インクリメンタルPageRankによる重要Webページの効率的な収集戦略

山田雅信[†] 高橋俊行[††]
田浦健次朗[†] 近山隆[†††]

サーチエンジンのインデックスやデータベースを構築するために，今日ではWebクローラが広く利用されている．しかし，すべてのWebページを十分な頻度で収集することは困難であるため，重要なWebページを効率的に収集することが重要となる．本論文では，小さなオーバヘッドで重要なWebページを高速に収集可能なクローラの収集戦略を提案する．さらに，我々の収集戦略によるクローラと従来の収集戦略によるクローラにより，WWWのサブセット上でWebページの収集実験を行い，我々の収集戦略の有効性を確認した．また，我々の収集戦略は少ないオーバヘッドで実装でき，クローラの速度を遅くするものではないことが確認された．

Efficient Collection Strategies of Important Web Pages by Incremental PageRank

Masanobu Yamada,[†] Toshiyuki Takahashi,[††] Kenjiro Taura[†]
and Takashi Chikayama[†††]

Many search engines today use web crawlers to collect and index web pages. Since collecting all the Web pages in a reasonable amount of time is nearly impossible, crawlers should collect important Web pages efficiently. In this paper, we propose a small-overhead strategy that guides crawlers to important web pages fast. Experimental results show our strategy improves previously known starategies. It is also confirmed that our strategy can be implemented with a small overhead, so it does not drag the crawling speed.

1. はじめに

今日，多くのサーチエンジンにおいて，Webページを自動的に収集するプログラム（クローラ，（WWW）ロボット，スパイダーなどと呼ばれる）が用いられている．クローラに収集され，索引付けられる検索対象は膨大であるため，ユーザからの検索要求に対する検索結果も膨大なものになってしまうことが多い．そのため，検索結果をWebページの重要度に基づき的確にランク付けし，検索結果をソート，提示することはサーチエンジンにとって非常に重要な要素となり，この分野に関して，様々な研究がなされ成果をあげてきた．その代表的なものにGoogleのPageRank[1)]がある．

WWWのページ数は急速に増えており，クローラが，すべてのWebページを十分な頻度で収集することは困難である．たとえば，最も多数のページを収集しているGoogleも，1カ月に1度程度の収集頻度で収集しているのは全Webページの30%程度であるといわれている．そのため，高品質な検索エンジンを構築するためには，収集の段階から重要なページを収集する必要がある．ここで，重要なページとは，検索エンジンの目的からは，後に検索結果として上位にランクされやすいページのことである．ページの順位付けにPageRankを用いているのであれば，それはPageRankが高いページということになる．

本論文では，PageRankの高いページの高速な収集を可能にする方式「インクリメンタルPageRankに基づく収集」を提案する．提案方式では，収集済みのWebページと，それらから直接リンクされている

† 東京大学大学院情報理工学系研究科
Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo
†† 科学技術振興機構
Japan Science and Technology Agency
††† 東京大学大学院新領域創成科学研究科
Department of Frontier Informatics, Graduate School of Frontier Sciences, The University of Tokyo

未収集ページからなるグラフに対して，各ページのPageRankの近似値を少ないオーバヘッドで計算，更新していく．そして，未収集ページ中，上記で計算されるPageRankの近似値が最大のものを，次に収集する．新たにページが収集されることでグラフの形が変わると，それに応じて他のページのPageRankの値を更新する．収集中に用いられているPageRankの近似値が，最終的に得られるグラフに対する真のPageRankの近似値に近ければ，本手法が有効となることが期待される．

本論文の以下の構成は次のとおりである．2章で関連研究について述べる．3章でインクリメンタルPageRankについて詳しく述べる．4章で実験結果，5章でまとめと今後の課題について述べる．

2. 関連研究

2.1 PageRankアルゴリズム

PageRank[1]はPageらによって提案されたアルゴリズムであり，その計算により，Webのリンク構造のみを用いてWebページ群から客観的な人気度が高いと思われるページを機械的に算出することができる．PageRankでは，「多くの良質なページからリンクされているページは，やはり良質なページである」という再帰的な関係により，すべてのページの重要度を判定する．この関係を具体的に見ると**図1**のようになる．あるページの現在のスコアを，そのページのoutgoing link数で割った値が，それぞれリンク先のページのスコアに加算されるという関係になっている．

