作られていることが多いため, リンクを用いた情報発見には両者を区別して用いることが重要と考えられるからである．

もっとも, サイトという言葉は日常的に用いられているが，それ自体曖昧な概念であり，唯一の定義もない．たとえ, いくつかの研究で用いられている「ひとりの個人，ひとつの企業，ひとつの集団がつくるページの集合をそれぞれサイトとする」という定義を用いたとしても, ウェブのページには作者の情報は一般的には含まれていないので，サイトを計算することは困難であると考えられてきた．

これに対して，近年，浅野ら ([4], [5],[7]) によって，ある時点での URL とリンクのデータ (スナップショットと呼ぶ) からこの**ディレクトリベースドサイト**というモデルと**フィルタ**という手法を用いて，サイトを近似的ではあるが実用的には十分な精度で識別する方法が発表された．さらに, これらの研究では, trawling と彼らの提案した極大クリーク法を用いて, サイトを用いた枠組みがコミュニティ発見に役立つこと, とくに個人サイトを含むコミュニティを多く見つけることに大きな効果があることを示している．だが, リンクを用いた有力なコミュニティ発見法である最小カット法に関しては, この枠組みがどのような効果を持つかはわかっていない.

本論文では，このサイト識別の方法と豊田，喜連川らによって収集・作成された 2003 年度 2 月の .jp ドメインを主体とした日本語ページからなるスナップショットを用いて，実際にサイトのデータとサイト間グラフのデータを構築し，最小カット法をサイト間グラフとウェブグラフ両方の上で動かし，双方の性能を評価する．

結果として，サイト間グラフ上の方が 2.5 倍

程度大きなコミュニティが得られ，精度はサイト間グラフもウェブグラフもさほど差はなかった．このことから, 最小カット法にもサイトを用いた枠組みが実用上有効であるということがわかった．

本論文は 6 節からなる．2 節ではサイトを単位とした枠組みとその実現方法, すなわちディレクトリベースドサイトとフィルタについて説明する．3 節では, Flake らが提案した最小カット法について説明する．4 節では, 最小カット法でページを単位とした枠組みを用いる際の問題点について説明し, 5 節では，ページを点とするウェブグラフとサイトを点とするサイト間グラフに対しそれぞれ最小カット法を用いた実験結果を示し，ふたつの実験結果を比較し評価している．6 節は, まとめである．

2 サイトを用いた枠組みの実現

2.1 ディレクトリベースドサイト

定義 1 あるウェブサーバーに対して, $\{d_1, ..., d_k\}$ を与えられたディレクトリの集合とする．ただし, 各 d_i $(1 \leq i \leq k)$ はルートディレクトリまたは他の d_j $(j \neq i)$ のサブ (子孫含む) ディレクトリではないものとする．このとき, 各 i について, d_i をトップディレクトリとするディレクトリベースドサイト D_i を, d_i とそのすべてのサブディレクトリに含まれるページの集合と定義する．また, このサーバー内の $\{d_1, ..., d_k\}$ (およびそれらのサブディレクトリ群) に含まれていないページの集合を, サーバーの管理者のディレクトリベースドサイトと呼ぶ．便宜上, サーバーの管理者のディレクトリベースドサイト以外のディレクトリベースドサイトを, ユーザーのディレクトリベースドサイトと呼ぶ．

もし, 与えられたサーバーの全てのページがサーバーの管理者のディレクトリベースドサイトに属しているならば, (i.e. 定義 1 で $k = 0$), このサーバーをシングルサイトサーバーと呼ぶ．そうでないならば, (i.e. 定義 1 で $k \geq 1$ かつ少なくともひとつのディレクトリが与えられている), このサーバーをマルチサイトサーバーと呼ぶ．

