Vol.2010-MPS-79 No.8
2010/7/12

Webアーカイブにおける差分収集に用いるWebページの更新依存度分析

廣道尚弓[†1]　吉井直子[†1]
高田雅美[†1]　城和貴[†1]

膨大な数の Web サイトが開設され、それらの多くが定期的、または不定期に更新されるため、Web アーカイブが重要となっている。そこで、Web クローラはいつ収集すべきかという問題に直面している。特に、差分収集は Web ページを収集するにあたり、収集間隔が重要な問題となる。本稿では、Interval Graph と Heap Tree 構造を用いて、差分収集に用いる適切な収集間隔を予測するための新たなモデルを提案する。そのモデルは Web ページの更新依存度を用いて Web サイトの更新間隔を推定する。モデルの予備実験は本稿で示す。

Update Dependence Analysis of Web Pages for Incremental Crawling

Naomi Hiromichi,[†1] Naoko Yoshii,[†1] Masami Takata[†1] and Kazuki Joe[†1]

As huge number of web sites are created anywhere, Web archiving is an important task since most Web sites are updated periodically or non periodically. Any web crawler faces to the problem of "When should we collect?". Especially, incremental crawling has an essential problem of time interval for web page collections. In this paper, we propose a new model to predict the appropriate interval time for incremental web crawling by using an extended interval graph and heap tree structures. The model estimates the update interval of web pages in a web site with the information of modification dependence information of the web pages. Preliminary experiment of the model is shown in the paper.

1. はじめに

Web アーカイブとは Web ページの保存・公開を目的とするアーカイブでインターネット上の図書館としての役割を担っている．近年，Web サイトの開設数は著しく成長しており，多くが定期的，または不定期に更新されている．そのため，世界中で Web アーカイブの開発が盛んに行われている．開発は各国の国立図書館を中心に進められており，最大規模の Web アーカイブ機関は，Web 全体のアーカイブ作成を行っている Internet Archive[1] である．日本では，国立国会図書館が WARP（Web ARchiving Project）[2] を行っている．

Web 情報は日々刻々と変化していくもので，その更新のタイミングは Web ページ毎に異なる．この変化に対し WARP の収集回数では対応できないため，収集できない Web ページ情報が存在する．そこで、更新・変更された Web ページに関してリアルタイムで差分収集を行うサーバを構築する必要がある．

Web ページを差分収集するにあたり，収集する間隔が重要な問題となる．本稿では，差分収集する際に必要となる適切な収集間隔を予測するために，新たなモデルを提案する．そのために，Interval Graph[3] と Heap Tree 構造[4] を用いる．モデルは，Web ページが更新される間隔に関して，Web ページ間の依存度を用いて，収集間隔を推定する．

本稿の構成は，第 2 章で Interval Graph について説明する．第 3 章では Web サイトの分割法について，第 4 章で分析対象 Web ページについて，第 5 章で分析結果について述べ，第 6 章でまとめる．

2. Interval Graph

本章では，Interval Graph について説明する．Interval Graph は，Perfect Graph の一種で，最大クリーク数と彩色数が同数であるという特徴を持つ[3]．グラフ $G = (V, E)$ の区間表現とは，数直線上の区間の集合 I であり，次の 2 つの条件を満たすものである．1 つ目は，V の各頂点は I の区間と 1 対 1 対応する 2 つ目は，G において頂点が隣接するための必要十分条件は，対応する区間同士が重なりを持つことである区間表現を持つグラフを Interval Graph といい，あらゆる方法に応用が可能なグラフ構造である．図 1 は Interval Graph，図 2 は区間表現をそれぞれ表している．

†1 奈良女子大学 大学院人間文化研究科
Graduate School of Humanities and Sciences, Nara Women's University

