

平成 28 年度 修士論文

ノード次数を考慮した

P 2 P ランダムウォークファイル検索の性能評価

平成 29 年 2 月提出

千葉大学大学院工学研究科
建築・都市科学専攻都市環境システムコース
指導教員 塩田 茂雄 教授

15TM0358

松尾 仁一朗

ABSTRACT

Peer-to-peer networks in which users communicate directly as peers with equal peers are used recently. Because the clients individually process the P2P network, processing requests and service requests from a large number of users do not concentrate on a specific server, and the network load can be reduced. However, it is difficult to obtain the information required by the user in the P2P network without the server. In Gnutella's method called flooding that broadcasts a query for content search to a neighboring node like Gnutella, since the query reaches the end of the network, it is possible to search without fail, but the problem of increasing the network load due to the occurrence of redundant queries there is.

Various studies are being conducted to solve this problem. In this research, we focus on a method to randomly walk one or more queries in the network and retrieve content that meets the conditions [1] [2]. Unlike flooding, queries are not copied at nodes in the middle and their number does not increase. For efficient content retrieval, it is desirable that queries can visit a corrupt node on the network in as short a time as possible. A random walk has been proposed that preferentially visits high-order nodes while avoiding visited nodes [3].

In this research, we focus on a method to randomly walk one or more queries in the network and retrieve content that meets the conditions. Unlike flooding, queries are not copied at nodes in the middle and their number does not increase. Evenly visiting nodes in the network theoretically showed that search efficiency is improved and predicted that higher order node bias of basic random walk search can be relaxed by giving priority to low order nodes. Therefore, we propose a "node degree dependent type" random walk search in which the information of the degree of the adjacent node can be used when the query selects the next destination from the adjacent nodes, and a method of preferentially visiting the lower order nodes It is clarified by simulation experiment that it is effective for content search. In addition, by having nodes or queries have already visited node information, we devise multiple methods to avoid revisiting visited nodes and compare the effectiveness with simulation experiments.

概要

ユーザ同士が同格のピアとして直接通信を行う P2P ネットワークが近年利用されている。P2P ネットワークはクライアントがそれぞれ分散的に処理を行うため、多数のユーザからの処理の要求やサービスの要求が特定のサーバに集中することがなく、ネットワーク負荷を軽減することができる。しかし、サーバの存在しない P2P ネットワークではユーザが必要としている情報を得ることは困難である。Gnutella のようにコンテンツ検索のためのクエリを隣接ノードへブロードキャストするフラッディングという手法ではネットワークの末端までクエリが届くため、もれなく検索することが可能だが、冗長なクエリの発生によりネットワーク負荷が増大するという課題がある。

この課題を解決するためにさまざまな研究が行われている。本研究では、一つもしくは複数のクエリをネットワーク内にランダムウォークさせ、条件に合うコンテンツを検索させる手法に焦点を当てる[1][2]。フラッディングと異なり、クエリは途中のノードでコピーされてその数が増えたりしないものとする。効率的なコンテンツの検索のためには、クエリがネットワーク上のすみずみのノードまでなるべく短時間で訪問できることが望ましい。既訪問のノードを避けながら、高次数ノードを優先的に訪れるランダムウォークが提案されている[3]

本研究では、一つもしくは複数のクエリをネットワーク内にランダムウォークさせ、条件に合うコンテンツを検索させる手法に焦点を当てる。フラッディングと異なり、クエリは途中のノードでコピーされてその数が増えたりしないものとする。ネットワーク内のノードを均等に訪問することが、検索効率を高めることを理論的に示し、低次数ノードを優先することで基本ランダムウォーク検索の高次数ノードバイアスを緩和することができると予想した。そこで、クエリが隣接ノードの中から次の移動先を選ぶ際に隣接ノードの次数の情報が利用できる「ノード次数依存型」ランダムウォーク検索を提案し、低次数ノードを優先的に訪問させる手法がコンテンツ検索に有効であることを、シミュレーション実験により明らかにする。加えて、既訪問ノード情報をノードもしくはクエリに持たせることにより、既訪問ノードの再訪問を回避する複数の手法を考案し、その有効性をシミュレーション実験により比較する。

目次

第1章 序論	1
1.1 背景	1
1.2 目的	2
1.3 論文の構成	3
第2章 関連事項	4
2.1 クライアントサーバーモデル	4
2.2 P2P	6
2.2.1 Pure (ピュア) P2P モデル	7
2.2.2 Hybrid (ハイブリッド) P2P モデル	8
2.2.3 スーパーノードモデル	9
2.3 オーバーレイネットワーク	10
2.3.1 構造型オーバーレイ	11
2.3.2 非構造型オーバーレイ	11
2.4 関連研究	12
第3章 検索手法	13
3.1 フラッディング	13
3.2 有効なコンテンツ検索の考察	15
3.3 ノード次数を考慮したランダムウォーク検索	17
3.4 既訪問ノード回避アルゴリズム	18
3.4.1 ノード記録型	18
3.4.2 クエリ記録型	19
3.4.3 複数クエリを用いた場合	20

第4章 シミュレーション評価：単一クエリ	21
4.1 評価条件	21
4.2 既訪問ノード回避なし	22
4.3 ノード記録型既訪問ノード回避アルゴリズム	25
4.4 クエリ記録型既訪問ノード回避アルゴリズム	27
第5章 シミュレーション評価：複数クエリ	31
5.1 評価条件	31
5.2 複数クエリ	32
5.2 クエリ数による比較	34
第6章 まとめ	40
参考文献	38
謝辞	40

第 1 章

序論

1.1 背景

インターネットの世界では, 要求を送るクライアントと処理を担当するサーバからなるクライアント-サーバ・モデルが一般的である. クライアント-サーバ・モデルでは, サーバがデータを蓄積し, 処理を行うため, 情報の管理が容易となる. しかし, クライアント数の増加にともない, サーバ側の処理負荷が高くなる傾向にある. さらに, サーバが停止した場合, サービス自体が停止する. そのため, サーバには多数のアクセスに耐えるため高い処理能力, 広帯域の回線が必要である.

それに対して, ユーザ同士が同格のピアとして直接通信を行う P2P ネットワークが近年利用されている. P2P ネットワークはクライアントがそれぞれ分散的に処理を行うため, 多数のユーザからの処理の要求やサービスの要求が特定のサーバに集中することがなく, ネットワーク負荷を軽減することができる. しかし, サーバの存在しない P2P ネットワークではユーザが必要としている情報を得ることは困難である. Gnutella のようにコンテンツ検索するためのクエリを隣接ノードへブロードキャストするフラッディングという手法ではネットワークの末端までクエリが届くため, もれなく検索することが可能だが, 冗長なクエリの発生によりネットワーク負荷が増大するという課題がある.

この課題を解決するためにさまざまな研究が行われている. クエリの生存時間 (TTL) に制限を設け, 段階的に制限を緩めながらフラッディングを繰り返す *iterativedeepening*[4]や, ネットワーク内にツリー構造を持つオーバーレイを構成し, そのツリーに沿ってクエリをブロードキャストするフラッディングを改良した *LightFlood*[5]が提案されている.

また, フラッディングのように全ての隣接ノードへクエリをブロードキャストするのではなく, 隣接ノードの中からランダムに一つ選択しクエリの転送を行なうことを繰り返しコンテンツの検索を行うランダムウォークベース検索が提案されている[6][7]. k-

random walk と呼ばれる複数個のメッセージを同時にネットワーク内でランダムウォークさせる手法[6]や、既訪問のノードを避けながら、高次数ノードを優先的に訪れるランダムウォークを提案されている[3]. 塩田らは各ノードが自分を頂点とする TTL 制限付き最短路木を保持し、その最短路木に沿ってクエリを転送するネットワークブロードキャストを提案した[7]. また、ランダムウォーク型検索手法として、自分から n ホップ以内に位置するノード集合の経路情報およびファイルインデックスを所持しておきコンテンツ検索のヒット率を向上させる手法や[8],[9], クエリを受信した向きとはなるべく異なる向きへとクエリを転送する指向性ランダムウォーク検索を提案した[10].

低次数ノードの重要性に着目した研究として、(ネットワークを制御する上で) 影響力の大きいノードが、いわゆる次数の大きいハブノードには少なく、むしろ低次数ノードに多いことや、情報のより速やかなネットワーク内への拡散のためには、情報を受け取ったノードは低次数ノードに優先的にその情報を送信することが望ましいことがしめされている[11][12].

1.2 目的

本研究では、一つもしくは複数のクエリをネットワーク内にランダムウォークさせ、条件に合うコンテンツを検索させる手法に焦点を当てる. フラディングと異なり、クエリは途中のノードでコピーされてその数が増えたりしないものとする. ネットワーク内のノードを均等に訪問することが、検索効率を高めることを理論的に示し、低次数ノードを優先することで基本ランダムウォーク検索の高次数ノードバイアスを緩和することができると予想した. そこで、クエリが隣接ノードの中から次の移動先を選ぶ際に隣接ノードの次数の情報が利用できる「ノード次数依存型」ランダムウォーク検索を提案し、低次数ノードを優先的に訪問させる手法がコンテンツ検索に有効であることを、シミュレーション実験により明らかにする. 加えて、既訪問ノード情報をノードもしくはクエリに持たせることにより、既訪問ノードの再訪問を回避する複数の手法を考案し、その有効性をシミュレーション実験により比較する.

1.3 論文の構成

以下に、本論文の構成を示す。

第1章 序論

本研究の概要について述べる。

第2章 関連事項

P2P ネットワークの概要や関連研究について述べる。

第3章 検索手法

P2P ネットワークで用いられるファイル検索手法について述べる。

第4章 シミュレーション評価：単一クエリ

単一クエリで各検索手法をシミュレーションで比較する。

第5章 シミュレーション評価：複数クエリ

複数クエリで各検索手法をシミュレーションで比較する。

第6章 まとめ

本研究のまとめを述べる。

第2章

関連項目

2.1 クライアント-サーバ・モデル

クライアント-サーバ・モデルとは、パソコンなどのクライアントのように何らかの処理を要求する側と、クライアントの要求に従って処理を行う側であるサーバの役割分担が明確に分かれたシステムを指す。すなわち、このモデルでは、クライアントの指示によってサーバが処理を行い、その結果をサーバからクライアントに返す。

クライアント-サーバ・モデルの典型的な例として、World Wide Web (WWW) システムが挙げられる。例えば、ユーザは Web クライアントであるブラウザを用いてホームページを閲覧する場合、まずこのホームページのアドレスを入力する。ブラウザは入力されたアドレスを閲覧したいという要求を Web サーバへ送信する。そして、Web サーバが該当するデータをブラウザへ送り返すことで閲覧することができる。Web サーバは各種ファイルやメタデータを管理し、処理の大半を担う。なお、ここではメタデータとは各種ファイルに関する属性情報を指す。例えば、ファイル名やファイルサイズ、ファイルフォーマット種別、ファイル所有者の IP アドレスやポート番号などがあげられる。データがサーバに蓄積し、処理を行うため、情報の管理が容易となる。しかし、すべての情報がサーバに集中するため、クライアント数が増加するとともにクライアントからサーバへの要求も増加し、サーバからクライアントへのレスポンスも増加する。そのため、サーバ側のトラフィックが増加し、処理負荷が高くなる傾向にある。また、処理負荷に関わらず、何らかの要因でサーバが停止した場合、クライアントからの要求はすべて受け付けることができなくなり、サービス自体が停止す

る。そのため、サーバには多数のアクセスに耐えるため高い処理能力、広帯域の回線を使用することが必要となる。また、ロード・バランサーと呼ばれる負荷分散装置で1つのサーバに処理が集中しないように分散させるなどの対策が必要である。

一般的にクライアント・サーバ・システムは、P2Pを利用したシステムよりも初期コストや運用コストが高く、データやユーザの集中管理や制御を行うのに向くシステムである。また、P2Pシステムと比較してシステム・アーキテクチャが単純なため、小規模サービスであればP2Pシステムよりも開発が容易な場合がある。