図 1 はあるウェブサーバー内のディレクトリ $\{d_1, d_2, d_3\}$ が与えられたときのこのサーバー中

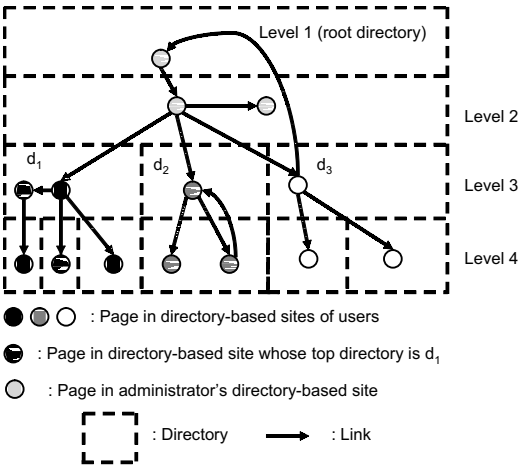

図 1: マルチサイトサーバーのディレクトリベースドサイトの例．

のディレクトリベースドサイトを表している．この図では, 黒, ダークグレー, 白の円がそれぞれ d_1, d_2, d_3 をトップディレクトリとするディレクトリベースドサイト内のページを表し, ライトグレーの円はサーバーの管理者のディレクトリベースドサイト内のページを表している．

定義 1 において, あるサーバーが与えられたときディレクトリの集合 $\{d_1, ..., d_k\}$ も与えられないといけない．したがって, 現実のユーザーのサイトにできるだけ一致するようなディレクトリの集合 $\{d_1, ..., d_k\}$ を, そのサーバーに関する何らかのデータから見つけ出す手法を構築しなくてはならない．この手法については, 次に説明する．

2.2 フィルタ

ディレクトリベースドサイトを URL とリンクのデータから発見する方法の概要について説明する．

入力はウェブサーバーの集合 S とする．与えられたサーバー $s \in S$ に対して, $U(s)$, $L(s)$ をそれぞれ s 内にある URL の集合, リンクの集合とする．この手法は, フィルタリングフェイズとエラー訂正フェイズ (error correction of filters using cliques, ECFC と書く) からなっている．

フィルタリングフェイズでは, 7 つのフィルタがある．それぞれ Filter i $(0 \leq i \leq 6)$ と表すことにする．各 Filter i は, 与えられたサーバーの集合から下に記述する条件を満たすサーバーを識別し, 残りのサーバーを次のフィルタに渡す．

S_i を Filter i の入力とし, I_i を Filter i で識別されたサーバーの集合とする. すなわち, $S_0 = S$ であり, $S_i = S_{i-1} \setminus I_{i-1}$ となる. ECFC は I_5 and I_6 中の, 誤って識別されたサーバーを発見しなおして, 誤りを正す. 最終的に残ったサーバー, すなわち $F = S_6 \setminus I_6$ 内のサーバーはシングルサイトサーバーと見なされる.

Filter 0: このフィルタでは, 各レンタルウェブサーバーや ISP(Internet Service Provider) のサーバーのユーザーのサイトに対応するディレクトリの階層に関する知識データベースを用いる. なお, 第 1 階層がルートディレクトリに対応し, 第 j 階層は第 $j-1$ 階層のディレクトリ群の子ディレクトリ群に対応する. データベースの各要素は, 文字列と整数のペアからなる. 例えば (u-page.so-net.ne.jp, 3), (geocities.co.jp, 3), (highway.ne.jp, 2) のようになっている.

このフィルタは, 名前の接尾辞が, データベース中のある要素の文字列にマッチしているようなサーバーをすべて見つけ, それらのサーバーを, その要素の整数と等しい階層にあるディレクトリ群をユーザーのディレクトリベースドサイトのトップディレクトリ群とするマルチサイトサーバーであると識別する. すなわち, その整数を x とするならば, そのサーバー内の URL 群を x 番目のスラッシュ記号で区切り, ユーザーのディレクトリベースドサイトの名前とみなす. x 個のスラッシュ記号を含んでいない URL に対応するページは, 管理者のディレクトリベースドサイト内のページとみなす.