実際の計算は，Webブラウジングにおけるユーザの行動をモデル化（Random Surfer Model）し，PageRankを十分長い時間が経過したある時点におけるユーザがそのページを訪れている確率として計算している．そのモデルは，

- 多くの場合，現在のページに存在するリンクをたどって次のWebページを訪問する，
- ときどきは，ブックマークを利用，既知のURLを直接入力することなどにより，直接はリンクで結ばれていないページを訪問する，

というものである．なお，上記において「ときどき」の値には15%という値がよく用いられている．

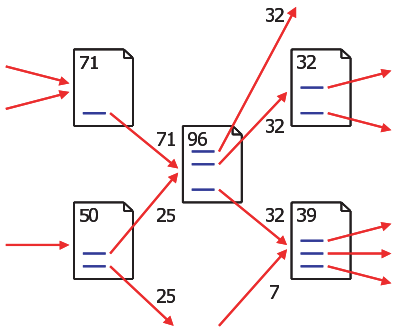

図1 PageRankの概念
Fig. 1 Notion of PageRank.

2.2 クローラおよびその収集戦略

クローラに関する過去の研究は，クロールの大規模，並列化に関するものが多い．代表的なものとしては，WWWの統計情報に基づき再クローリングのタイミングを最適化するための研究，クローラのアーキテクチャおよび実装方法[2]，地理的に分散したWebページを効率的に収集するためのクローラの分散配置アルゴリズム，および，それら多数のクローラを互いに協調動作させるためのアーキテクチャの提案[3]などである．

クローラがすべてのページを短時間で集めることができないため，重要ページを高速に収集するための戦略（収集ページの優先順位付け）が重要となる．なかでも，PageRankを検索結果の順位付けに用いるのであれば，PageRankが高いページを優先的に集める手法が重要となる．文献4)は，Webページの収集過程において，すでに収集済みのページ群に対してPageRankの計算を行い，それに基づいてページを優先度付けする手法を提案している．本論文で我々が提案するインクリメンタルPageRankも同様の方式である．しかし，文献4)は，収集過程にPageRankの計算を行うコストを無視しており，示されたのは提案手法がPageRankの高いページを「先に」収集することだけである．この方式が実用的な価値を持つためには，収集済みグラフに対するPageRankの計算・維持を少ないコストで行わなくてはならず，我々は本論文で，その一方式を提案している．

2.3 PageRankの高速化に関する研究

Adaptive PageRank[6]やBlockRank[7]は，PageRankの計算コストを小さくするための方法論である．Adaptive PageRankはPageRankの伝播を多数繰り返すことを前提としたうえで，繰返し処理の各段階ですでに値が収束したと判断したページを以降の処理から除外，徐々に計算対象を減らしていくことにより全体の計算コストを削減している．BlockRankはページのURLによりWWW全体をいくつかのまとまった集合（Block）とし，Blockごとの PageRankとBlock内の各ページのPageRankを別個に計算，後にその結果を統合することにより最終的なPageRankを求めている．この方法では大規模なPageRank計算を小規模なPageRank計算の集合として処理する

ことにより全体の計算コストを削減している．

さらに，これらを組み合わせて用いることも可能であるが，こられの方式はどれもクローリングと並行して用いることができるほどに，計算コストを大幅に削減するものではなく，また，頻繁にデータが増えていく（インクリメンタル）ような状況での利用は困難である．計算はインクリメンタルなデータに対しても少ないコストで行われなければならず，提案方式ではそれらについても考慮している．

2.4 知的クローラ

収集戦略の特別な場合として，あらかじめほしいページのトピックを与えて，そのトピックに合致する Web ページを重点的に収集するクローラがあり，知的クローラと呼ばれている．その一例として Focused Crawler [5] があげられる．Focused Crawling のための基本的な手がかりは，あるトピックに関するページから直接リンクされているページもまた，そのトピックである可能性が高い，という仮定である．クローラはこれを利用して，収集済みのページの中，求めるトピックへの適合度が高いと判定したページから直接リンクされているページを優先的に選んで収集する．これは本研究の手法と似ているが，我々は「質」の基準として特定のトピックへの適合度ではなく，PageRank を用いている．つまり，あくまで特定のトピックによらない大規模情報検索サービスを提供するサーチエンジンでの利用を想定している．