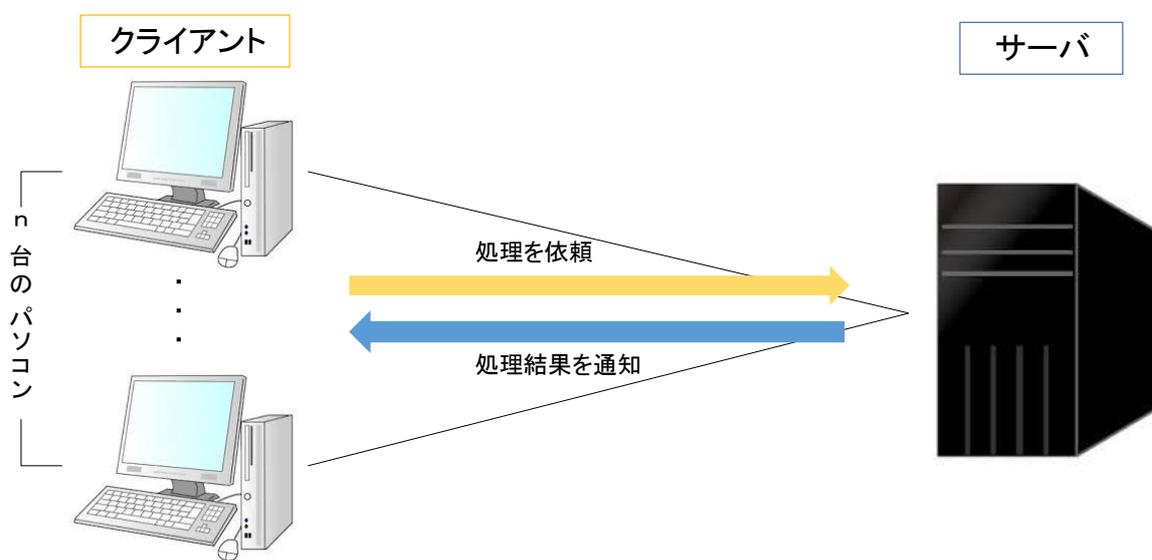


図1. クライアント・サーバ・モデル

2.2 P2P

P2P（ピア・ツー・ピア）とは「対等・同等」という意味を持つ“Peer”という言葉の頭文字を用いた略語であり、正式には「Peer to Peer」と呼ばれる。ピアとは、インターネット・システムにおけるコンピュータなどの端末を指しており、コンピュータ同士が直接かつ対等に通信を行うネットワークのことを P2P ネットワークという。また、ネットワーク内のコンピュータ端末をノード（Node）とも呼ぶ。ノードの接続関係をリンク（Link）と呼び、リンクで接続されたノードは互いに隣接関係であることを示す。あるノードに接続されているリンク数をそのノードの次数と呼ぶ。

P2P における各コンピュータ端末がそれぞれ対等であることは、クライアントとサーバの区別を行わないことを意味する。つまり P2P ネットワークに存在するすべてのコンピュータがあるときはサーバとして処理を行い、あるときはクライアントとして要求を送る。そのため、必要に応じて機能を切り替えることができる、中央サーバが必要ないネットワークになっている。さらに、すべてのコンピュータが同等に機能し情報を分散させることから、情報を集約させておく中央サーバが基本的には必要ない。P2P ネットワークとは二役をこなす同等なコンピュータが相互接続された。ネットワークシステムであるといえる。

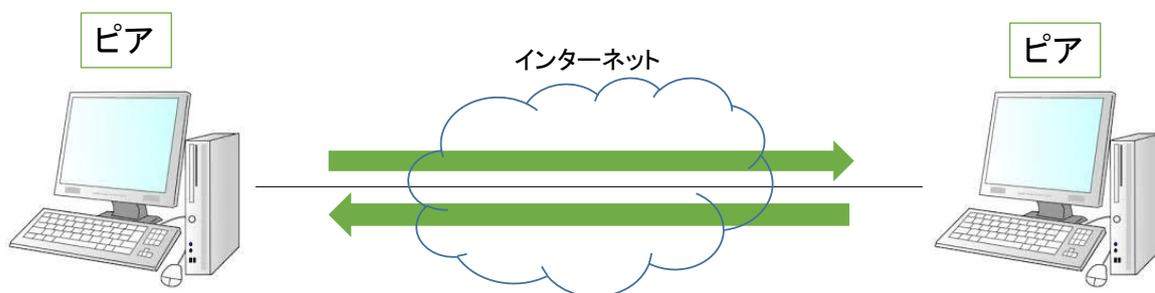


図 2. P2P

P2P はその仕組みによりいくつかの種類に分類できる。本稿では、

- Pure（ピュア）P2P モデル
- Hybrid（ハイブリッド）P2P モデル
- スーパーノードモデル

の 3 つに分類された各 P2P アーキテクチャについて述べる。

2.2.1 Pure（ピュア）P2P モデル

PureP2P モデルは、サーバを介することなくピア端末（パソコンなど）同士のみが直接通信を行うものである。ピュア P2P ではファイルとメタデータは各ピアが保有している。すなわち、各ピアがクライアントとサーバの機能を両方兼ね備えていることになる。また、各ピアは一般的に同一の機能を持っている。

ピュア P2P モデルでは、自律的に P2P ネットワークを形成する。自分以外のピアに処理を依頼するには処理依頼メッセージを発行するが、そのメッセージはリンクをたどってそのメッセージはリンクをたどって他のノードに流通していく。

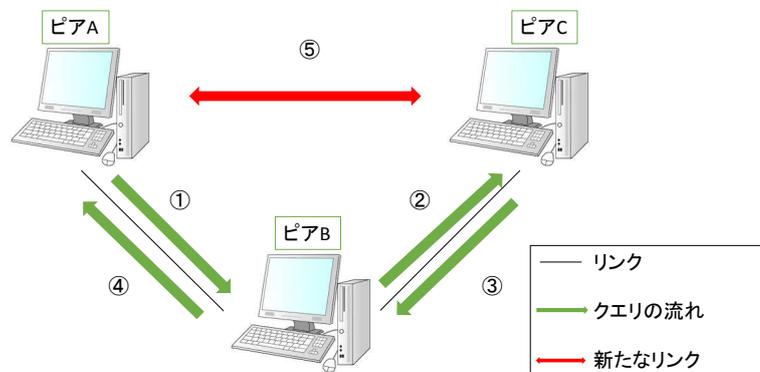


図 3. PureP2P モデルにおける検索手順

例えば、図 3 において、ピア C が保有するファイルをピア A が検索するときのファイル検索手順を以下に示す。

- ① ピア A はクエリをピア B に送信する。
- ② ピア B はファイルを保有しないため、クエリをピア C に送信する。
- ③ ピア C はファイルを保有しているため、メタデータをクエリの送信元へ送信する。
- ④ ピア B はメタデータを送信元であるピア A に送信する。
- ⑤ ピア A はメタデータを基にピア C と直接通信する。

Pure P2P の特徴として、クライアント・サーバ・モデルと比較して、

- (1) 耐障害性が高い。
- (2) スケーラビリティが高い。(多くのピアが参加できる、システム全体で大量のコンテンツが保有できる)
- (3) 第三者によって、ピアの制御や集中管理をすることが難しい。

などが挙げられる。また、匿名性の高いファイル共有システムが、PureP2P を使っているのも特徴（3）と関連がある。なお、あるピアが最初に P2P ネットワークに参加するためには、すでに P2P ネットワークに参加している各ピアの情報（IP アドレスやポート番号）が必要である。

2.2.2 Hybrid（ハイブリッド）P2P モデル

Hybrid P2P モデルでは、各種ファイルのメタデータをインデックス・サーバ（検索サーバ）と言われるサーバで管理し、ファイル自体（音楽ファイルや画像ファイルなど）は、各ピアが保有する。インデックス・サーバとはクライアントが検索したいコンテンツなどのキーワードを指定して問い合わせると、サーバの中から該当するコンテンツを検索するサーバである。

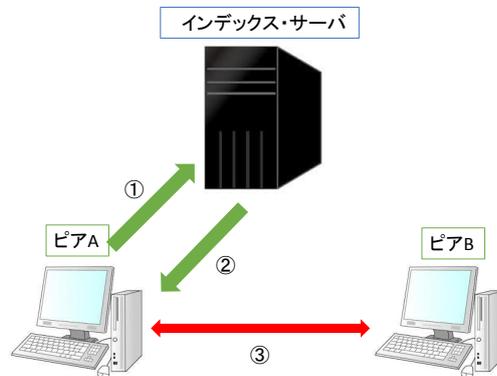


図 4. HybridP2P モデルにおける検索手順

例えば図 4 において、ピア B が保有するファイルをピア A が検索するときのファイル検索手順を以下に示す。

- ① ピア A はインデックス・サーバに検索要求を送信する。
- ② インデックス・サーバは該当するファイルを保有するピア B のメタデータをピア A に送信する。
- ③ ピア A はメタデータを基にピア C と直接通信して

HybridP2P モデルにおけるインデックス・サーバの役割を以下に示す。

- (1) ピアがアップロードしたメタデータの蓄積
- (2) ピアからの検索要求によるメタデータの蓄積
- (3) メタデータのピアへの提供

なお、インデックス・サーバの補助的な役割として、

- (4) ピア間で直接通信できない場合に、サーバ経由でピア間がファイルの送受信を可能にする（NAT 越え）

(5) ユーザの認証を行う

などが挙げられる。代表例として Napster（創業当初）がある。

クライアント-サーバ・モデルと比較するとファイルの送受信はピア間で直接行うため、インデックス・サーバ側にかかるトラフィックや処理負荷を軽減することが可能である。しかし、サーバが停止するとサービスそのものが停止してしまう。そのため、インデックス・サーバの負荷分散対策は、クライアント-サーバ・モデルと同様に必要となる。ただし、PureP2P モデルと違い、ユーザ制御やファイル流通の集中管理が可能となる。

2.2.3 スーパーノードモデル

スーパーノードモデルは、PureP2P モデルを変形したモデルであるため、一般的には PureP2P モデルの特徴を備えている。すべてのピアがファイルのアップロード、ダウンロードなどの同一機能をもつ PureP2P モデルとは異なり、スーパーノードと呼ばれる検索機能なども備えるピアがネットワークに参加していることが特徴である。ピア群から自律的に選択されたスーパーノードは、一般のピアとは異なり高度な処理を行い、特別なデータ（例えば個人情報に関するメタデータ）を管理する場合がある。なお HybridP2P モデルのインデックス・サーバと同様な機能を持つ場合がある。

また、スーパーノード群でネットワークを構成し、スーパーノードの高機能なサービスの提供を支援する。スーパーノードになる条件としては、ピアノ処理性能（メモリ容量や CPU の使用など）が高いこと、グローバル IP アドレスを持って居ることなどが挙げられる。KaZaA や Skype などがこのモデルに相当する。

2.3 オーバーレイネットワーク

オーバーレイネットワークとは、既に存在する下層のネットワークの物理的構造の上位層において仮想的なネットワークを形成し、さまざまな機能追加を行うための技術である。

P2P ネットワークにおいては、IP ネットワークを基盤のネットワークとし、その上位層にオーバーレイネットワークとして、各ピアが通信リンクで接続された P2P ネットワークが構築される。P2P ネットワークのピア間は、通信リンクが IP ネットワークの上位層で仮想的に構成され、IP ネットワークとは別のネットワークトポロジーを持ったネットワークを定義している。

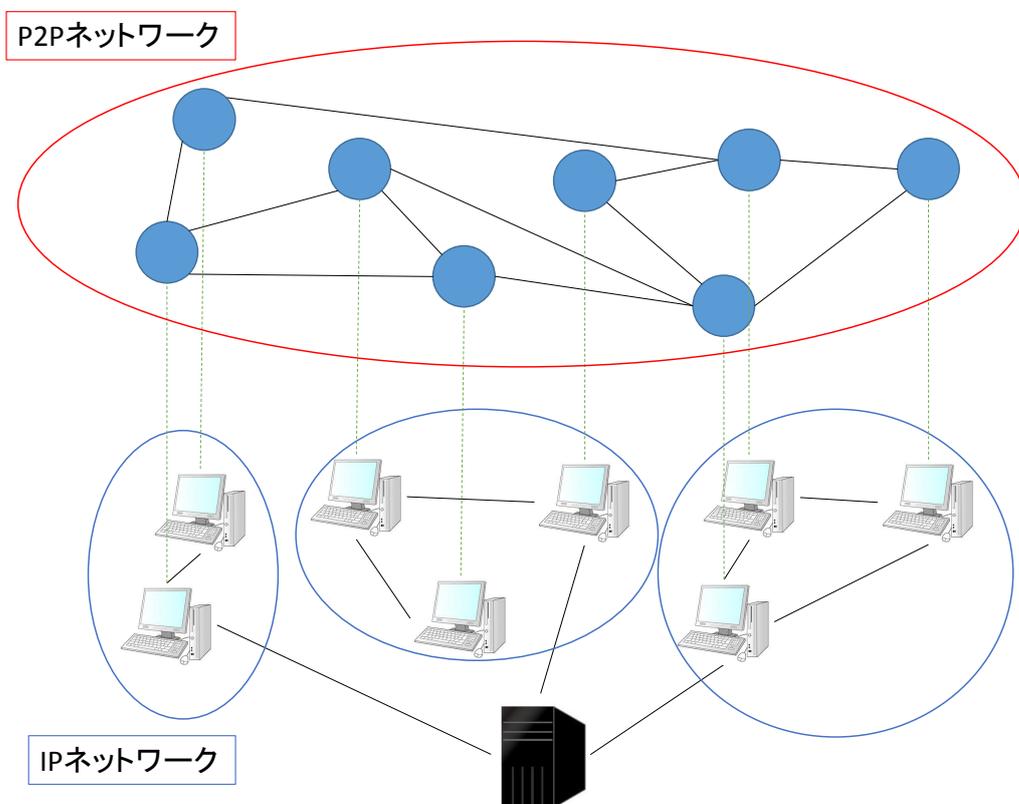


図 5.オーバーレイネットワーク概念図

オーバーレイネットワークは、既存のネットワークを基盤としてそのまま利用できる点や、柔軟で障害に強いネットワーク・サービスを必要に応じて構築することが可能である点などから、注目されているネットワーク技術である。