例えば, `www05.u-page.so-net.ne.jp/` というサーバーは上記の 1 つの要素にマッチするので, このサーバー内のページの URL `www05.u-page.so-net.ne.jp/jd5/niwachan/top.html` は 3 番目のスラッシュ記号で分けられ, `www05.u-page.so-net.ne.jp/jd5/niwachan/` がディレクトリベースドサイトの名前となり, このページはこのサイト内の `top.htm` という名前に対応することになる. この知識データベースの要素数はわずか 17 個ですんでいる.

Filter 1: このフィルタは, その中の URL 集合の少なくともひとつが, チルダ記号 (ただしスラッシュ記号の直後のもの) を含んでいるようなサーバーをマルチサイトサーバーとして識別する. そのような URL に関しては, 最初に現れた/ の次のスラッシュ記号までをユーザーのディレクトリベースドサイトの名前とみなす. 例えば, URL `www.mars.dti.ne.jp/~tk491114/pictures/` があった場合, サーバー `www.mars.dti.ne.jp/` はマルチサイトサーバーとして識別され, このページを含むサイトの名前は `www.mars.dti.ne.jp/~tk491114/` となる.

Filter 2: このフィルタでは, 有名な会社や組織のサーバーに関する知識データベースを用いる. データベースの各要素は, (sony.co.jp) や (yomiuri.co.jp) のように, 文字列からなる. このフィルタは, 名前の接尾辞が, データベース中のある要素の文字列にマッチしているようなサーバーをすべて見つけ, シングルサイトサーバーとして識別する. この知識データベースの要素数は 735 である.

Filter 3: このフィルタは, サーバー内のすべてのページがひとつのディレクトリ内にあるようなサーバーをシングルサイトサーバーとして識別する.

Filter 4: このフィルタは, ページ数が与えられたパラメーター c 以下であるようなサーバーをシングルサイトサーバーとして識別する. 文献 [5] では, このフィルタの誤りを最小化するために, いくつかの c の値を試し, 結果として $c = 20$ を採用している.

Filter 5: このフィルタは, 各サーバー $s \in S_5$ のすべてのページと $L(s)$ から誘導される有向グラフ $G(s)$ をまず作り, これを連結成分分解する. 複数の連結成分が存在した場合, それらを $C_1, C_2, ..., C_k$ で表し, $d(C_i)$ を $\min\{level(x) \mid x$ は C_i に含まれるページ $\}$ とする. ただし $level(x)$ は x の URL 中のスラッシュ記号の数とする.

$C_1, ..., C_k$ は $d(C_i)$ に関する昇順で並べられているものとする. . もし $d(C_1) = d(C_2) = \cdots = d(C_k) \geq 2$ であるか $\exists i, d(C_1) = \cdots = d(C_i) < d(C_{i+1}) = \cdots = d(C_k)$ であるならば, このフィルタは s をマルチサイトサーバーであると識別し, 第 $d(C_k)$ 階層のディレクトリをユーザーのディレクトリベースドサイトのトップディレクトリであるとみなす.

Filter 6: サーバー s の第 j 階層にあるディレクトリの集合を $D_j(s)$ で表すことにする. このフィルタは, 各サーバー s について, $|D_j(s)|/|D_{j-1}(s)|$

が与えられたパラメータ以上である (i.e. 第 j 階層にあるディレクトリの数がそれらの親ディレクトリの数よりずっと大きくなる) ような階層 $j \geq 2$ があれば, それを見つける. そのような j が見つかり, s が次に述べる条件をすべて満たすならば, このフィルタは s をマルチサイトサーバーとみなし, 第 j 階層をユーザーのディレクトリベースドサイトのトップディレクトリ群の階層とする.

条件 1: 第 $h(h \geq j)$ 階層のディレクトリにあるページから, 第 k $(k < j)$ 階層のディレクトリのページへのリンクの数が, 与えられたパラメータ以下である. 条件 2: 第 $j-1$ 階層のディレクトリにあるページから, 第 j 階層のディレクトリのページへのリンクの数が, 与えられたパラメータ以下である. 条件 3: 他のサーバーのページから第 j 階層のディレクトリのページへのリンクの数が, 与えられたパラメータ以上である.