3. インクリメンタル PageRank に基づくクローラ

前章で述べたように，ページ収集の順序付けに，収集済みページの PageRank を用いたクローラは過去に提案されている [4] が，その方式には，PageRank の計算コストが高いという問題がある．正確な PageRank の計算には大容量のメモリと，PageRank 値の伝播を多数回繰り返すことが必要で，ページが収集されるたびに PageRank を何度も再計算するのは実用的ではない．

また，PageRank の計算量を小さくするための方法論も報告されている [6),7] が，計算コストという問題の根本的な解決策とはなっていない．

我々が提案する手法も文献 4) と同様，収集済みのページ群，およびそこから直接リンクされている未収集ページ（つまり，URL は既知だが，ページ本体はまだ収集されていないページで，以降では「発見済み URL」と呼ぶ）の PageRank を計算する．そしてそのときの PageRank が高い URL を重要と見なし，収集優先度を上げる．ただし，PageRank そのものを求める代わりに，計算を適当なところで打ち切った近似値を求め，コストを大幅に削減する．こうして求める近似値を以下では近似 PageRank と呼ぶことにする．既存の PageRank に基づく手法が，収集優先度の決定にあまり関与しないページもその計算対象とし続けるのに対し，本手法ではこのようなページは以降の計算において計算対象とはならない可能性が高く，そのために多少のオーバヘッドを要したとしても，トータルとしてのオーバヘッドを削減できるため実用的なコストでの計算が可能となる．

この方法は，

- 我々は近似 PageRank を，クローラの収集優先度を決定するために用いるだけであり，検索結果をソートするための PageRank と厳密に一致した値を必要とするわけではない，
- 収集にともなうリンク構造の拡張による近似 PageRank の変化が比較的大きい部分のみの再計算で，目的を達成するのに十分な近似が得られるであろう，

という観測と期待に基づいており，インクリメンタルなデータに対し近似 PageRank を求める手法であることから，これをインクリメンタル PageRank と呼んでいる．

収集中の主なデータ構造として，収集済みページ，および発見済み URL からなるグラフ（以下では「収集済みグラフ」と呼ぶ），および発見済み URL の優先度キュー（以下では「URL キュー」と呼ぶ）がある．収集済みグラフの各ノードには近似 PageRank 値が保持されている．URL キューは近似 PageRank の順番でソートされている．具体的な収集の流れは以下のようになる．

Web ページの収集 URL キューから，最も近似 PageRank 値の高い URL を取り出し，収集する．収集したページの近似 PageRank スコアに 1 を加算する．

リンクの抽出 収集したページに含まれるリンクを抽出し，発見された URL の登録を行う．

近似 PageRank 値の更新 収集済みグラフの変化に従って，近似 PageRank 値を更新する（**図 2**）．この更新の詳細は以下で述べる．

収集優先度の決定 上記の計算によって変化したページの重要度に基づいて URL キュー内で発見済み URL を並べ替える．

以上を繰り返すことによって Web ページの収集を行う．収集時に近似 PageRank のスコアにある値を

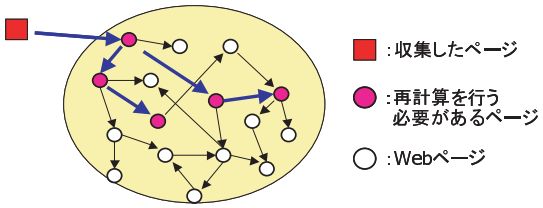

図 2 インクリメンタル PageRank の再計算
Fig. 2 Re-calculation of incremental PageRank.

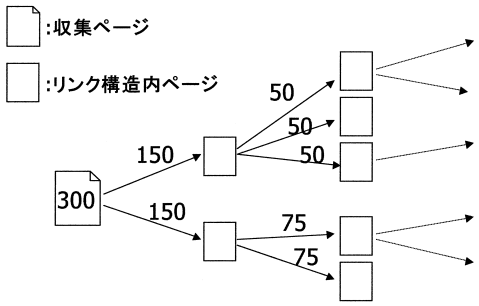

図 3 インクリメンタル PageRank スコアの分配
Fig. 3 Distribution of an incremental PageRank score.