Vol.2010-MPS-79 No.8
2010/7/12

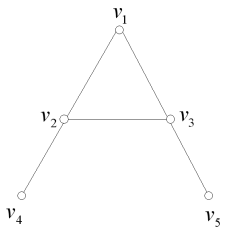

図 1 Interval Graph

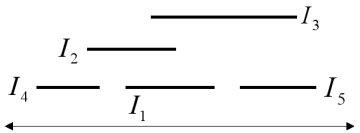

図 2 区間表現

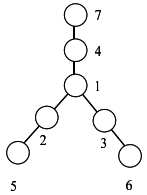

図 3 Max-Tolerance Graph

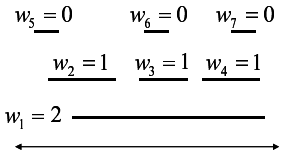

図 4 区間表現

ここで，Interval Graph を一般化したものとして Max-Tolerance Graph[5] を紹介する．このグラフクラスは Interval Graph と同様に区間表現を持っており，違いとして，Max-Tolerance Graph はそれぞれの区間 I_i に重み w_i を持つ．Max-Tolerance Graph$G = (V, E)$ について，頂点が隣接するための必要条件は以下の通りである．

$v_1, v_2 \in E \Leftrightarrow |I_1 \cap I_2| \geq max(w_1, w_2)$

ここで，$w_i \leq |I_i|$ である．すべての重みが 0 となる場合，Interval Graph の定義と一致するため，Interval Graph は重み 0 の Max-Tolerance Graph である．図 3 は Max-Tolerance グラフの一例であり，図 4 はその区間表現を表している．

Interval Graph は多くの利用法が挙げられるが，本稿では区間を Web ページが更新される間隔とし，更新間隔が同じ Web ページを連結させることで収集の効率化を図る．ここで，更新のペースにある規則を見出すことができれば，1 つの Web ページの更新頻度を調査するだけで，同時に収集することができる，というメリットがある．そのために，まず更新のペースに関しての依存関係を調べる必要がある．

3. サイト分割法

本章では，熊谷氏・山名氏によって提案されたサイト分割法[6] について説明する．

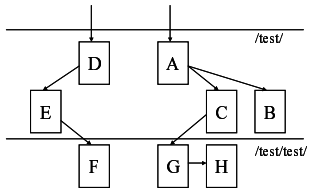

図 5 リンク構造の例

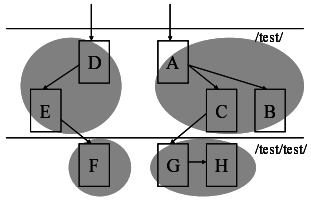

図 7 リンク構造による分割

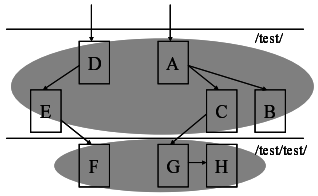

図 6 ディレクトリによる分割

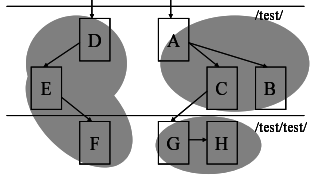

図 8 リンク構造による分割の改良

Web ページの依存関係を分析するにあたり，まず Web ページの更新状況を調査する必要がある．1 つの Web サイトにおいて，サイトのトップページはサイトの更新を反映していると考えられる．1 つの Web サイトには，異なるページが複数存在している．そのようなページ 1 つ 1 つに，更新状況をお知らせする言わばトップページのような役割を持つページが複数存在すると考えられる．そこで，本節では Web サイトを複数のグループに分割する手法について説明する．

サイト分割法についての詳細を説明する前に，リンク構造を簡潔にするために，深度という尺度を定義する．ここで，深度とはトップページからの近さを表す尺度として用いる．トップページを深度 1 と定義し，トップページからリンクを辿るたびに，深度が 1 つずつ上がる．トップページからの経路が複数存在する場合は，最も小さい深度，つまり，トップページからの最短経路を適用する．

3.1 手法 A：ディレクトリによるサイト分割

Web サイトをいくつかのグループに分割するにあたり，まず最も単純な分割方法がディレクトリ単位での分割である．1 つのディレクトリを 1 つのグループとし，ディレクトリ内で最も深度が浅いページをトップページと定める．ここで，グループ内のトップページ以外のページをメンバーページと呼ぶことにする．図 5 に示すリンク構造の場合を考える．この構造の場合，図 6 において，灰色の楕円で囲まれた部分がそれぞれグループとなる．1 つ目のグループ (A,B,C,D,E) の場合，トップページは最も深度の浅いページ A またはページ D