もし, 同一のサーバーに関して複数の階層がこれらの条件をすべて満たすならば, 上記のパラメータに基づいた点数付けをすることによって, 1 つの階層だけを選ぶ. これらのパラメータは s のページ数によって複雑に変化するため, 詳細に関しては [5] を参照されたい.

ECFC: I_5 または I_6 に含まれる各サーバーに s 対して, ECFC は Filters 5 または 6 で発見されたディレクトリベースドサイトを点とし, それらのサイト間のリンクを辺とする有向グラフを作る. このグラフ中に, 辺 (u, v) と (v, u) がともに存在するような 2 点 u, v が存在した場合, ECFC はサーバー s をシングルサイトサーバーと識別し直す. 便宜上, I_e を ECFC で識別されたサーバーの集合とする.

この手法の誤り率は以下のように評価される. この手法が与えられたあるサーバー s をシングルサイトサーバーと識別した場合, s 中のページを見てすべてが管理者によって作成されたと考えられるなら, これは正しいと判断する. また, s をマルチサイトサーバーと識別し, d をユーザーのディレクトリベースドサイトのひとつとした場合, d 中のページ v とその親ディレクトリ中のページ w を見て, それらが異なる作者によって作成されていると考えられる場合, これは正しいと判断する.

表 1: 2003 年のスナップショットを用いて識別されたサーバーの数とディレクトリベースドサイトの数.

Filter	Remains	Identified	Errors	Sites
0	2,947,658	16,898	3	1,191,696
1	2,930,760	439,188	15	1,488,401
2	2,491,572	36,520	1	36,520
3	2,455,052	1,209,431	0	1,209,431
4	1,245,621	810,666	0	810,666
5	434,955	96,533	40	722,179
6	338,422	3,331	74	409,289
ECFC	335,091	(20,283)	4	-241,378
F.R.	335,091		9	335,091
Total				5,961,895

表 1 は, 2003 年のスナップショットにこの手法を適用した結果を記している. Filters 0 から 6 と ECFC で識別されたサーバーの数, 誤りの数, 得られたディレクトリベースドサイトの数を記した (F. R. は, Final Remains の略である).

“Remains” 列の Filter i $(0 \leq i \leq 6)$ に対応する各行では $|S_i|$, ‘ECFC” 行と “Final Remains” 行では $|S_6 \setminus I_6|$ の値を表してある. “Identified” 列の Filter i $(0 \leq i \leq 6)$ に対応する各行では $|I_i|$, ‘ECFC” 行では $|I_e|$ の値を表している ($I_e \subset I_5 \cup I_6$ であるから, $|I_e|$ はカッコ中の数字で表している).

“Errors” 列は各々I_i, I_e (“ECFC” 行), F (“Final Remains” 行) 中からランダムにサンプリングされた 150 個のサーバー集合のうち, この手法の結果が誤っていたものの数を表している. なお, これらの誤りの数をそれぞれ e_i $(0 \leq i \leq 6)$, e_e, e_f で表すことにする. F 中のサーバーはシングルサイトサーバーとして識別されたことになるので, e_f はサンプル中のマルチサイトサーバーの数となる.

“Sites” 列はそれぞれ I_i, I_e, F から見つかったディレクトリベースドサイトの数を表している. ECFC は Filters 5 and 6 で誤って識別されたサーバーを見つけ, そのようなサーバー $s \in I_e$ 内のディレクトリベースドサイトを削除して, そのサーバーをシングルサイトサーバーとして識別し直す. よって, “ECFC” 行の数値は, 負の値となっている.

“Total” 行はこの手法で見つかったディレクトリベースドサイトの数の総和を表している. 結果として, この手法で 2,947,658 個のサーバーか

表 2: ウェブグラフおよびサイト間グラフの規模

グラフ	点の数	辺の数
ウェブグラフ	65,746,408	1,032,820,388
サイト間グラフ	5,961,895	72,540,379

ら 5,961,895 個のディレクトリベースドサイトを発見した.