加算しているのは，既存の PageRank が計算に先んじて全ページに初期値を与えるのと同じ意味を持つ．提案手法におけるこの値は，一定のものならば値そのものは任意でかまわない性質のものであり，今回その値として 1 を用いた．つまり，ページが収集された直後の近似 PageRank 値は収集優先度決定時の近似 PageRank 値にこの 1 を加算した値となる．

さらに，近似 PageRank 値の更新は，収集されたページの持つ近似 PageRank 値を，リンク先に等分配することで行われる．必要があれば，リンク先のページにおいても，分配された分をさらにリンク先へと分配することを繰り返す（**図 3**）．

この分配をどこまで行うかで，計算のコストと近似の精度が変化する．以降の実験では以下の 4 つの方式を比較している．

Depth Limit 1（DL1）： 新たに収集されたページから直接リンクされているページだけ更新する．最も計算コストが小さい．

Page Number Limit（PL）： DL1 を行った後，さらにある一定のページ数 L だけ更新を続ける．具体的には，新たに収集したページからの幅優先探索により L ページだけ更新する．L を大きくすると，計算の正確さが増す代わりに計算コストが大きくなる．ここでは 8 を用いたが，これは主に計算時間との兼ね合いから決めたものである．

Value Ratio（VR）： DL1 に加えて以下を行う．分配されるスコアが分配されるページのもともとの近似 PageRank 値と比べて，ある閾値以上大きい場合のみ更新とリンク先ページへの分配を行う．すなわち，ページ p からページ q にスコア β を分配するとする．この分配以前のページ q の近似 PageRank 値を s_q としたとき，β/s_q がある閾値以上なら s_q を更新し，q のリンク先ページ分配し，リンク先ページでも同様に，条件を満たす限り分配を継続する．閾値としては 1.5 を用いた．

Accumulated Value Ratio（AR）： VR と似ているが，あるページ p にスコアが分配されても，p のリンク先への分配はただちには行わずに，p には分配されたがまだリンク先へは分配していないスコアを，p 自身のスコアとは別に記録しておく．これを a_p と書く．a_p と p のスコア s_p の比 a_p/s_p が閾値よりも大きくなった時点で，a_p を p のリンク先へ分配する．閾値としては，29/30 を用いた．

VR や AR は，「あるページのスコアが大幅に変動した場合には，そこからリンクされているページにもその影響を伝播させるべきである」という直感に基づいている．VR では，q に対する「ある 1 回の更新」による q のスコアの変動（β/s_q）が十分少なければ，その更新を無視する．したがってこの方式では，小さな更新が積み重なることによって q のスコアが大きく変化する場合でも，更新が行われないことになる．これを改善したものが AR で，まだ伝播されていない更新を別途累計，記憶しておき，その累計が大きくなった時点で，それらの更新の累計を伝播させるものである．

VR の閾値 1.5 は，スコアが s のページ q に対して，$1.5s$ 以上のスコアが，1 回の更新によって加算された場合にのみ q のスコアを更新するということである．この値を大きくすると結果的にほとんどの更新が無視される．良い閾値を選ぶのは難しいが，我々は，PageRank の高いページは，多数のページからリンクされることによって PageRank が高くなる場合が多いという直感から，この値を小さめに設定している．AR の閾値 30 は，最後に更新の伝播を行った時点でのスコアが s であったページ q のスコアが，$30s$ に達した時点で，q のリンク先への伝播を改めて行うことを意味している．閾値は，小さくするとより頻繁に伝播を行うことになり，計算量が大きくなるため，近似の品質を損なわない範囲で大きな値を選びたい．こ

こでもやはり良い閾値の選び方は難しいが，AR と同じ直感（PageRank の高いページは多数のページからリンクされている）に基づき，大きめの閾値を設定して計算量を減らしている．

4. 実験結果

4.1 使用データセット

使用した Web ページは http://www.yahoo.co.jp を起点として幅優先探索により 2003 年 4 月下旬から 5 月上旬にかけて収集された，約 1650 万ページからなるデータである．クロールは，我々が開発した蔵王（Zao）[8),9)] を用いて行っている．これらのページからリンクを抽出してグラフを構築し，そのグラフをもとに各収集方式による収集のシミュレーションを行う．

4.2 比較した収集戦略

収集戦略とはつまり，URL キューの優先度付けにほかならない．以下に述べるすべての戦略に共通の規則として，戦略による優先度付けができない場合は同一ページ p から発見されたページ群（すなわち p 内に書かれた URL 群）は，p と同一サイトにないものを p と同一サイト内のページよりも優先する．

今回比較した収集戦略は以下のとおりである．

幅優先（Breadth First; BF）： 通常の幅優先探索である．つまり，発見済み未収集ページは，それが発見された順に収集される．優先度の言葉でいえば，早く発見されたものほど優先度が高い．