Vol.2010-MPS-79 No.8
2010/7/12

となる．この場合，ページ A，ページ D ともにトップページとする．同様にして，グループ (F,G,H) の場合，トップページはページ F とページ G となる．

3.2 手法 B：リンク構造によるサイト分割

ディレクトリによって分割する手法では，グループ内におけるトップページの数が複数になる場合があり，更新チェックをする際にコストが大きくなってしまう．そこで，本節では，ディレクトリではなくリンク構造によってサイトを分割する手法を紹介する．この手法は，まずディレクトリによってサイトを分割した後，リンク構造によって更に細かく分割する．

まず，自分自身とは異なるディレクトリからリンクされているページをトップページとして候補に挙げる．図 5 において，トップページ候補となるのはページ A，ページ D，ページ F，ページ G である．次に，トップページからリンクで繋がっている同じディレクトリ内のページをグループとする．図 7 において，灰色の楕円で囲まれた部分がそれぞれグループとなる．よって，(A,B,C)，(D,E)，(F)，(G,H) がグループとなる．

3.3 手法 C：リンク構造によるサイト分割の改良版

リンク構造によって分割した場合，手法 B グループ (F) のように，メンバーが 1 ページだけのグループが生成してしまう．これは収集において非効率的となる．そこで，本手法ではメンバーが 1 ページだけの場合はトップページとしないことにする．その結果，ページ F の親ページであるページ E がトップページとなる．図 8 の灰色の楕円で囲まれた部分がそれぞれグループとなる．よって，(A,B,C)，(D,E)，(E,F)，(G,H) がグループとなる．

ここで，更なる効率化をは図るために，チェックするグループ数を削減する．本手法では，「上位のグループが更新されていない場合は，下位のグループの更新チェックを行わない」とする．上位のグループとは，下位のグループのトップページにリンクを張っているページが属するグループのことである．図 8 の場合では，(G,H) の上位グループは (A,B,C) である．例えば，ページ A が更新されていないと判明した場合，ページ G の更新チェックは行わない．また，グループ (A,B,C) の上位グループのトップページが更新されていない場合は，ページ A もページ G もチェックを行わないとする．これにより，チェックするグループ数が大幅に少なくでき，コストを削減することが可能となる．

4. 収集対象 Web ページ

本章では，分析に用いる Web ページについて述べる．収集の対象となるのは奈良女子大学のホームページ[7] とする．更新情報データの取得期間は 2009 年 7 月から 2010 年 3 月までの 9ヶ月である．各ページを 1 時間おきに収集し，更新の有無をチェックする．収集

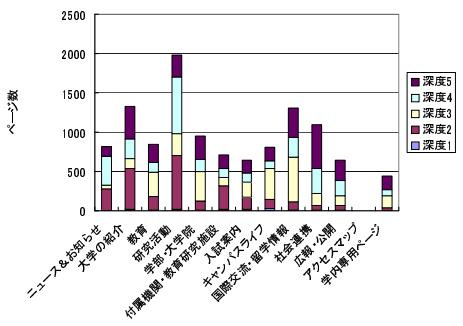

図 9 収集結果

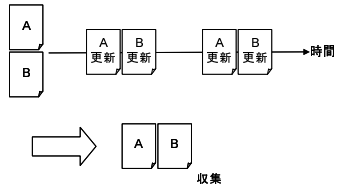

図 10 収集の流れ

には，Web クローラである Heritix[8] と，ファイル取得を行う wget コマンド[9] を用いる．Heritrix，wget 共にリンクを辿って収集を行うことができるため，本実験では辿るリンク数を 5 に設定する．これは，奈良女子大学ホームページを調査した結果，リンクを 5 回辿れば全てのページを網羅できると判明したためである．

図 9 は，全ての収集結果である．横軸は奈良女子大学ホームページのトップページに張られている 13 のリンク名である．縦軸はリンク毎に深度 1〜5 のページ数を表している．

これらの更新情報をもとに，Web ページ間における更新頻度の依存関係を分析する．例として，図 10 において，ページ A が更新されるとその後必ずページ B が更新されるという依存関係が成り立つ場合，ページ A の更新の有無をチェックするだけでページ B の更新の有無も判明する．よって，ページ A が更新された瞬間にページ A，B の両方を収集することが可能になる．これは，収集のコスト削減に繋がる．