この手法の誤りの総数は次の式でおおよそ評価できる. $\sum_{0\leq i\leq 6} e_i \times |I_i|/150 - |I_e| \times (1 - e_e/150) + e_f \times |F|/150$.

この式で評価した誤り率は, わずか 3.0% であることから, この手法はほとんど正しくすべてのサーバーを識別していると言える. 誤りの原因の解析などに関しては, [5] を参照されたい. 2000 年, 2002 年のスナップショットに対するこの手法の結果は, [5] および [7] などに載っている.

今回用いたスナップショットから [5], [4] の手法を用いてサイトを抽出した後, 実際にウェブグラフとサイト間グラフを構築した結果は, 表 2 のようになった.

3 最小カット法

本研究ではコミュニティを発見するために Flake らによって提案された最小カット法を主に扱っている. なお, コミュニティとは関連を持っているページの集合のことであるが, そのような理想的コミュニティを見つけるのは困難であるため, Flake らの手法は, 近似コミュニティを出力することを目的としている. 得られた近似コミュニティのサイズが大きく, 実際に関連しているページの割合も大きいならば, それは良い近似コミュニティということになる.

アルゴリズムは以下のようになっている.

1. ウェブグラフ G を作る. G は, ユーザーから与えられたシードページ集合 S と, ページ $v \in S$ にリンクしている (または v からリンクされている) ページの集合 P, $u \in P$ からリンクされているページの集合 Q を点集合とし, それらの間のリンクを辺集合とする.
2. G に仮想ソース, 仮想シンクを加え, 各辺に適切に容量を設定したネットワーク G'_0 を作る. 仮想ソースを s, 仮想シンクを t で表す.
3. $i = 0$ から c について, 以下を実行する. c は与えられた回数とする.
   (a) G'_i の s から 1 番近い最小カットを求める. このとき, G'_i の点集合を V として, 最小カットを表す辺集合を $E(X, V - X)$ で表し, X を s を含む側の点の集合, $(V - X)$ を s を含まない側の点の集合とする.
   (b) $i < c$ ならば, $(X \setminus S) - \{s\}$ の中で最も次数の大きい点 u を見つけ, 新しいシードとみなす. すなわち S に u を加え, (1), (2) と同じようにしてネットワークを作る. これを G'_i とする.
   (c) $i = c$ ならば, $X - \{s\}$ を近似コミュニティとして出力する.
   (d) i を 1 増やす.

なお, G において, S の点と P の点の間の辺は, たとえリンクが片方向にしかない場合でも, 両方向に辺があることにする. これは Flake らが, 片方向のままではシードページにリンクしているページが (最小カットの性質上) コミュニティに入らないのだが, シードページに隣接しているページはシードページに関連している可能性が高いと考え, それらもコミュニティに入れられるようにしたためである.

$G = (V, E)$ とすると, ネットワーク $G'_0 = (V', E') = (V \cup \{s\} \cup \{t\}, E \cup \{(s, x)|x \in S\} \cup \{(x, t)|x \in V\})$ の各辺 $e \in E'$ の容量 $c(e)$ は次のように与えられる.

- $e \in E$ ならば $c(e) = k$. (k は与えられたパラメータで, Flake らは $k = 3$ としている)
- $e = (s, x)$ ならば $c(s, x) = \infty$.
- $e = (x, t)$ ならば $c(x, t) = 1$.

また, Flake らは $c = 3$, すなわち新しいシードを 3 回見つけている (最小カットを求める回数は $3 + 1 = 4$ 回となる). さらに彼らは, G を作る際に, リンク次数が大きい (本論文では 50 以上としている) ページはポータルなどである可能性が高い (よってコミュニティには入れたくない) ため, そのリンク先を無視している.

もうひとつ重要な点として, Flake らは G を作る際にサーバー内へのリンクは無視している.