逆リンク数（Back Link Count; BLC）： あるページをさしている収集済みページの数を，そのページの逆リンク数という．この戦略では，発見済み未収集ページを，逆リンク数によって優先度付け（逆リンク数が大きいほど優先度が高い）する．ただし，逆リンク数が同一の場合，URL がたどるディレクトリ階層の数（'/' 記号の数）が少ないものを優先し，それも等しい場合 URL の文字列長が短いものを優先する．

ページを収集するたびに，それが直接指す発見済み未収集ページの逆リンク数を更新する必要がある．

PageRank（PR）： 文献 4) で提案された方式で，収集済みグラフ上で PageRank をつねに計算しておき，それによって優先度付けを行う．ただし，当然のことながら PageRank の計算は非常にコストがかかるため，1 ページ収集するごとに PageRank を再計算すれば，まったく実用的ではない．ここでは，16 万 5000 ページ収集するごとに 1 回再計算を行っている．これは，全データを収集する間に約 100 回の PageRank 計算を行うことになる．また，各再計算時には前回の再計算時の PageRank の値を初期値として，1 度だけ各ページの PageRank 値を更新している．すなわち，通常の PageRank の計算で行うような収束判定や収束するまでの繰返しは行っていない．

インクリメンタル PageRank（IPR）： 我々が提案する方式であり，1 ページ収集するごとに，3 章で述べた計算を行う．

4.3 評価方法

ここでは，収集戦略を評価するための基準について述べる．なお，便宜上，データセットに存在する Web ページ群を P，データセットからクローラが収集した Web ページ群を p とする．また，時刻 t までに収集したものを $p(t)$ とする．P に関しては，前述のようにあらかじめ恒常的な重要度が求められている．

基準 A

P の重要度の上位のものが $p(t)$ にどれくらいの割合で含まれているかにより評価を行う．重要度が上位のページとは，最終的な PageRank の順位がある閾値以内に含まれるページのことで，以降の実験では全ページの 0.1%，1%，10%を閾値として用いている．収集開始時の割合は 0%であり，データセットに含まれるすべての Web ページを収集し終えたとき，すなわち $P = p$ のとき，この割合は収集戦略に関係なく 100%となる．しかし，すべての Web ページを収集する過程での収集の順序は各収集戦略により異なるため，ある時刻 t での割合は異なってくる．重要度が上位の Web ページの効率的収集という観点において，ある時刻 t における割合が大きいもの，ある収集した割合に対して収集時間 t の短いものほど優れた収集戦略である可能性が高い．

基準 B

クローラにより，ある期間内に収集された Web ページの重要度の合計値により評価を行う．$P = p$ のとき，その総和は収集戦略に関係なく一定の値をとるが，ある時刻 t までの総和は上記と同様の理由により異なる．収集した Web ページの全体的な質を考慮する際，この総和が大きいものほど優れた収集戦略である可能性が高い．

基準 C

データセットにおける重要度ランキングの高いものから順に収集できているかにより評価を行う．具体的にはまず，クローラに収集されたページ i に対して，$orderT(i)$ はページ i がそのクローラによって何番目に収集されたかを表すとする．一方，$orderR(i)$ は，

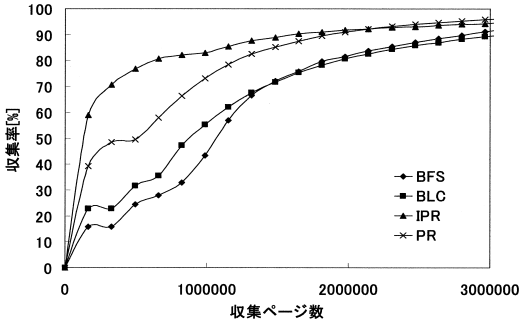

図 4 上位 0.1%の収集率
Fig. 4 The collection ratio of top 0.1%.

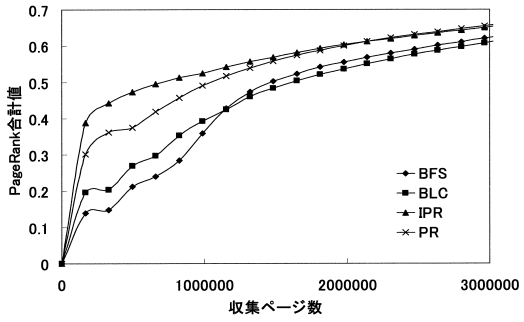

図 5 上位 1%の収集率
Fig. 5 The collection ratio of top 0.1%.