5. 更新依存度分析

本章では，Web ページが更新される依存関係について，分析を行った結果について述べる．

5.1 分 析 方 法

分析は図 11 のような流れで行う．まず第 3 章で説明したサイト分割を用いて，収集した Web ページをグループに分ける．図 12 から図 14 はその結果である．手法 A はディレクトリで分割しているため，グループ数が最も少ないことが分かる．また，手法 B と手法 C に関して，手法 B で 1 ページのみのグループが手法 C では上位グループに加えられるため，手法 B より手法 C の方がグループ数が少なくなることが分かる．

Vol.2010-MPS-79 No.8
2010/7/12

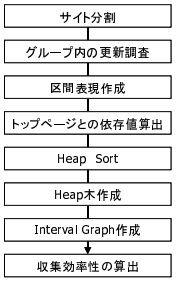

図 11 分析の流れ

	グループ数	総ページ数	1グループあたりの平均ページ数
ニュース＆お知らせ	4	814	202.5
大学の紹介	20	1327	326.75
教育	13	847	208.5
研究活動	18	1980	490.5
学部・大学院	6	954	237
付属機関・教育研究施設	17	712	173.75
入試案内	18	645	156.75
キャンパスライフ	27	811	196
国際交流・留学情報	12	1312	325
社会連携	5	1093	272
広報・公開	10	646	159
アクセスマップ	1	1	0
学内専用ページ	1	440	109.75

図 12 サイト分割の結果 (手法 A)

	グループ数	総ページ数	1グループあたりの平均ページ数
ニュース＆お知らせ	131	814	6.21
大学の紹介	225	1327	5.90
教育	250	847	3.39
研究活動	533	1980	3.71
学部・大学院	101	954	9.45
付属機関・教育研究施設	311	712	2.29
入試案内	228	645	2.83
キャンパスライフ	507	811	1.60
国際交流・留学情報	382	1312	3.43
社会連携	77	1093	14.19
広報・公開	258	646	2.50
アクセスマップ	1	1	1.00
学内専用ページ	125	440	3.52

図 13 サイト分割の結果 (手法 B)

	グループ数	総ページ数	1グループあたりの平均ページ数
ニュース＆お知らせ	106	814	7.68
大学の紹介	167	1327	7.95
教育	231	847	3.67
研究活動	358	1980	5.53
学部・大学院	89	954	10.72
付属機関・教育研究施設	280	712	2.54
入試案内	192	645	3.36
キャンパスライフ	421	811	1.93
国際交流・留学情報	277	1312	4.74
社会連携	61	1093	17.92
広報・公開	138	646	4.68
アクセスマップ	1	1	1.00
学内専用ページ	109	440	4.04

図 14 サイト分割の結果 (手法 C)

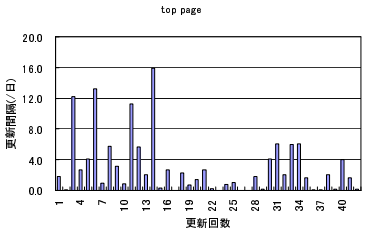

図 15 トップページの更新状況

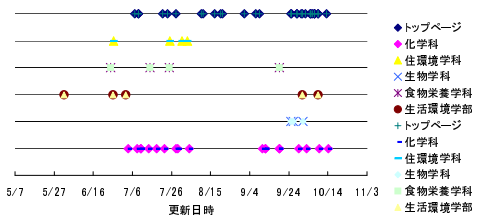

図 16 区間表現

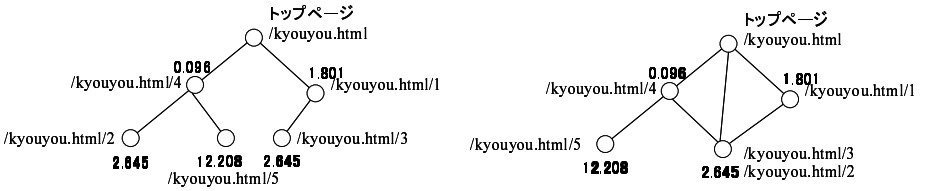

図 17 Heap Tree から Interval Graph へ変換

次に，グループのトップページに関して，更新の有無を調査する．図 15 は奈良女子大学ホームページのトップページの更新間隔を調べた結果である．横軸は更新回数，縦軸は前回の更新からの日数を表している．図からも分かるように，常に一定の間隔ではなく，更新される時期によって大きなばらつきがあることが確認できる．