なお, ページ p からページ q へリンクが張られているとき，p, q が同じサーバー内にあればそのリンクは**サーバー内へのリンク**と呼ぶ．p, q がそれぞれ別のサーバーにあれば**他サーバーへのリンク**と呼ぶ．p, q が同じサイト内にあればそのリンクは**サイト内へのリンク**と呼ぶ．p, q がそれぞれ別のサイトにあれば**他サイトへのリンク**と呼ぶ.

4 従来の枠組みの問題点

上記のように, Flake らの手法は，サーバー内へのリンクは無視している．理由は，企業ではそのサーバーが企業のサイトに対応するシングルサイトサーバーとなっていることが多いからである．その場合例えば，そのサーバー中のページは，閲覧者がサイト中どのページからでも好きなページを見られるようにリンクが多数張られていることが多く，結果としてそのようなサーバー内のリンク構造は密になっていることが多い. 最小カット法はリンクが密な部分を近似コミュニティとして出力するため，サーバー内のリンクを無視しないと同じ企業のページばかり集まってしまい，コミュニティとして意味がないものになってしまう恐れがあるからである．よって確かに, これはページを単位とした枠組みのもとでは必要なことではあるが, 我々は下に述べるようないくつかの問題点を見つけた.

(1) シードページから他サーバーへのリンクがないと, このページは G において t への辺しか無いことになり, コミュニティ生成に寄与できない. シードページがサイトのトップページであり, 他のサイトのリンクは別のリンク用のページのみにあるような場合 (典型的な個人サイトなどに多く見られる) に生じる.

(2) レンタルサーバー, ISP, 大学のサーバー等では 1 つのサーバー中にユーザーのサイトが複数存在する. そのため, サーバー内のリンクを無視すると同一サーバー中のユーザーのサイト間のリンクも無視されてしまうことになり, これらのサイト間の関係は見いだせなくなる.

(3) 関連の強いサイト同士で**相互リンク**を張り合っていることはウェブではよく見られるが, あるサイト A, B 同士が相互リンクを張っていても，いかなるページの組 $(a \in A, b \in B)$ もリンクを張り合っていない場合, ページを単位として考えるウェブグラフでは相互リンクとして扱われない.

5 従来の枠組みとサイトを単位とした枠組みとの比較

サイトを単位とした枠組みを使うと, これらの問題点は (ディレクトリベースドサイトが正しく判別されていれば), 定性的に解決されることは明らかである.

したがって, サイトを単位とした枠組みを用いることで, 最小カット法の結果が改善されると期待される．本論文では, ページを単位とした枠組みと, サイトを単位とした枠組みでの最小カット法の比較実験を 25 のトピックに関しておこなった．まず Google を用いてそれぞれのトピックを検索した結果の中から, リンクの次数が大きいページ 2, 3 個を用意してシードとした．用意したシードページを入力として，ページを点とするウェブグラフとサイトを点とするサイト間グラフに対して最小カット法を適用し，それぞれ近似コミュニティを得て，比較・評価した.

表 3: ウェブグラフとサイト間グラフ上での最小カット法の結果

トピック	ウェブグラフ	サイト間グラフ
カブトムシ	2/8(2/8)	12/15
オーロラ	5/6(2/5)	5/15
イルカ	5/8(3/6)	21/31
風の谷のナウシカ	0/0(0/0)	11/28
椎名へきる	2/2(2/2)	4/5
猫 (1)	5/8(4/7)	8/11
猫 (2)	22/22(5/5)	13/15
犬	2/2(2/2)	5/12
豆柴	1/1(1/1)	5/7
テニス	22/23(11/12)	15/22
頭痛	8/8(8/8)	10/14
トロンボーン	8/8(6/6)	20/25
バイク	9/9(8/8)	20/25
チーズ	4/6(4/6)	9/17
パチンコ	7/11(4/5)	16/21
保田圭	23/33(12/22)	7/10
ラーメン	57/76(6/14)	6/13
野鳥	13/18(12/15)	18/29
D&D	5/17(5/11)	8/15
JoJo	4/4(3/3)	12/21
ワイン	0/11(0/3)	6/11
倉木麻衣	5/12(5/12)	6/10
チェス	3/8(3/6)	4/6
タクティクスオウガ	0/6(0/6)	9/21
コーヒー	12/19(0/4)	5/8