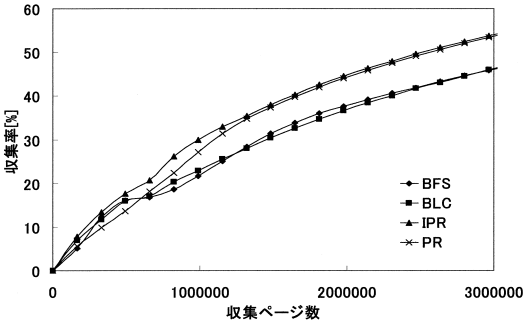

図 6 上位 10%の収集率
Fig. 6 The collection ratio of top 10%.

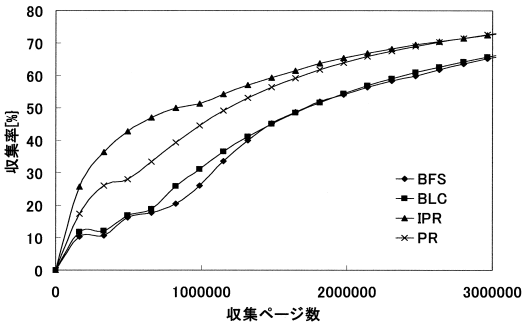

図 7 収集済みページの PageRank 合計値
Fig. 7 The PageRank Sum of collected pages.

ページ i の PageRank が全体の中で何番目に大きいかを表すとする．理想的なクロールは，$orderT(i)$ と $orderR(i)$ の番号が一致し，誤差が大きいほど理想的なクロールからはほど遠いものとなる．

また，理想的なクロールにおいて本来 5 番目に収集されるべきものが 10 番目に収集された場合と，本来 10,000 番目に収集されるべきものが 10,005 番目に収集されるのとでは誤差は同じでもその意味するものが大きく違う．ここでは前者の誤差の方がより深刻である考え，この誤差の重みを大きくする．その際，補正値 c を用いることにより，収集した起点の違いによる影響を緩和する．これにより評価関数を

$$\sum_{i \subseteq P} \frac{|orderT(i) - orderR(i)|}{\sqrt{orderT(i) + c}}$$

とする．最終的には評価しやすいようこれを $WorstCase$ で除算することによって正規化する．$WorstCase$ とはデータセットの重要度の低いものから順に収集してしまった場合の評価関数の値である．

4.4 収集戦略の性能

基準 A において，上位ページを定義する閾値として 0.1%，1%，10%の各値を用いた結果を**図 4**，**図 5**，**図 6** に示す．基準 B，基準 C の結果をそれぞれ**図 7**，**図 8** に示す．すべてのグラフにおいて，インクリメンタル PageRank の打ち切り手法としては DL1 を用いている．

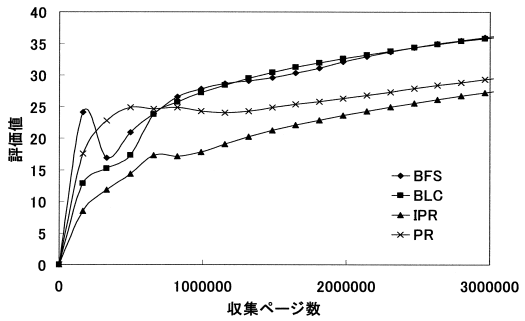

図 8 収集順序の誤差
Fig. 8 The error of a collection order.

どの評価基準においてもインクリメンタル PageRank（IPR）が，最も優れた収集順序を実現していることが読み取れる．もちろん PageRank（PR）は，PageRank 値の更新を頻繁に行うことで IPR よりも優れた収集順序となることが予想されるが，計算コストを考えるとそれによって PageRank の高いペー

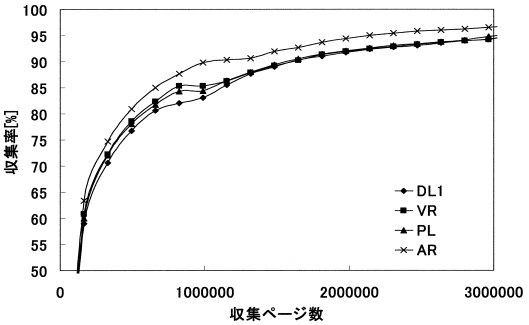

図 9 IPR の計算打ち切り手法の比較
Fig. 9 Comparison of various cut-off methods of IPR.