2.3.1 構造型オーバーレイ

構造型オーバーレイとはオーバーレイネットワークの構成時のルールが一定の数学的ルールに従うことを意味する。オーバーレイネットワークの構成が数学的なルールに基づくことを仮定したアルゴリズムを活用することにより、網羅的な情報探索を実現する。

構造型オーバーレイの特徴は、100%に近い探索成功率と非常に高い探索効率である。オーバーレイネットワークに格納されているすべての情報に対する探索が原則的に可能である。

構造型オーバーレイの代表例として分散ハッシュテーブル (DHT : Distributed Hash Table) が挙げられる。ハッシュテーブルとはキー (Key)、キーのハッシュ値、バリュー (Value) の3つを関連付けた表である。キーを用いて検索を行うが、キーを対応するハッシュ関数にて計算し、得られたハッシュ値を基にバリューが得られる。分散ハッシュテーブルは、ハッシュテーブルを複数のノードで分散管理する技術である。構造型オーバーレイネットワークはスケーラビリティであり、ピアの参加・離脱といったネットワークの動的な変化に対応可能である。だが、ハッシュ関数を用いた値からファイルを検索するために、完全一致検索でなければならない。

2.3.2 非構造型オーバーレイ

非構造型オーバーレイとは構造型オーバーレイとは異なり、オーバーレイネットワーク構成時にノード間の隣接関係や各ノードに格納する情報などが一定の数学的ルールに従わず、オーバーレイネットワークへの参加順などを基にランダムに近いルールで構成される。ネットワーク構成の自由度が高い反面、ファイルの検索効率は構造化オーバーレイに劣る。ファイルがノードと関連付けられていないため、検索手法の一つとして、ネットワーク内のノードに順に問い合わせしていくフラッディングとばれる手法が挙げられる。この手法は繰り返しクエリをコピーするためにネットワークトラフィックを圧迫する可能性がある。フラッディングではクエリに TTL (Time To Live) を用いて冗長なクエリを抑制するが、TTL が大きいと冗長なクエリが増加してしまい、TTL を小さくするとネットワークをくまなく探索することができず、コンテンツ入手失敗率が低下する。そのため、非構造型オーバーレイでは検索の網羅性を十分に保つ検索手法が必要とされる。

2.4 関連研究

非構造型 P2P ネットワークではネットワークへの加入・離脱が自由に行われる反面、コンテンツ検索を効率的に行うことが難しい。例えば Gnutella では、コンテンツ検索するためのクエリを隣接ノードへブロードキャストすることを繰り返すフラッディングによりコンテンツ検索を実現していた。フラッディングではクエリが冗長に複製・送信が行なわれることから、ネットワークへの負荷が懸念される。そのため、クエリの生存時間 (TTL) に制限を設け、段階的に制限を緩めながらフラッディングを繰り返す *iterativedeepening*[4] と呼ばれる手法へと改良された。また、Jiang らはネットワーク内にツリー構造を持つオーバーレイを構成し、そのツリーに沿ってクエリをブロードキャストするフラッディングを改良した *LightFlood* を提案している[5]。フラッディングのように全ての隣接ノードへクエリをブロードキャストするのではなく、隣接ノードの中からランダムに一つ選択しクエリの転送を行なうことを繰り返しコンテンツの検索を行うランダムウォークベース検索が提案されている[3][6]。Lv らは *k-random walk* と呼ばれる複数個のメッセージを同時にネットワーク内でランダムウォークさせる手法を提案している[6]。Adamic らは、既訪問のノードを避けながら、高次数ノードを優先的に訪れるランダムウォークを提案している[3]。塩田らは各ノードが自分を頂点とする TTL 制限付き最短路木を保持し、その最短路木に沿ってクエリを転送するネットワークブロードキャストを提案した[7]。また、ランダムウォーク型検索手法として、自分から n ホップ以内に位置するノード集合の経路情報およびファイルインデックスを所持しておきコンテンツ検索のヒット率を向上させる手法や[8],[9]、クエリを受信した向きとはなるべく異なる向きへとクエリを転送する指向性ランダムウォーク検索を提案した[10]。低次数ノードの重要性に着目した研究として、Liu ら[11]の研究がある。彼らは、(ネットワークを制御する上で) 影響力の大きいノードが、いわゆる次数の大きいハブノードには少なく、むしろ低次数ノードに多いことを示している。豊泉らは、情報のより速やかなネットワーク内への拡散のためには、情報を受け取ったノードは低次数ノードに優先的にその情報を送信することが望ましいことを示している[12]。

第3章

検索手法

3.1 フラッディング

フラッディングとは、Gnutellaによって一般的となった技術である。かつて、イーサネットや無線、組織内経路制御などの狭い範囲のネットワークで使われてきたブロードキャストと呼ばれる技術をもとにしたものである。

ネットワーク上に存在する送信可能なノードに対して要求メッセージ（クエリ）を洪水のように流す手法である。クエリを受信したノードは、クエリが送られてきたノード以外のすべての隣接ノードに対してクエリを送信する。隣接ノードは受け取ったクエリを自らと隣接するノードに対してクエリを送信する。このようにクエリ送信を繰り返すことでファイル検索を行う。目的のファイルを保有するノードは、クエリの送信を行わず、クエリ・ヒット・パケットをクエリの送信元の隣接ノードに返す。クエリ・ヒット・パケットは、クエリが送信されてきた経路を遡り、発信元である検索開

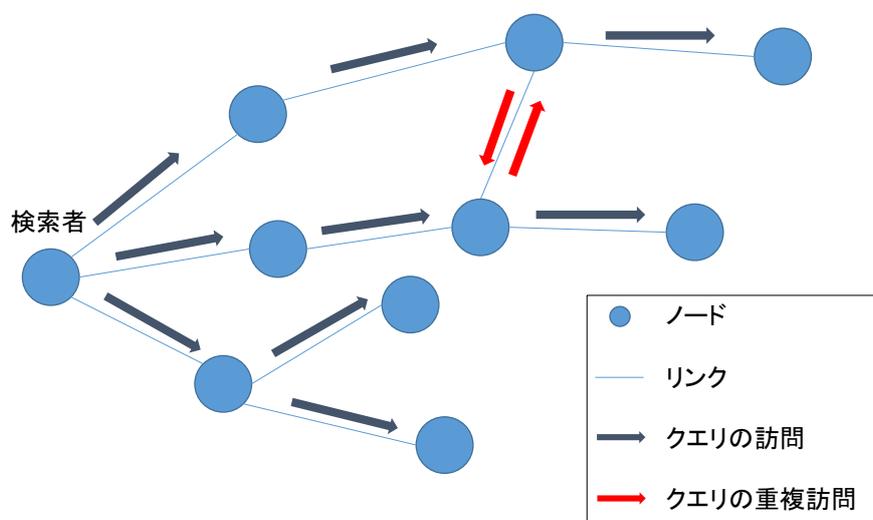


図 6. フラッディング概念図

始ノードへと転送され、該当するファイルの保有ノードの情報を手に入れることができる。

クエリは経路情報に依存することなくノード間をくまなく流れていくために、ネットワークの末端まで早く到達することができる。しかし、クエリ送信を繰り返していくと、一つのノードに同一のクエリが重複して到達することになり、冗長なクエリによりネットワーク負荷が高まる。これを防ぐために、クエリには **TTL (Time To Live)** と呼ばれるフィールドでクエリの有効ステップ数が設定されている。TTL はピア間での転送ごとに 1 ずつ減算され、0 になったら破棄される仕組みによって到達範囲を制限している。

フラッディングはデータを所持しているピアで検索の処理を行うため、柔軟な検索が可能となる。しかし、TTL によりクエリが伝達される範囲が限られてしまい、ネットワークの末端まで検索されない可能性がある。

今、 N 個のノードと L 本のリンクからなるネットワークにおいてフラッディングを行う場合を考える。リンクは、双方向に伝達可能な無向グラフとする。ノード i のリンク数を l_i とすると、各ノードは送信元以外のすべての隣接ノードへクエリを送信するため、そのクエリ数 Q_i は

$$Q_i = l_i - 1$$

で与えられる。すべてのノードが同様にクエリをそうしんするため、発生するクエリの総数 Q は

$$\begin{aligned} Q &= \sum_{i=1}^N (l_i - 1) + 1 \\ &= \sum_{i=1}^N l_i - \sum_{i=1}^N 1 + 1 \\ &= 2L - N + 1 \end{aligned}$$

となる。定数項として 1 が与えられるのは、始点となる発信元ノードは隣接するすべてのノードに対してクエリを送信するためである。また、ネットワーク内のすべてのノードをクエリが訪問するために必要な最小のクエリ数 Q_{min} は、すべてのノードが一つだけクエリを受信した場合であり、

$$Q_{min} = N - 1$$

で与えられる。以上から、フラッディングにより発生する冗長なクエリ数は

$$\begin{aligned} Q - Q_{min} &= (2L - N + 1) - (N - 1) \\ &= 2(L - N + 1) \end{aligned}$$

となり、ノード数とリンク数の関係式で表される。

3.2 有効なコンテンツ検索の考察

本節では低次数優先型の優位性を理論的に考察する。ただし、ランダムウォーク検索を理論的に厳密に取り扱うことは困難であるので、ランダムウォーク検索の代わりに、ネットワーク内のいずれか1つのノードを選んで訪問する試行を繰り返す検索法について考察する（ランダムウォーク検索では隣接ノードしか訪問先として選べないが、この検索法ではネットワーク内の任意のノードを選べるとする）。ノードの総数を N とし、ノード i が訪問のために選択される確率は p_i に等しいとする（重複訪問を許す）。 n 回の試行の結果、訪問済みのノード数を $S(n)$ で表す。 $l_i(n)$ を以下により定義する。

$$l_i(n) \stackrel{\text{def}}{=} 1(\text{n 回の試行終了後、ノード } i \text{ が訪問済みである}).$$

ここで $1(A)$ は A が真であれば 1、偽であれば 0 を取る indicator 関数である。このとき

$$S(n) = \sum_{i=1}^N l_i(n). \quad (1)$$

$l_i(n)$ は以下のように書ける。

$$l_i(n) = \sum_{k=1}^n 1(\text{ノード } i \text{ が } k \text{ 回目の試行で初めて訪問される}).$$

したがって、

$$\begin{aligned} E[l_i(n)] &= \sum_{k=1}^n P(\text{ノード } i \text{ が } k \text{ 回目の試行で初めて訪問される}). \\ &= \sum_{k=1}^n p_i(1-p_i)^{k-1} \\ &= 1 - p_i(1-p_i)^n. \end{aligned}$$

これを (2) に代入して

$$E[S(n)] = \sum_{i=1}^N E[l_i(n)] = N - \sum_{i=1}^N (1-p_i)^n$$

コンテンツがネットワーク内のいずれか一つのノードに存在する場合、 n 回の試行を繰り返しても所望のコンテンツを入手できない確率 $p_i(n; p_1, \dots, p_n)$ は

$$p_i(n; p_1, \dots, p_n) = \frac{N - E[S(n)]}{N} = \frac{1}{N} \sum_{i=1}^N (1 - p_i)^n. \quad (2)$$

同様の考察により，コンテンツがネットワーク内のいずれか一つのノードに存在する場合，条件に合うコンテンツを入手するまでにかかる時間（平均試行回数） $T(p_1, \dots, p_n)$ は以下で与えられることがわかる．

$$T(p_1, \dots, p_n) = \frac{1}{N} \sum_{i=1}^N \frac{1}{p_i}. \quad (3)$$

定理 1. $p_f(n; p_1, \dots, p_n)$ は $p_1 = \dots = p_n = \frac{1}{N}$ のときに最小値 $(1 - \frac{1}{N})^n$ をとる．

Proof. $f(x) \stackrel{\text{def}}{=} (1 - x)^n$ は凸関数であることから，式(2)より，

$$\begin{aligned} p_i(n; p_1, \dots, p_n) &= \frac{1}{N} \sum_{i=1}^N f(p_i) \geq f\left(\frac{1}{N} \sum_{i=1}^N p_i\right) \\ &= f\left(\frac{1}{N}\right) = \left(1 - \frac{1}{N}\right)^n \\ &= p_f\left(n; \frac{1}{N}, \dots, \frac{1}{N}\right). \end{aligned}$$