表 4: 平均精度と平均サイズ

グラフ	平均精度	平均サイズ
ウェブグラフ	74%	4.90
サイト間グラフ	66%	12.47

表 3 にはウェブグラフでの結果とサイト間グラフでの結果をそれぞれ載せている．各セルの数値の分母は近似コミュニティのサイズ (ページ数またはサイト数), 分子は実際に関連していたページまたはサイトの数である．ウェブグラフでの結果では, 同じサイトから複数のページをコミュニティに入れてしまうことが多く見られた．これは本質的にはコミュニティのサイズに寄与しないので, ウェブグラフの結果の列のカッコ内に, 得られたサイトの数で数えた結果を記した．例えば「コーヒー」に関する結果では, 12/19(0/4) となっているが, この 12 ページはすべてシードページと同じサイトのものであったので, 実質は新しい情報はまったく得られていないと見なすほうが妥当であろう．なお，シードページや検索エンジンのページは近似コミュニティのページとしてはカウントしなかった．

表 4 では, 平均精度と平均サイズ (それぞれ近似コミュニティ中トピックに関連しているものの割合および数) を表している．なお, 関連しているか否かはコミュニティのすべてのページまたはサイトを実際に調べて, 我々が判定したものである．

結果は，精度は実用上さほど差がないといえるが, 平均サイズ, すなわち関連あるページまたはサイトの数ではサイト間グラフでの結果の方が平均して 2.5 倍の量を得ることができた．

6 まとめ

本論文では，最小カット法というウェブ上でのコミュニティ発見手法についてサイトを単位とした枠組みがどのように役立つかを調べた．我々は，浅野らによって提案されたサイト識別の方法と豊田，喜連川らによって収集，作成された 2003 年 2 月の .jp ドメインを主体とした日本語ページからなるスナップショットを用いて，実際にディレクトリベースドサイトのデータとサイト間グラフのデータを構築し，最小カット法をウェブグラフとサイト間グラフ両方の上で動かし，双方の性能を評価した．

実際に様々なトピックに関して，それぞれシードページの集合を作成し，それぞれに関してウェブグラフとサイト間グラフの上で得られる近似コミュニティを比較したところ，ほとんどの場合についてサイト間グラフ上で動かしたほうが大きな近似コミュニティが得られ，得られた近似コミュニティ内の実際に関連しているサイトの割合も十分実用的な高さであった．

今後の課題としては, 最小カット法の改善の研究である [6] などを参考にすることで, サイト間グラフの特長を活かした改良を最小カット法に施すことで, より精度を改善することなどが挙げられる．

参考文献

[1] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. of the 9th SODA*, pp. 668-677, 1998.

[2] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. In *Proc. of the 8th WWW Conference*, 1999.

[3] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of Web communities. In *Proc. of the 6th SIGKDD*, pp. 150–160, 2000.

[4] Y. Asano, H. Imai, M. Toyoda, and M. Kitsuregawa. Applying the site information to the information retrieval from the Web. In *Proc. of the 3rd WISE*, pp. 83–92, 2002.

[5] Y. Asano. *A new framework for link-based information retrieval from the Web*. PhD thesis, The University of Tokyo, March 2003.

[6] N. Imafuji, and M. Kitsuregawa. Finding a Web community by maximum flow algorithm with HITS score based capacity. In *Proc. of the 8th International Conference on Database Systems for Advanced Applications*, pp.101-106, 2003.

[7] Y. Asano, H. Imai, M. Toyoda, and M. Kitsuregawa. Finding neighbor communities in the Web using an inter-site graph. In *Proc. of the 14th DEXA*, pp. 558–568, 2003.