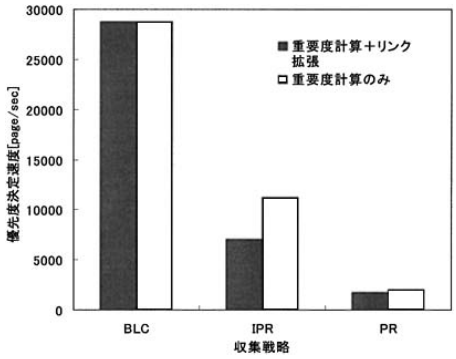

図 10 優先度決定速度
Fig. 10 Priority determination speed.

ジを速く収集することにはならない．

次にインクリメンタル PageRank を用いた収集法において，様々な打ち切り手法による性能を**図 9** に示す．インクリメンタル PageRank の計算の延長を行ったことによりどの打ち切り手法でも収集精度の向上が見られたが，その向上はわずかなものであった．PL のように単に計算の延長を行っただけではコストパフォーマンスは向上せず，VR や AR のように，計算の延長が与える影響度により打ち切るを決定した場合は比較的小さなオーバヘッドでの性能向上を実現することができた．

4.5 グラフ構造・優先度更新の速度

図 10 に，各収集戦略における優先度付けにかかる計算の速度を 1 秒あたりに処理できるページ数によって示してある．計算に用いた CPU は，UltraSPARC 900 MHz（1 CPU）で，すべてのデータがメモリに収まった状態での速度である．

ここでのリンク拡張とは保持している Web グラフを更新することであり，重要度計算とは Web グラフの更新が完了した状態から各収集戦略に基づき各ページの重要度を算出することである．たとえば BLC の場合，ページの重要度を算出するのに Web グラフを使用しないので，重要度計算とリンク拡張の両方を行った場合と重要度計算のみを行った場合で優先度決定速度は同じとなる．

IPR の速度は打ち切り手法 DL1 において約 7,000 ページ/秒で，仮にこの計算のために専用の CPU が 1 台割り当てられているとすると，1 日に 6 億ページを収集できる速度である．これは Google が，数十億ページを 1 カ月に 1 度位の頻度で収集していることを考えれば，クロールのボトルネックとはならない数字である．

もちろんこれは乱暴な計算であり，現実のクロールにおいては，グラフ全体がメモリに収まることは考えられず，それ以前に 1 台の計算機からグラフ全体がアクセスできるわけでもない．したがって近似 PageRank や URL キューの更新自体がディスクやネットワーク I/O を含んだ処理になる．しかしながら，それらの I/O の大部分は提案したような優先度付けを行わないクローラ（たとえば幅優先順に収集を行うクローラ）でも発生する処理であり，我々がそれに加えて行っている処理の負荷は，この実験によって表されている．さらに，ほとんどのクローラが複数の CPU を用いていることや，クローラの速度が多くの場合ネットワークやディスクの速度に束縛されていることを考えると，IPR の負荷はクロールの速度を落とさずに行うことが可能であろうと考えられる．

一方，収集戦略 PR は，IPR に比べて 1/3 程度の性能である．これは，全体の計算コストを抑えるために PageRank の計算を 165,000 ページごとに 1 回としたうえでの数字で，しかも収集の品質は IPR よりも劣っていた．実際には，クロールすべきページ数が増えるに従って，PageRank の計算をさらに間引かなければ，PR の計算コストはますます増大する．

まとめると，大規模なクロールにおいても IPR は実用的であり，一方 PR はそうでないことが予想される．

5. まとめと今後の課題

小さなオーバヘッドで PageRank の高いページを効率的に収集するための収集戦略とてインクリメンタル PageRank を提案した．実際のクロールによって収集された 1,650 万ページを用いた実験により，従来の手法よりも優れた収集戦略であることが確認された．また，優先度付けのオーバヘッドも少なく，現在のクローラに新しいボトルネックを生じる可能性が少ないことも分かった．