次に，更新情報をもとに区間表現を作成する．図 16 は区間表現の一例である．グラフの横軸は更新の日時を表しており，それぞれのページが更新された日をプロットしたものである．この区間表現をもとに，各 Web ページについて，グループ内トップページとの依存度を算出する．以下に依存度の定義を示す．

$$\text{依存値} = \frac{\sum(\text{トップページの更新日} - \text{メンバーページの更新日})}{\text{メンバーページの全更新回数}}$$

上式で算出した値が小さいほど，トップページとの更新日時の差がないと分かり，トップページとの依存度が大きいことがいえる．

次に，算出した依存値に関して，Heap ソートを行う．Heap ソートは Interval Graph のグラフ構造として採用する．これは，Heap Tree の根が必ず最小値となることから，トップページとの依存度が最も高いページを基準にすることができるためである．

その後，作成した Heap Tree をもとに，Interval Graph を作成する．第 2 章より，Interval Graph は 2 つの区間が重なるときに辺で結ばれるという特徴を持つ．そこで，作成した Heap Tree から，依存値が同じ値のページを連結させることによって，Interval Graph を作成する．図 17 は Heap Tree から Interval Graph への変換例である．左が Heap Tree，右が Interval Graph を表している．ノードは Web ページを意味しており，数字はトップページとの依存値を示している．この場合，ページ/kyouyou.html/2 とページ/kyouyou.html/3 の依存値が 2.645 で一致しているため，Interval Graph の定義において，区間が重なることを意味している．そこで，この 2 つのページを連結させると、左図の Interval Graph が生成される．よって，2 つに分割されていたページ郡が 1 つに合体したため，先ほど連結させた 2 ページはトップページとも関係性があることがいえ，さらに辺で結ぶことができる．

また，図 18 から図 20 は，Heap Tree から Interval Graph へ変換した際のノード数の変化を表したグラフである．作成した Interval Graph に基づいた更新間隔を用いて収集のシミュレーションを行い，Web ページの依存関係と更新されたページをいかに逃さずに収集できるかということに関して，関係性を分析する．分析には，カバー率という指標を定義す

Vol.2010-MPS-79 No.8
2010/7/12

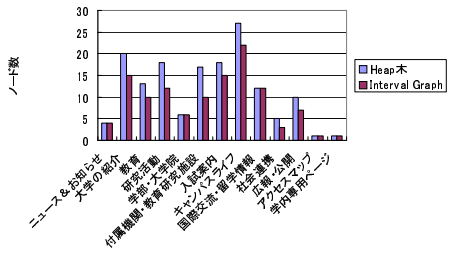

図 18 Heap Tree から Interval Graph への変換 (手法 A)

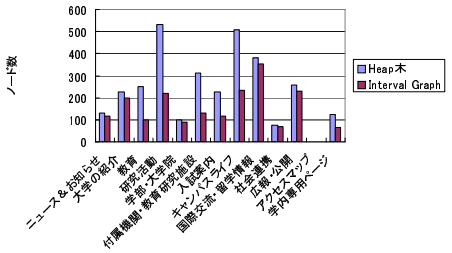

図 19 Heap Tree から Interval Graph への変換 (手法 B)

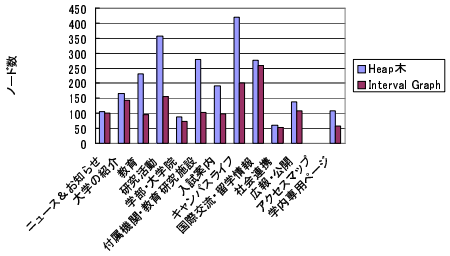

図 20 Heap Tree から Interval Graph への変換 (手法 C)