以上より証明された．

定理 2. $T(p_1, \dots, p_n)$ は $p_1 = \dots = p_n = \frac{1}{N}$ のときに最小値 N をとる．

Proof. $g(x) \stackrel{\text{def}}{=} x^{-1}$ は凸関数であることから，式(3)より，

$$\begin{aligned} T(p_1, \dots, p_n) &= \frac{1}{N} \sum_{i=1}^N g(p_i) \geq g\left(\frac{1}{N} \sum_{i=1}^N p_i\right) \\ &= g\left(\frac{1}{N}\right) = N \\ &= T_v\left(\frac{1}{N}, \dots, \frac{1}{N}\right). \end{aligned}$$

以上より証明された．

定理 1 および定理 2 は，ネットワーク内の各ノードを均等に訪問することが，入手失敗確率および入手時間最小化の点で最良戦略であることを示している．これら定理は，あくまでネットワーク内の任意のノードを繰り返し訪問する思考実験に関する結果であるが，ランダムウォーク検索においても同様の結果が成立することが十分に期待できる．

ランダムウォーク検索とは、クエリが隣接ノードから一つを訪問することを繰り返して、条件に合うコンテンツを検索する手法である。本論文では、クエリが隣接ノードのいずれか一つを等確率で選んで移動するランダムウォーク検索を「基本ランダムウォーク検索」と呼ぶ。基本ランダムウォーク検索では、ノードをクエリが訪問する確率はそのノードの次数に比例することが知られている。つまり、基本ランダムウォーク検索ではクエリが高次数ノードを高頻度に訪問する傾向がある。このため、条件に合うコンテンツがたまたま低次数ノードに存在すると、クエリがなかなかコンテンツを所持するノードに到達できず、コンテンツ発見までに時間がかかったり、検索に失敗したりする危険性がある。定理1より、コンテンツ入手失敗率を下げるには、ネットワーク内の各ノードを均等に訪問することが有効であると示された。そこで、低次数ノードを優先することで、基本ランダムウォーク検索が有する高次数ノードへのバイアスが緩和され、コンテンツ入手失敗率を低下させる効果があると予想した。各ノードが、その隣接ノードの次数情報を所持していることを前提として、クエリが隣接ノードの中から低い次数を有するノードを選んで移動する手法を「低次数優先型ランダムウォーク検索」と呼ぶ。

3.3 ノード次数を考慮したランダムウォーク検索

ここでは、提案手法において、隣接ノードの次数情報を基にノードの選択確率の決定する方法を説明する。

ノード i の隣接ノード群を $N(i)$ 、ノード $j \in N(i)$ の次数を d_j とする。本稿で考察するノード次数を考慮したランダムウォーク検索では、ノード i に存在するクエリはノード $j \in N(i)$ に以下の確率で移動することとする。

$$p_{i \rightarrow j} = \frac{d_j^c}{\sum_{k \in N(i)} d_k^c} \quad (4)$$

ここで c は優先度を決定するシステムパラメタとする。例えば、 $c = -1$ は次数に反比例する確率で移動先ノードを選択する手法に相当し、 $c \rightarrow -\infty$ の極限においては、最小次数の隣接ノードを選択する手法と等価となる。同様に、 $c = 1$ では、次数に比例する確率で移動先ノードを選択する手法に相当し、 $c \rightarrow \infty$ の極限においては、最大次数の隣接ノードを選択する手法と等価となる。また、 $c = 0$ では、次数に依存せず、すべての隣接ノードを当確率で移動先に選択する基本ランダムウォーク検索と等価となる。

以上のように、 $c > 0$ では、高次数ノードを優先し、 $c < 0$ では、低次数ノードを優先する。 $|c|$ が大きくなるほど次数による優先度が大きくなる。

3.4 既訪問ノード回避アルゴリズム

3.4.1 ノード記録型

各ノードはクエリの移動元の隣接ノード，および移動先の隣接ノードを「既訪問ノード」として記憶しておく．クエリが移動先ノードを選ぶ際には既訪問ノードは対象から除外し，式 (4) の計算の際にも $N(i)$ から既訪問ノードは除外しておく．本稿ではこれを「ノード記録型既訪問ノード回避アルゴリズム」と呼ぶ．ただし，全ての隣接ノードが既訪問ノードである場合には，隣接ノードの中からランダムに移動先を選ぶものとする．以下にノード記録型の動作例を示す．

例)

Step 1. ノード a をクエリが訪問する．

Step 2. ノード a からノード b にクエリが移動． ノード a はノード b を既訪問ノードとして記録． ノード b はノード a を既訪問ノードとして記録．

Step 3. ノード b からノード c にクエリが移動． ノード b はノード c を既訪問ノードとして記録． ノード c はノード b を既訪問ノードとして記録

クエリが以上のように移動した場合，ノード c において次の移動先ノードを選択する際に，既訪問であるはずのノード a がノード c には既訪問ノードとして記録されていないため移動先候補に含まれてしまい，重複訪問する可能性がある．

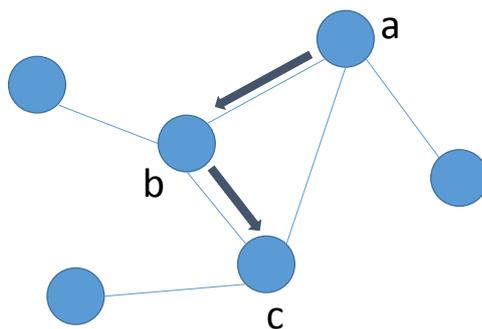


図 7. ネットワーク概念図

3.4.2 クエリ記録型

本研究では、さらに既訪問ノード情報をクエリに記録する手法について検討した。この手法では、各ノードに付与した固有の ID をクエリが記録する。隣接ノードの内、既訪問ノード ID と一致する ID をもつノードを移動先候補から除外する。本稿ではこれを「クエリ記録型既訪問ノード回避アルゴリズム」と呼ぶ。以下に、クエリ記録型の動作例を示す。

例)

Step 1. ノード a をクエリが訪問する。クエリはノード a を既訪問ノードと記録する。

Step 2. ノード a からノード b にクエリが移動。クエリはノード b を既訪問ノードとして記録。

Step 3. ノード b からノード c にクエリが移動。クエリはノード c を既訪問ノードとして記録。

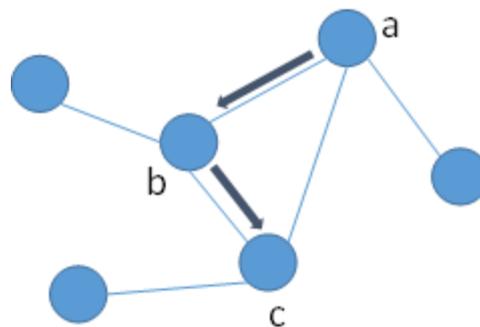


図 8. ネットワーク概念図

このとき、ノード記録型ではノード a を重複訪問してしまう可能性があったが、クエリ記録型ではこの問題点を解消できるため、重複訪問回数を削減し、より効率的な検索が可能であると予想できる。

3.4.3 複数クエリを用いた場合

複数クエリに用いたランダムウォークにおける既訪問ノード回避アルゴリズムの適用方法について説明する。

ノード記録型では、ノードに記録した情報をクエリが使用するため、自然と各クエリ間で情報の共有ができる。考えられる負荷は、各ノードに隣接ノードの既訪問情報を記録するのみなので、極めて小さくなる。

それに対してクエリ記録型では、各クエリが独自に既訪問ノード情報を記録している。各クエリが所持する既訪問ノード情報を共有し、すべてのクエリが所持する既訪問ノード情報を用いる方法を、「クエリ記録（共有あり）」と呼ぶ。既訪問ノード情報を共有せず、各クエリが独自の情報のみを用いる方法を、「クエリ記録（共有なし）」と呼ぶ。クエリ記録共有なしの場合、単一クエリの場合と同じく、各クエリが独自の既訪問ノード情報だけを管理するため、その負荷は極めて小さくなる。

また、クエリ記録（共有あり）の場合は、図のように、各クエリの既訪問ノード情報をクラウドのデータベースで管理する方法が想定できる。この場合も負荷は極めて小さくなる。

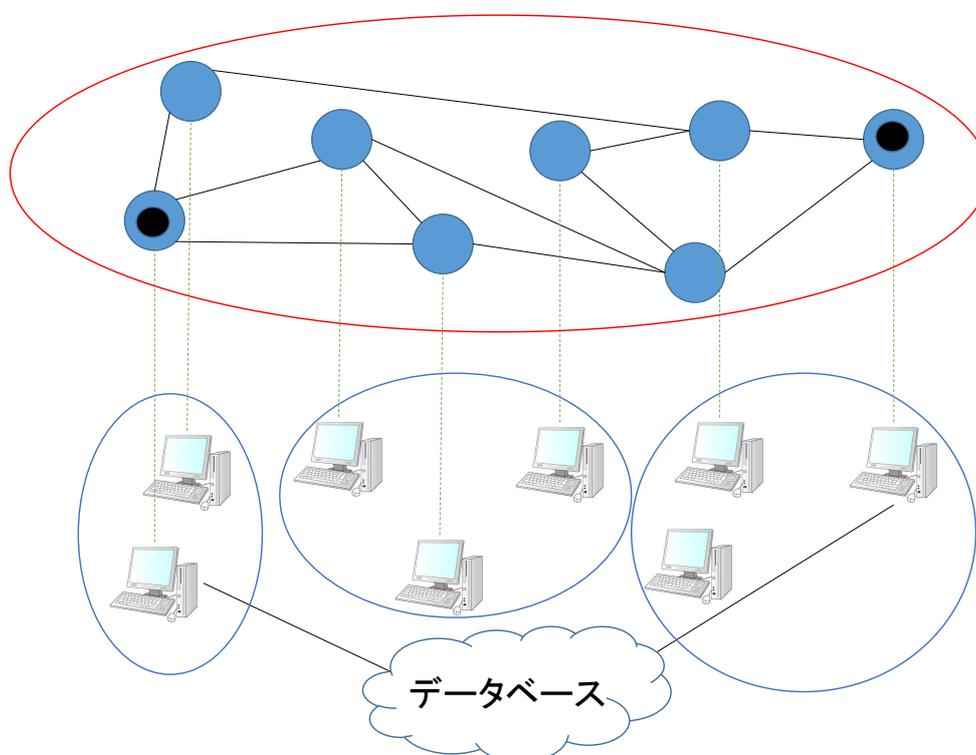


図9. クエリ記録（共有あり）概念図

第4章

シミュレーション評価：単一クエリ

4.1 評価条件

ノード次数を考慮したランダムウォークを用いてコンテンツ検索を行った場合の検索効率をシミュレーションにより評価した。シミュレーションに使用するネットワークは、次数分布がべき乗則に従うグラフ（以下、べき乗則グラフ）、Gnutella ネットワーク、Facebook ネットワークの3つである（表1）。べき乗則グラフは、べき指数が2に従うように次数列を生成後、Havel-Hakimi アルゴリズムにより次数列を再現するグラフを生成し[13]、さらにリンクの張り替えを行って次数相関を極力取り除いたものを用いた。ノード数は5000とした。Gnutella および Facebook ネットワークはSTANFORD UNIVERSITYにて公開されているGnutella および Facebook のデータセット[14]の一部である。Gnutella ネットワークは有向グラフであるが、シミュレーションでは全てのリンクが双方向に移動可能な無向グラフとして扱う。またGnutella ネットワークでは、一部ノードが閉鎖的なネットワークを形成しているケースも存在したが、これらは非常に規模が小さいことから評価の対象から除外した。レアコンテンツを想定し、コンテンツはネットワーク内のいずれか一つのノードのみが保持するものとした。シミュレーションは、ノード次数を考慮したランダムウォーク検索で低次数ノード、高次数ノードの優先度をあらわすシステムパラメタ c を $-\infty \sim \infty$ に変えて評価した。実際には $c = -\infty, \infty$ をそれぞれ最低次数型、最高次数型とし、 $c = -4, 4, -3 \leq c \leq 3$ は0.25間隔で、コンテンツが未入手である確率（入手失敗率）を評価する。ステップ数（訪問回数：重複訪問も含める）をノード数の0.5, 1, 1.5, 2倍とした。

ネットワーク	ノード数	リンク数	平均次数
べき乗則グラフ	5000	15915	3.183
Gnutellaネットワーク	8104	26008	3.209279368
Facebookネットワーク	4039	88234	21.84550631