また，本手法において特に DL1 による打ち切りを用

いた場合，1 ページあたりに収集優先度を決定するための計算コストである単位計算コストは収集したページに含まれるリンク数に依存するため，そのコストは収集を通してほぼ一定と考えることができる．それに対し，既存の PageRank に基づく手法の単位計算コストは，収集済みページ数に依存するため収集精度と単位計算コストの改善の両立は困難である．このことからも妥当な計算量で優れた収集精度を実現した本手法は，PageRank 以外の既存手法と比べても大規模なクロールに向いているといえる．

また，今後の課題として以下の項目をあげる．

- 使用したデータセットは実際の WWW の大きさからするとまだ十分な大きさとはいえず，より大きなデータセットを用いた実験を行う必要がある．
- 実際のクローラによって収集された際のクロールの境界がどれくらい収集実験に影響を与えているかなどの配慮が不十分であった．上記と合わせて，より実際の WWW に近い形での評価を行う必要がある．
- より最適なクロールのパラメータを選択することでさらに収集の性能を向上させることが可能であると考える．また，最適なパラメータを動的に変更するような機構も重要であると思われ，それらの調査と検討を行う必要がある．
- 収集対象となる WWW の特性を理解し，それを収集戦略にフィードバックする必要がある．
- 提案した手法やその評価は 1CPU 上での動作を前提としたものであった．今後は複数の CPU 上での動作も考慮して，収集戦略や実装を見直す必要がある．
- Web ページの更新頻度に着目する収集手法なども有用であり，本手法との併用も考え，さらなる改善が必要である．

参 考 文 献

1) Page, L., Brin, S., Motwani, R. and Winograd, T.: The PageRank Citation Ranking: Bringing Order to the Web, Technical report, Stanford University (1998).
2) Shkapenyuk, V. and Suel, T.: Design and Implementation of a High-Performance Distributed Web Crawler, Technical report, Polytechnic University (2001).
3) 森 英雄，河野浩之：実測データに基づく分散協調型 WWW データ収集アルゴリズムの性能評価，第 2 回インターネットテクノロジーワークショップ論文集 (1999).
4) Cho, J., Garcia-Molina, H. and Page, L.: Efficient Crawling Through URL Ordering, *Proc. 7th International World Wide Web Conference* (1998).
5) Chakrabarti, S., van den Berg, M. and Dom, B.: Focused crawling: a new approach to topic-specific Web resource discovery, *Proc. 8th World Wide Web Conference* (1999).
6) Kamvar, S., Haveliwala, T. and Golub, G.: Adaptive Methods for the Computation of PageRank, Technical report, Stanford University (2003).
7) Kamvar, S., Haveliwala, T., Manning, C. and Golub, G.: Exploiting the Block Structure of the Web for Computing PageRank, Technical report, Stanford University (2003).
8) Takahashi, T., Soonsang, H., Taura, K. and Yonezawa, A.: World Wide Web Crawler, *Poster Session of the Eleventh International World Wide Web Conference* (2002).
9) Takahashi, T., Taura, K. and Yonezawa, A.: Zao Crawler Home Page (2002). http://www.kototoi.org/zao/

(平成 16 年 1 月 31 日受付)
(平成 16 年 6 月 7 日採録)

山田 雅信（正会員）

1977 年生．2004 年 3 月東京大学大学院情報理工学系研究科電子情報学専攻修士課程修了．同年 4 月より日立製作所ソフトウェア事業部勤務．並列・分散処理，ネットワークプログラミング等に興味を持つ．

高橋 俊行（正会員）

1971 年生．1998 年東京大学大学院理学系研究科情報科学専攻博士課程単位取得退学．技術研究組合新情報処理開発機構研究員を経て，現在独立行政法人科学技術振興機構 CREST 研究員．並列・分散処理，プログラム言語に興味を持つ．

田浦健次朗（正会員）

1969 年生．1997 年東京大学大学院理学博士（情報科学専攻）．1996 年より東京大学大学院理学系研究科情報科学専攻助手．2001 年より東京大学大学院情報理工学系研究科電子情報学専攻講師．2002 年より同助教授．並列・分散処理，プログラム言語に興味を持つ．ACM，IEEE，ソフトウェア科学会各会員．

近山　隆（正会員）

1982 年東京大学工学系研究科情報工学専攻博士課程修了，工学博士．第五世代コンピュータプロジェクトに従事の後，東京大学工学系研究科を経て，現在同新領域創成科学研究科基盤情報学専攻教授．プログラム言語，並列処理，知識処理等に興味を持つ．