る．カバー率とは，更新されたページをどの程度網羅的に収集できているかを示している．以下に定義を示す．

$$\text{カバー率 (\%)} = \frac{\text{収集したページ中の更新回数}}{\text{全更新回数}}$$

この指標が 100%に近づくほど，その手法が優れているといえる．分析において，依存値によって Heap Tree から Interval Graph に変換する際，連結させる基準を変化させ，それによって生じる違いを確認する．

5.2 分析結果

本節では，分析の結果について述べる．図 21 から図 24 は収集回数とカバー率の関係を表したものである．それぞれ，Heap Tree から Interval Graph に変換する際に，ノードを連結させる基準値を変化させている．図 21 は，連結させる基準となる依存度から，上下に 50%の区間を設けたものである．図 22 は区間の幅を 20%，図 23 は 5%，図 24 は区間の幅を持たせずそのままの値を使用している．それぞれ横軸は更新したページを何%収集できて

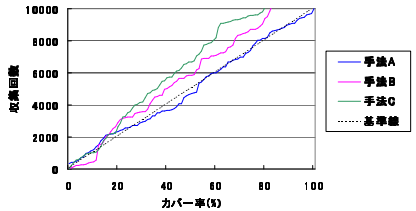

図 21 区間幅：50%

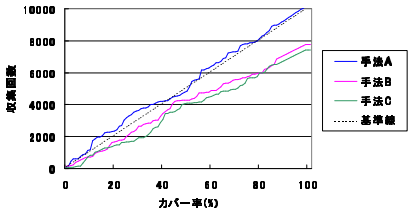

図 22 区間幅：20%

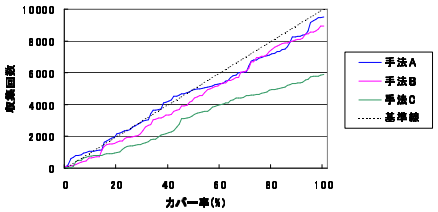

図 23 区間幅：5%

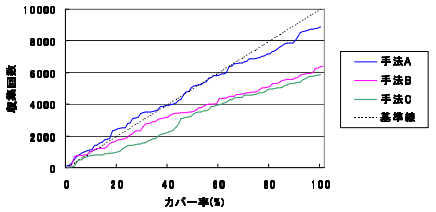

図 24 区間幅：なし

いるかというカバー率を表しており，縦軸は収集回数を表している．実験では 10000 回収集を行っている．

図 21 は 50%の区間幅を持たせた結果である．グラフの点線は基準線を表しており，全ページの更新間隔の平均値を用いた場合の結果である．黄線は手法 A を用いてサイト分割を行った場合の収集結果である．同様に，茶線が手法 B，青線が手法 C をそれぞれ用いた場合の結果である．全ての手法において，カバー率が基準線より下回っていることが分かる．10000 回収集した場合，手法 A の場合のみ基準線と同様なカバー率を保っている．手法 B，手法 C に関しては 10000 回収集してもカバー率は 85〜90%に留まっている．これは，算出した依存度に前後 50%の幅を持たせたため，依存度が全ページにおいて一律になったことが原因であると考えられる．

図 22 は 20%の区間幅を持たせた結果である．図 21 と同様に，基準線，手法 A，手法 B，手法 C をそれぞれ表している．図 21 と比較して，手法 A に変化はないが，手法 B と手法 C は基準線よりカバー率が上回っていることが分かる．これは，図 21 より，区間の幅を狭めた結果であると考える．手法 A に関しては，ディレクトリという固定された基準で分割しているため，依存度を変化させてもカバー率は変化しないと分かる．手法 B と手法 C の場合，約 8000 回でほぼ更新されたページを収集可能であるため，平均値をもとに収集する

Vol.2010-MPS-79 No.8
2010/7/12

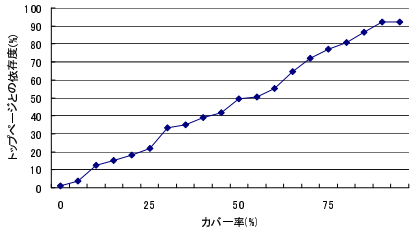

図 25 依存度とカバー率の関係

より，約 2000 回分の収集を削減できることが分かる．

図 23 は 5%の区間幅を持たせた結果である．このグラフも同様に，基準線，手法 A，手法 B，手法 C をそれぞれ表している．図 22 と比較して，手法 A に関しては初めて基準線よりカバー率を上回る結果となった．しかし，手法 B に関しては変化しないことが分かる．手法 C は図 22 よりさらにカバー率が向上していることが見て取れ，約 6000 回の収集でほぼ更新されたページを収集可能である．