表 1. シミュレーションに使用したネットワーク

4.2 既訪問ノード回避なし

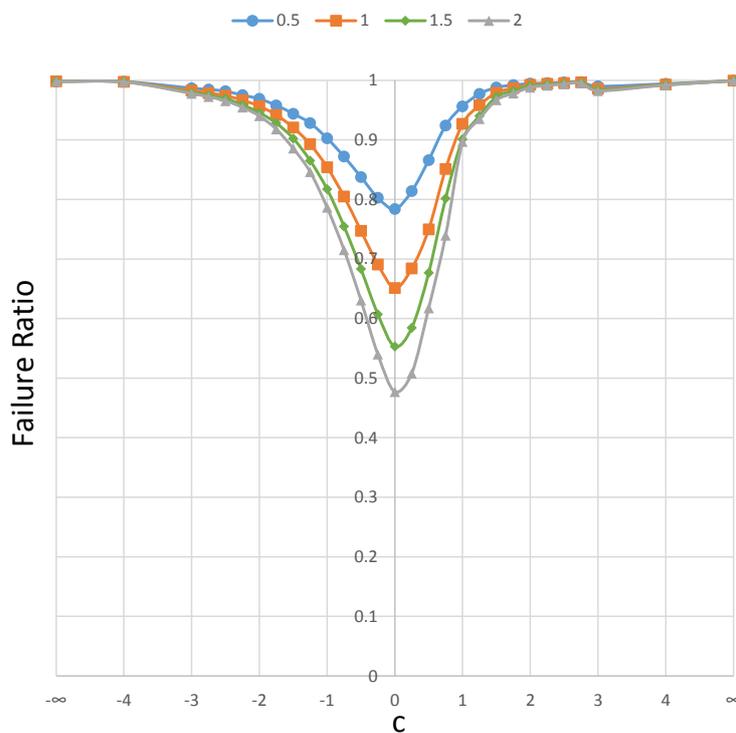


図 10. コンテンツ入手失敗率 (べき乗則グラフ)

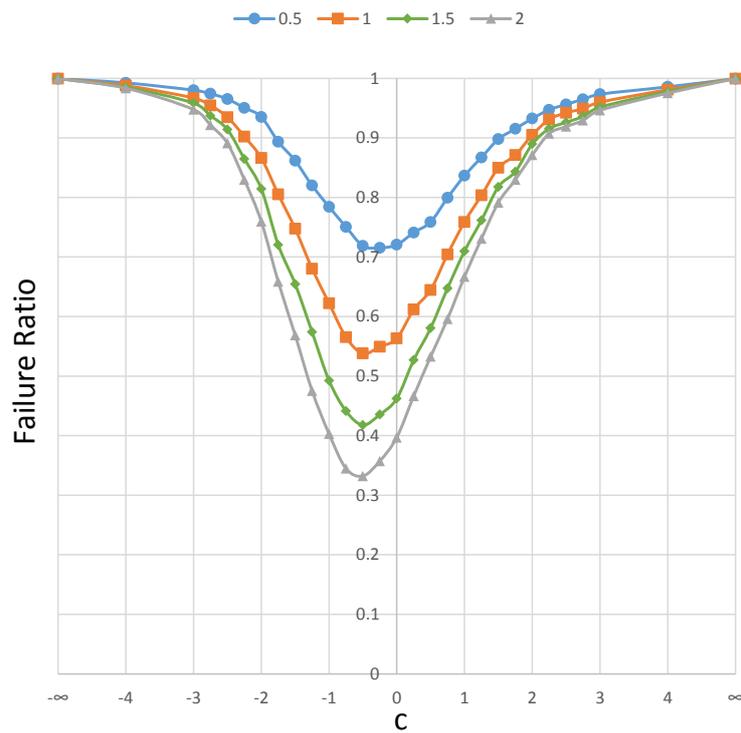


図 11. コンテンツ入手失敗率 (Gnutella ネットワーク)

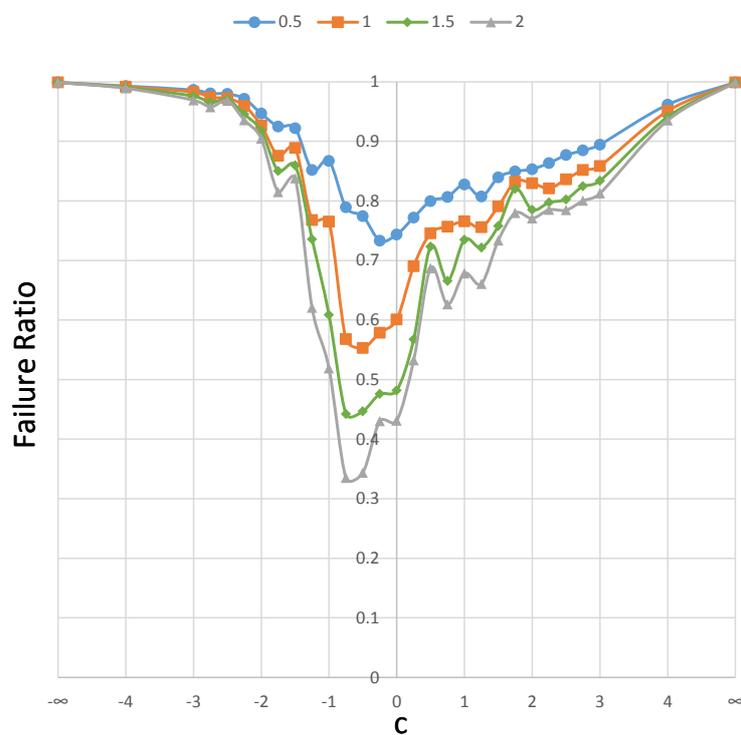


図 12. コンテンツ入手失敗率 (Facebook ネットワーク)

まず，既訪問ノード回避アルゴリズムを用いないときに c （次数を考慮したノードの優先度を表すシステムパラメタ）を変えて，コンテンツが未入手である確率（入手失敗率）を評価する．ステップ数は，ノード数の 0.5, 1, 1.5, 2 倍についてプロットした．結果を図 10. (べき乗則グラフ), 図 11. (Gnutella ネットワーク), 図 12. (Facebook) に示す．

べき乗則グラフ (図 10) では，どのステップ数においても $c = 0$ (基本ランダムウォーク) のときにコンテンツ入手失敗率が最小となった．また $c \neq 0$ では， $|c|$ が小さい領域において，良い性能を示した．

Gnutella ネットワーク (図 11), Facebook ネットワーク (図 12) では， $|c|$ が小さい領域で，低次数優先型が良い性能を示した．

3つのネットワークに共通して， $|c|$ が大きくなると，コンテンツ入手失敗率が大きくなる傾向が見られた．これは，ノード次数を考慮したランダムウォークの $|c|$ が大きくなるとともに，低次数優先型 ($c < 0$) では低次数ノード，高次数優先型 ($c > 0$) では高次数ノードへの優先度が大きくなるという特徴から，ある二つのノードを繰り返し訪問してしまう，重複訪問のループに陥ってしまい，抜け出すことができなくなる可能性が高まるからであると考えられる．

既訪問ノードを回避しない場合，ランダムウォークでは検索効率に限界があるため，既訪問ノード回避アルゴリズムを適用する必要があることがわかった．

4.3 ノード記録型既訪問ノード回避アルゴリズム

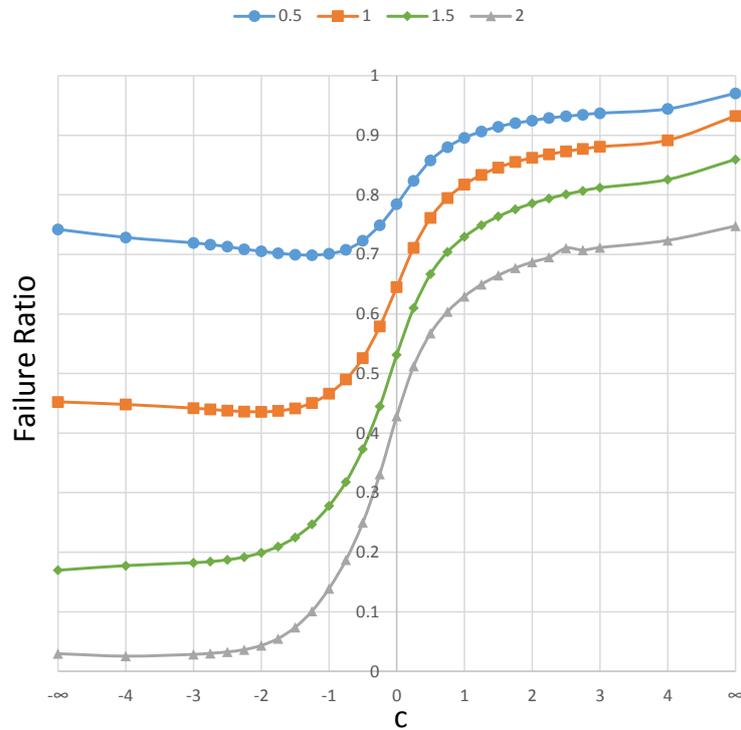


図 13. コンテンツ入手失敗率（べき乗則グラフ）

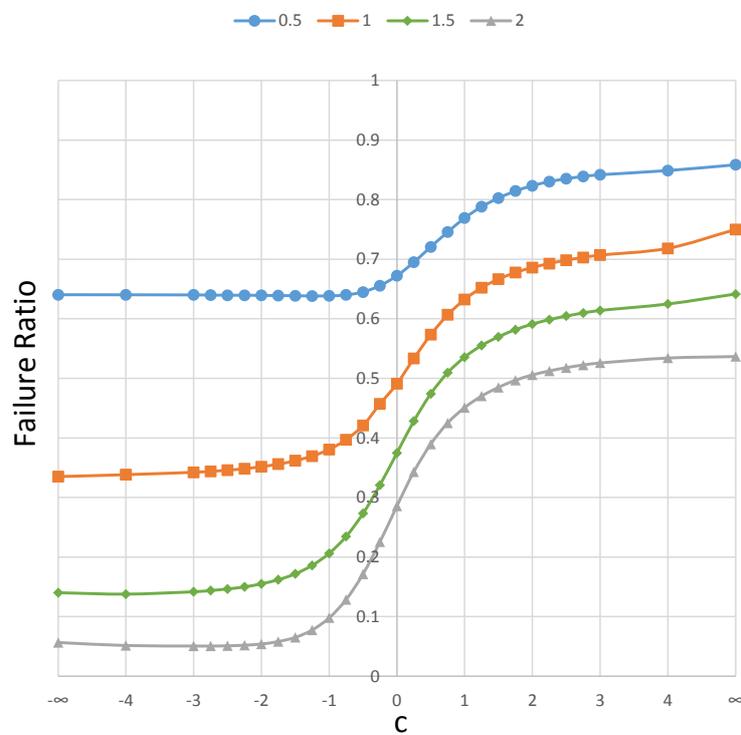


図 14. コンテンツ入手失敗率（Gnutella ネットワーク）

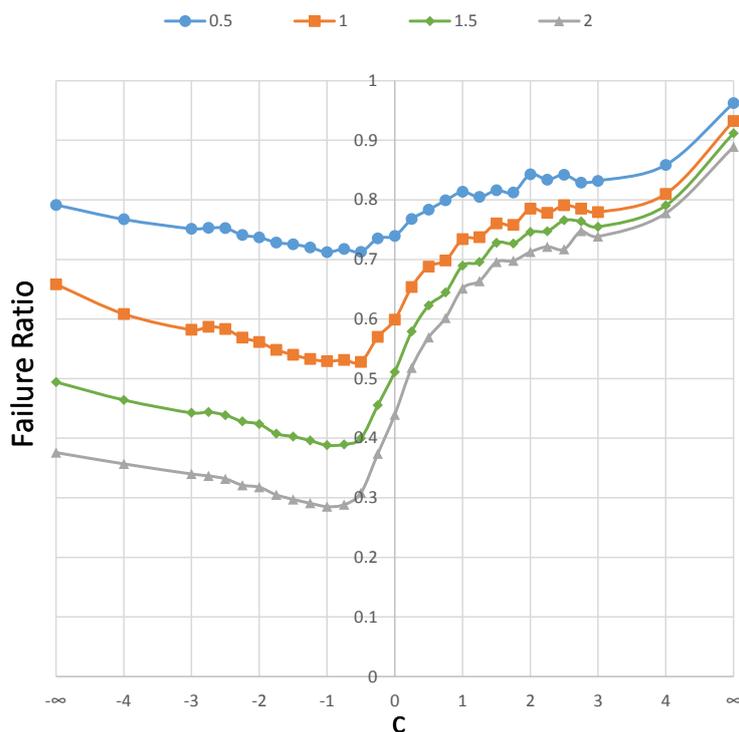


図 15. コンテンツ入手失敗率(Facebook ネットワーク)

ノード記録型既訪問ノード回避アルゴリズムを用いたときに c を変えて、コンテンツ入手失敗率を評価する。ステップ数は、ノード数の 0.5, 1, 1.5, 2 倍についてプロットした。結果を図 13. (べき乗則グラフ), 図 14. (Gnutella ネットワーク), 図 15. (Facebook) に示す。

べき乗則グラフ (図 13), Gnutella ネットワーク (図 14) では、低次数優先型が高次数優先型の入手失敗率を下回る結果となった。

Facebook ネットワーク (図 15) では、次数比例型 ($c = -1$) 付近で入手失敗率が最小になるが、 $-\infty$ に近づくと入手失敗率が大きくなる。これは、Facebook の平均次数が 21.846 と他のネットワークよりも大きいことに関係があると考えられる。ノード記録型既訪問ノード回避アルゴリズムでは、あるノードからは既訪問と記録されていても、他のノードから見ると既訪問と記録されておらず、重複訪問してしまうという問題を抱えている。Facebook では平均次数が大きいため、最低次数のノードの次数が他のネットワークと比較して大きくなると考えられる。そのため、 $c = -\infty$ で最低次数を必ず選ぶ方法では重複訪問回数は最低次数ノードの次数に比例するため、入手失敗率が大きくなると考えられる。

どのネットワークにおいても、ノード記録型は、 c が減少し、低次数優先型になると検索効率があがる。これは、低次数優先型にすることにより、クエリ訪問確率の高次数ノードバイアスを緩和できるためである。

4.4 クエリ記録型既訪問ノード回避アルゴリズム

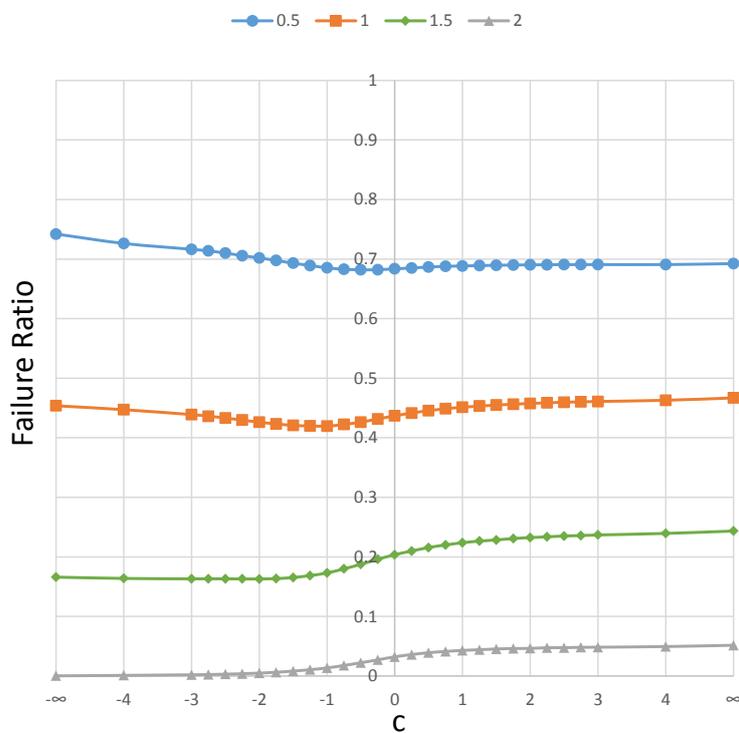


図 16. コンテンツ入手失敗率 (べき乗則グラフ)

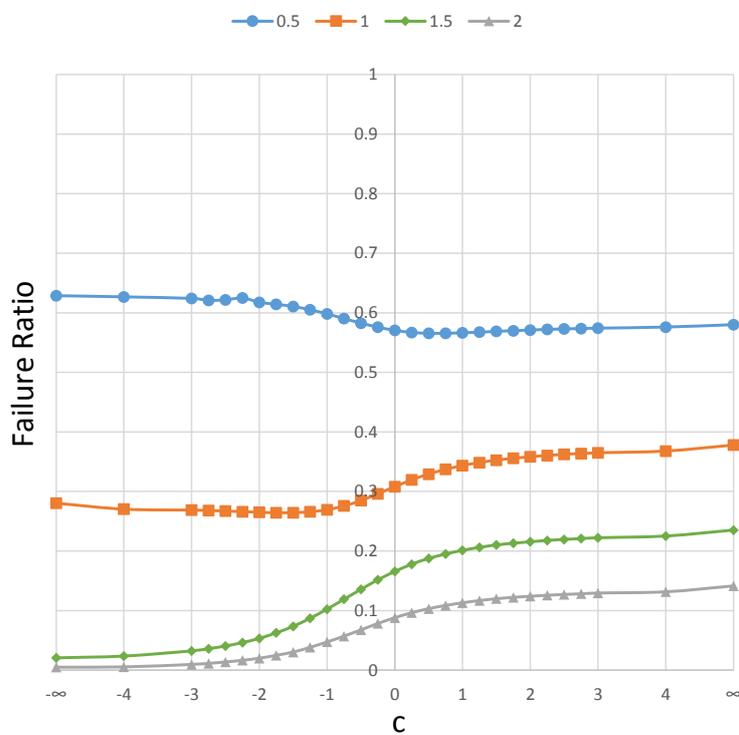


図 17. コンテンツ入手失敗率 (Gnutella ネットワーク)

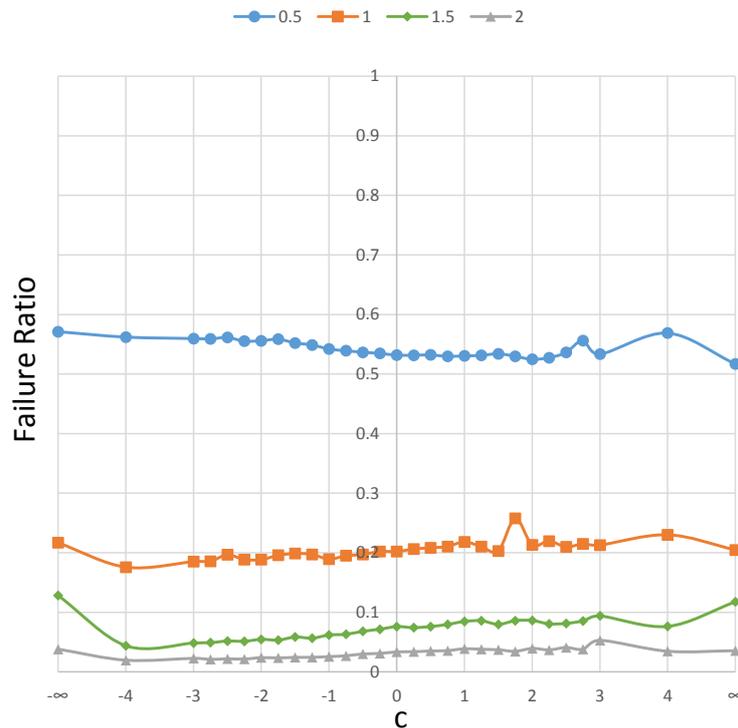


図 18. コンテンツ入手失敗率 (Facebook ネットワーク)

クエリ記録型既訪問ノード回避アルゴリズムを用いたときに c を変えて、コンテンツ入手失敗率を評価する。ステップ数は、ノード数の 0.5, 1, 1.5, 2 倍についてプロットした。結果を図 16 (べき乗則グラフ), 図 17 (Gnutella ネットワーク), 図 18 (Facebook) に示す。

どのネットワークにおいても、ステップ数がノード数の 0.5 倍のとき、おおむね低次数優先型より高次数優先型の入手失敗率が小さくなる傾向が見られた。逆に、ステップ数がノード数の 1 倍以上になると、おおむね高次数優先領域より低次数優先領域の入手失敗率が小さくなる傾向が見られた。このことから、検索が進んでいない段階では、低次数ノードより高次数ノードは、すべての隣接ノードが既訪問であり重複訪問せざるを得ないケースが少ないため、高次数優先型が有利であると考えられる。また、検索が進むと、高次数ノードでも隣接ノードがすべて既訪問のケースが増加するため、低次数優先型が有利であると考えられる。

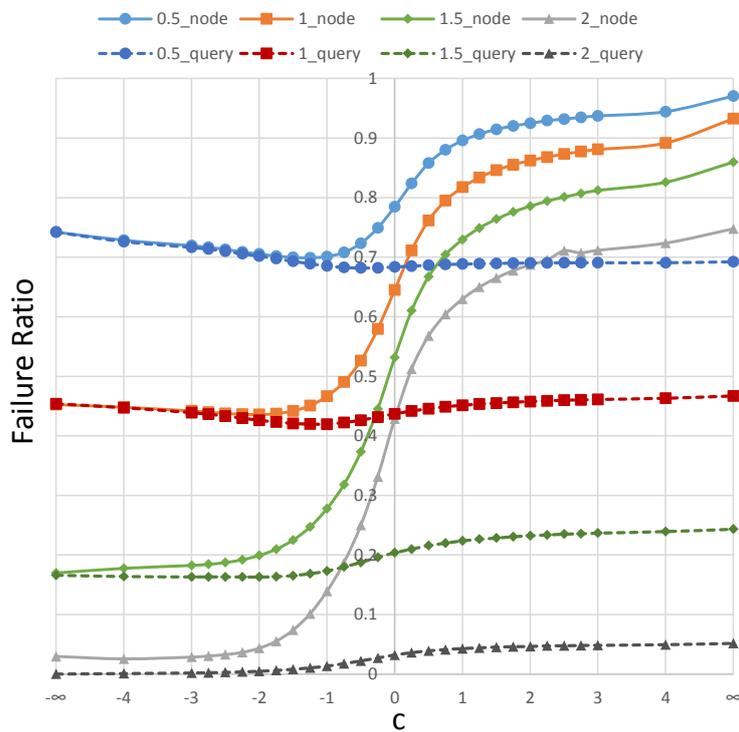


図 19. コンテンツ入手失敗率 (ベッキ乗則グラフ)

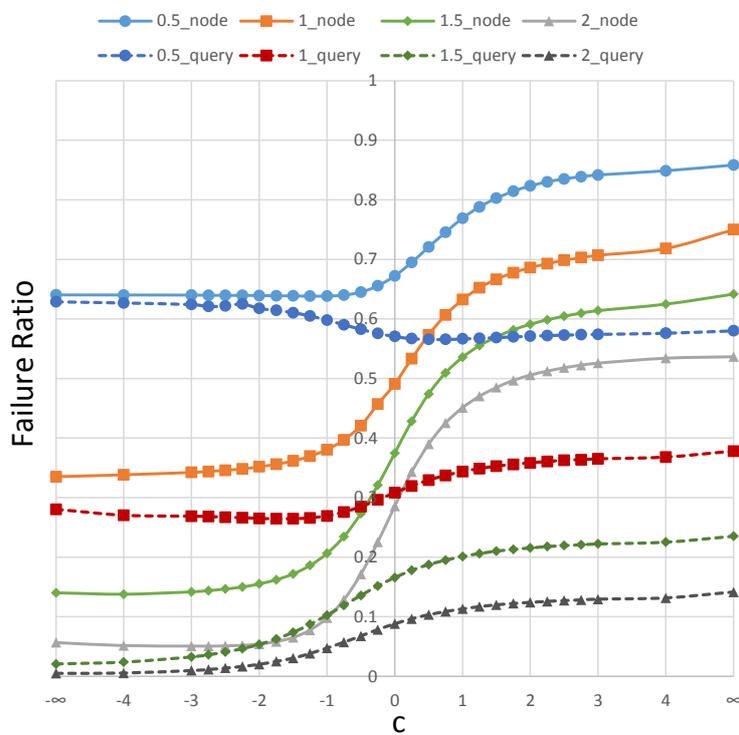


図 20. コンテンツ入手失敗率 (Gnutella)

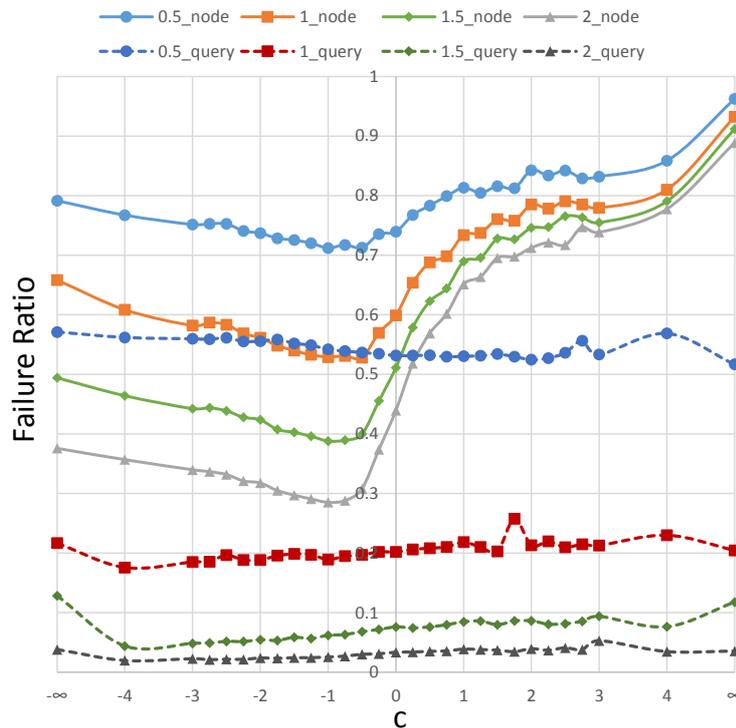


図 21. コンテンツ入手失敗率 (Facebook)

図 19 (べき乗則グラフ), 図 20 (Gnutella), 図 21 (Facebook) は 4.3 で示したノード記録型既訪問ノード回避アルゴリズムの結果と, 本章のクエリ記録型既訪問ノード回避アルゴリズムの結果を比較したものである. ここでは, 例えば 0.5_node はノード記録型でクエリのステップ数がノード数の 0.5 倍の回数に達した時の結果を示している.