図 24 は求めた依存値をそのまま用いて収集をシミュレーションした結果である．区間に幅を用いていないため，最も精度の良い結果となっていることが分かる．しかし，この方法では収集する全 Web ページの更新間隔とその依存関係を事前に調査する必要があるため，膨大な Web ページを所有する Web サイトの場合はほぼ不可能に近い．そこで，図 21 から図 23 のように，ある程度の依存関係が分かれば，その他のページは区間の幅を持たせることによって更新間隔を調べることなく同時に収集することが可能となる．本実験の場合は，手法 A に関しては図 23 と図 24 で変化が見られないため，区間の幅を 5%持たせても収集に影響はないと考える．また，手法 B に関しては図 23 と図 24 では収集回数に約 2000 回分の差が見られるため，手法 B を用いてサイト分割した場合は全更新情報を事前に取得する必要がある．手法 C は手法 A と同様に，図 23 と図 24 で変化が見られないため，区間の幅を 5%持たせても収集に影響はないと考える．

図 25 はカバー率と依存度の関係性を表したものである．グラフの横軸はカバー率，縦軸は主なグループのトップページと，全収集ページのトップページとの依存度を示している．依存度が高いほど，更新されたページを収集するカバー率が高いことが分かる．よって，Web サイトのトップページとの依存度が高いほど，更新されたページが同じ周期で収集される可能性が高いことがいえる．

6. ま と め

本稿では，更新された Web ページを差分収集する際に重要となる収集間隔を予測するためのモデルの提案を行った．その際，Web ページ間の依存関係を用いて，その更新間隔を推定する．依存関係は Heap Tree 構造を持つ Interval Graph を用いて表す．収集した Web ページをディレクトリ構造とリンク構造でそれぞれ分割し，リンク構造においては更に改良して分割を行う．

事前に取得した更新情報をもとに，依存関係を用いた手法をシミュレーションし，全更新ページの何%を収集可能であるか，更新回数との関係性を用いて分析した．その際，依存関係を表す指標をそのまま用いては，全ページの更新情報を事前に調べる必要があり，コストの削減に繋がらないため，指標に幅を持たせ，更新情報を調べる Web ページ数自体の削減を図る．その結果，リンク構造を用いて分割した手法は，幅を持たせない場合が最も精度が良くなった．しかし，ディレクトリ構造を用いた手法とリンク構造を用いて改良した手法では，幅を前後に 5%持たせた場合でも，幅を持たせない場合と精度に違いが見られなかったため，有効であると確認した．また，Web サイトのトップページと分割手法を用いたそれぞれのグループのトップページとの依存関係と，カバー率の関係について分析を行った．その結果，依存度が大きい，更新されたページを収集するカバー率が高いことが確認できた．よって，Interval Graph を用いて Web サイトの依存関係を表し，その関係性が強いほど，同じ収集周期で更新された Web ページを取得することが可能となる．

参 考 文 献

1) Internet Archive:http://www.archive.org/index.php
2) 国立国会図書館 インターネット情報選択的蓄積事業　WARP:http://warp.ndl.go.jp/
3) G. Hajos, Uber eine Art von Graphen, Internat. Math. Nachr. 11 (1957)
4) Horowitz, Ellis Sahni,Sartaj Anderson-Freed,Susan:Fundamentals of Data Structures in C (2ND),Silicon Pr (2007).
5) GLUMBIC M.C.,TRENK A.N.:Tolerance Graph, CAMBRIDGE UNIVERSITY PRESS (2004).
6) 熊谷 英樹, 山名 早人,” リンク構造を利用した Web ページの更新判別手法”,DEWS2004 (2004.3)
7) Nara Women’s University:http://www.nara-wu.ac.jp/
8) Heritrix:http://crawler.archive.org/
9) wget:http://www.gnu.org/software/wget/