クエリ記録型では, 特に高次数優先型において, ノード記録型と比較して大きく入手失敗率を減少させた. これは, より高次数のノードを訪問することが, 重複訪問回数削減効果が高いためであると考えられる.

Facebook ネットワーク (図 21) では, べき乗則グラフ, Gnutella ネットワークと比較して, クエリ記録型が大きく入手失敗率を低下させた. 各グラフの平均次数を比較すると, べき乗則グラフは 3.183, Gnutella ネットワークは 3.209 であるのに対して, Facebook ネットワークは 21.846 と他のネットワークと比較して平均次数が大きい. そのため, 1 ステップあたりのクエリ記録による重複訪問回数削減効果が他のネットワークよりも大きいと考えられる.

どのネットワークにおいてもノード記録型よりクエリ記録型の検索効率が高いことが示された.

第5章

シミュレーション評価：複数クエリ

5.1 評価条件

次に、ネットワーク内で複数クエリを同時にランダムウォークさせ、各既訪問ノード回避アルゴリズムにおける入手失敗率を評価する。

3.4.3 で示したクエリ記録型（共有あり）、クエリ記録型（共有なし）、ノード記録型の3つの方法を用いる。

また複数クエリを用いる場合、各クエリは初期位置としてネットワーク内のノードをランダムに選択する。

5.2 複数クエリ

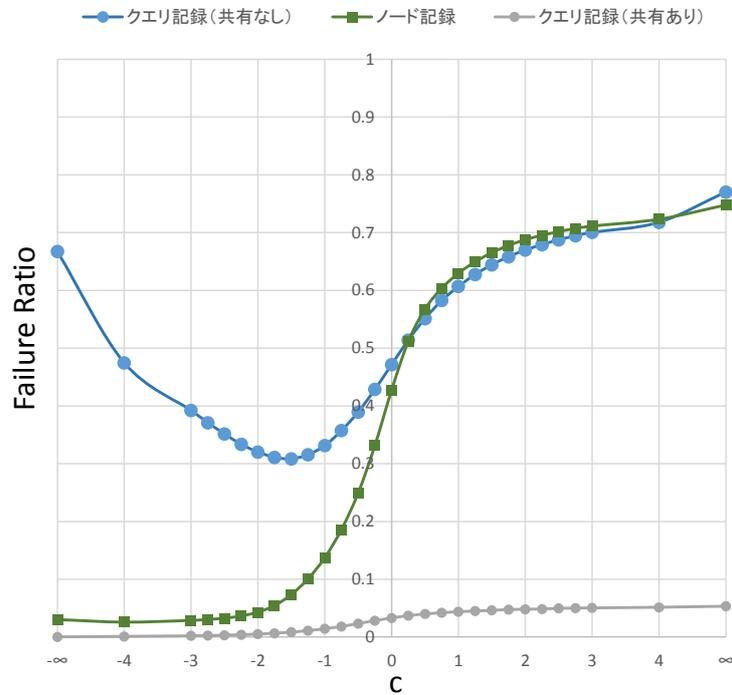


図 22. コンテンツ入手失敗率 (ベッキ乗則グラフ)

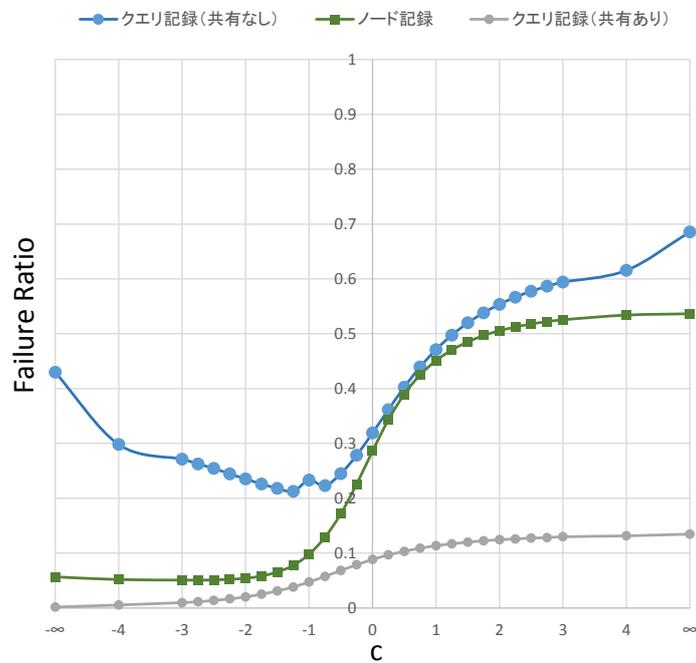


図 23. コンテンツ入手失敗率 (Gnutella ネットワーク)

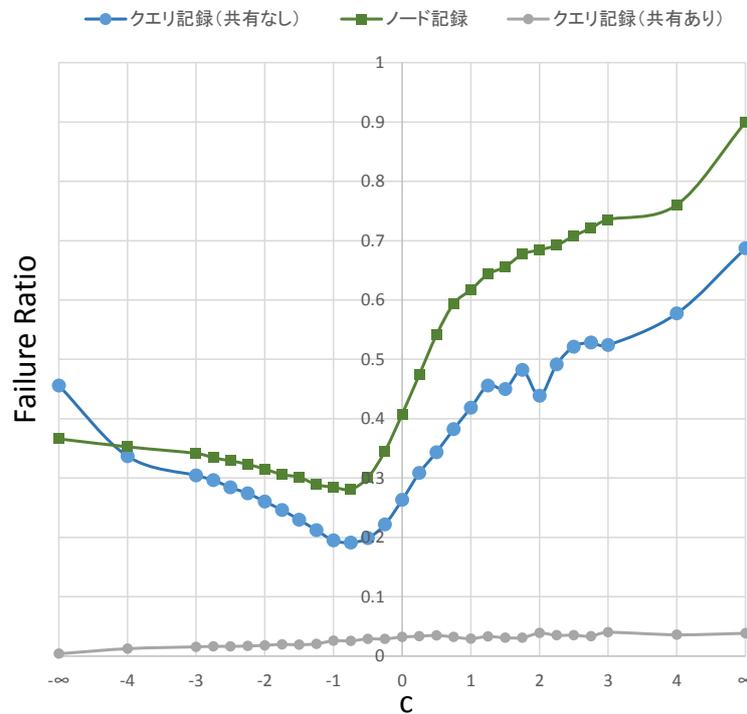


図 24. コンテンツ入手失敗率 (Facebook ネットワーク)

べき乗則グラフ (図 22), Gnutella ネットワーク (図 23) では高次数優先型では, クエリ記録型 (共有なし) とノード記録型で結果がおおむね一致した. 低次数優先型ではクエリ記録型 (共有なし) よりノード記録型の入手失敗率が小さくなった. しかし, Facebook ネットワーク (図 24) ではノード記録よりクエリ記録 (共有なし) の入手失敗率が小さくなる傾向が見られた. これは, Facebook の平均次数がべき乗則グラフ, Gnutella ネットワークと比較して大きいことによって, 1 ステップあたりの重複訪問確率削減効果が大きいためである. べき乗則グラフ, Gnutella ネットワークではクエリ記録した既訪問情報を共有しない分, ノード記録型が有利となるが, Facebook では, 1 つのクエリの 1 ステップあたりの重複訪問削減効果が, 共有しないことによるデメリットを打ち消すことによって, クエリ記録型 (共有なし) が有利となっていると考えられる.

クエリ記録 (共有なし) の検索効率は, クエリ記録 (共有あり) と比較して大きく劣化した. このことから, 検索効率を担保するためには, 既訪問ノード情報を共有する必要があることがわかる.

5.3 クエリ数による変化

クエリ数を増加させたときの各既訪問ノード回避アルゴリズムにおける入手失敗率を示す。クエリ 1 つの 1 ステップあたりの検索効率を評価するために、クエリ 1 個の N ステップの結果と、クエリ K 個の $\frac{N}{K}$ ステップの結果を比較する。ここでは、クエリ 1 個でノード数の 2 倍のステップ数、クエリ 40 個でノード数の $\frac{1}{20}$ 倍のステップ数、80 個でノード数の $\frac{1}{40}$ 倍のステップ数を比較した。

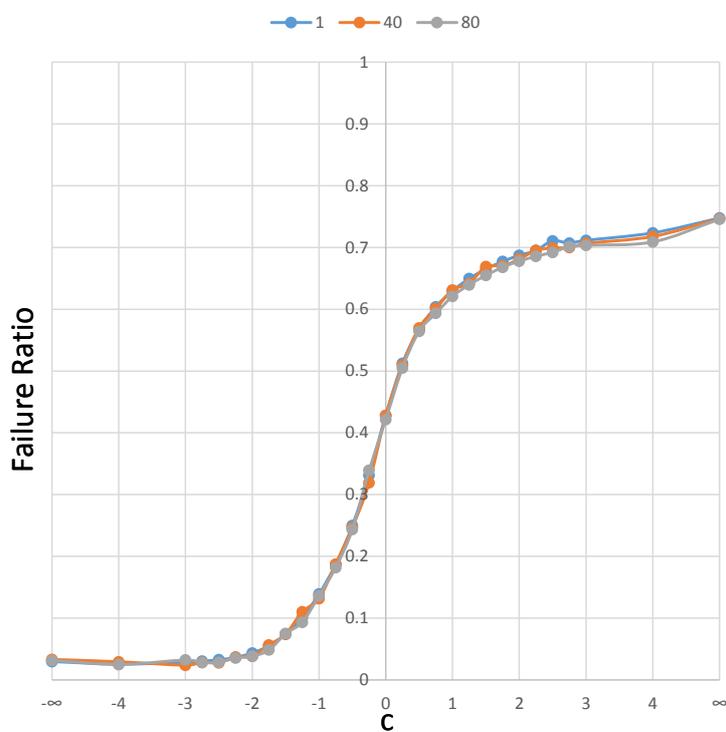


図 25. コンテンツ入手失敗率 (べき乗則: ノード記録)

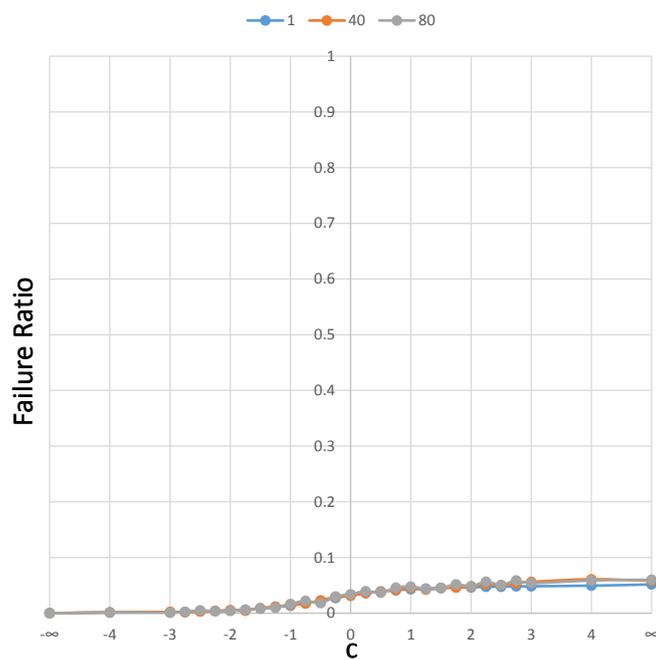


図 26. コンテンツ入手失敗率（べき乗則：クエリ記録（共有あり））

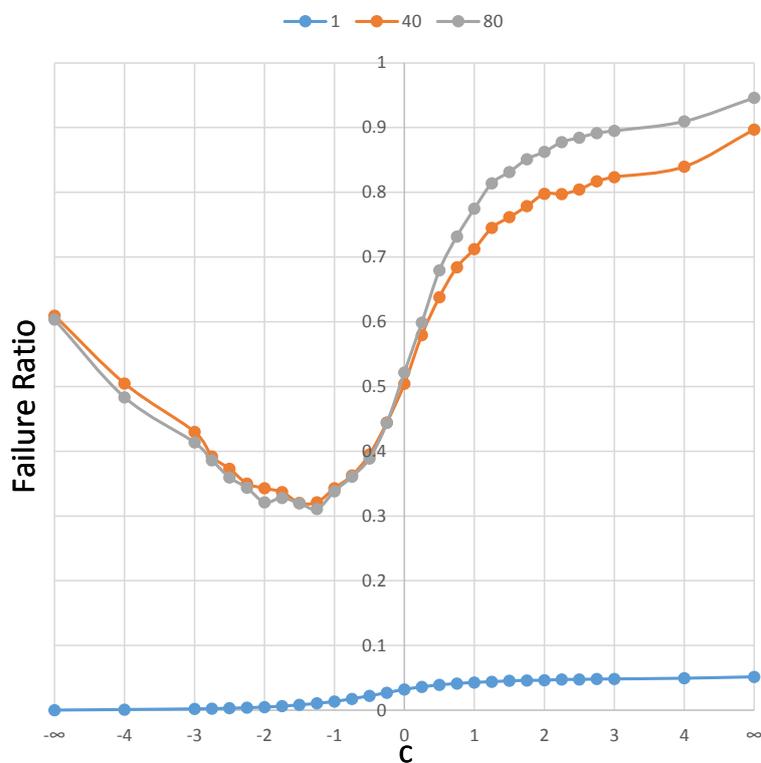


図 27. コンテンツ入手失敗率（べき乗則：クエリ記録（共有なし））

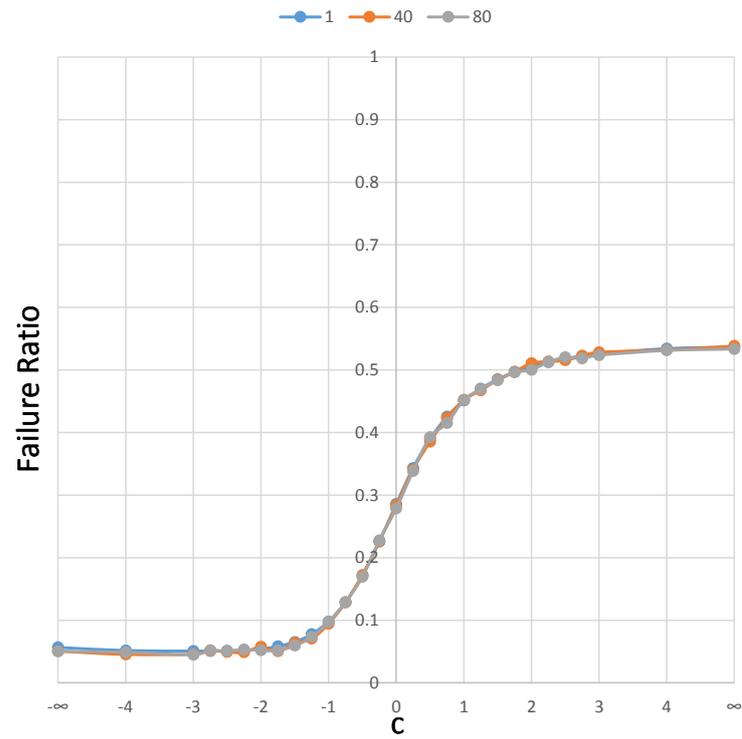


図 28. コンテンツ入手失敗率 (Gnutella : ノード記録)

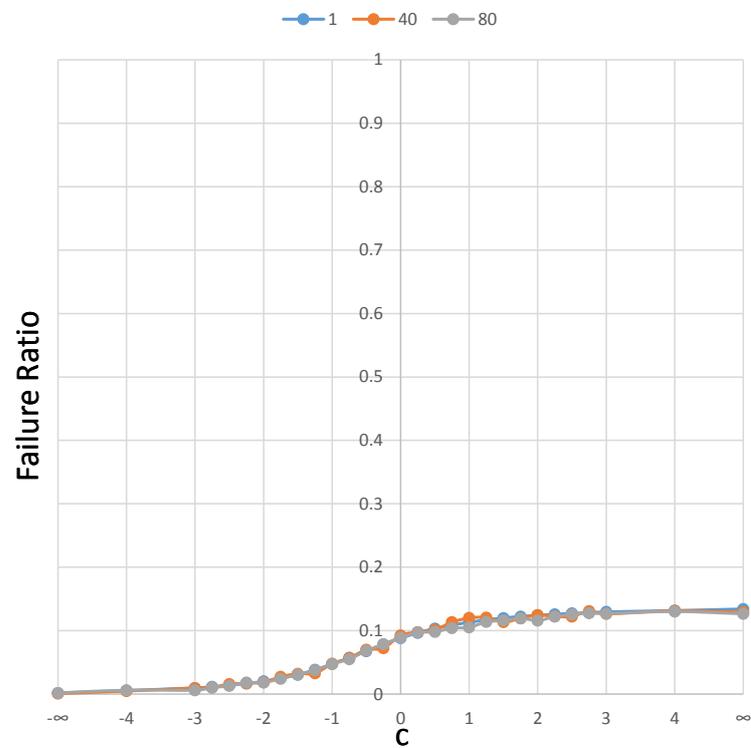


図 29. コンテンツ入手失敗率 (Gnutella : クエリ記録 (共有あり))

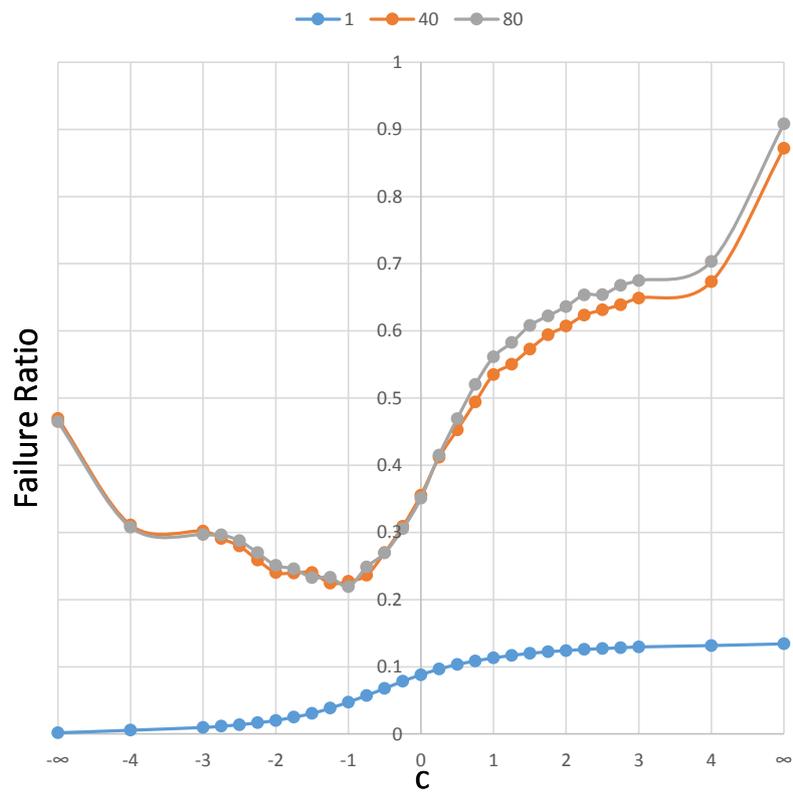


図 30. コンテンツ入手失敗率 (Gnutella : クエリ記録 (共有なし))

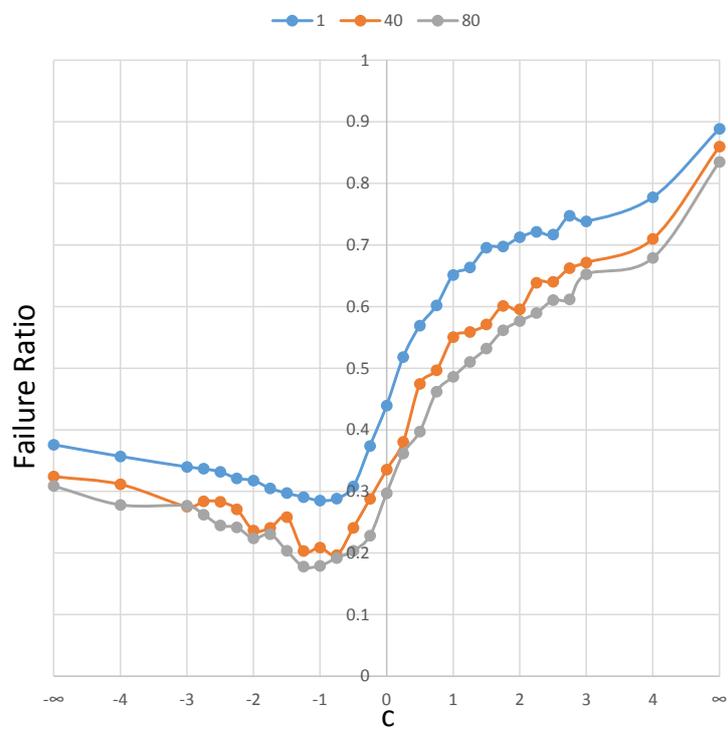


図 31. コンテンツ入手失敗率 (Facebook : ノード記録)

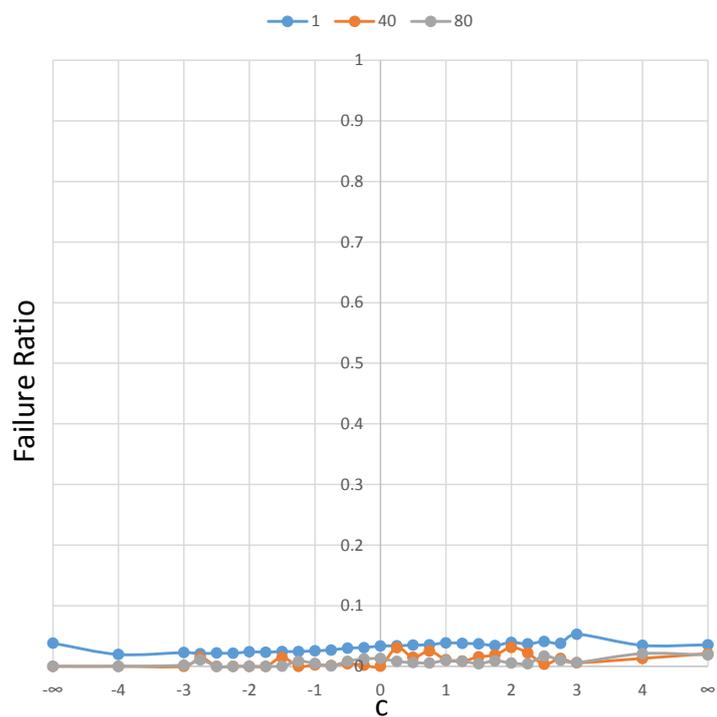


図 32. コンテンツ入手失敗率 (Facebook : クエリ記録 (共有あり))

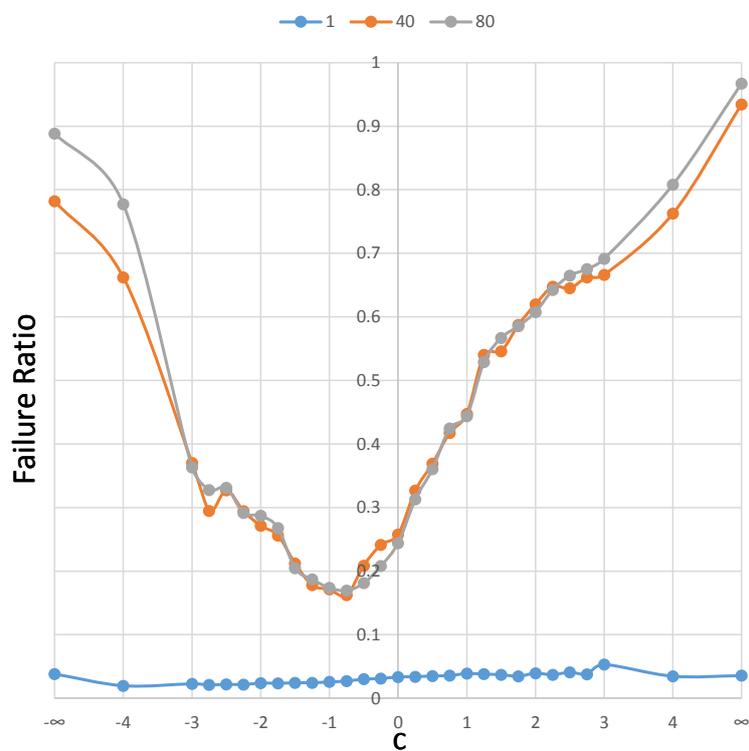


図 33. コンテンツ入手失敗率 (Facebook : クエリ (共有なし))

図 25-27 (べき乗則グラフ), 図 28-30 (Gnutella), 図 31-33 (Facebook) は, ノード記録, クエリ記録 (共有なし), クエリ記録 (共有あり) についてクエリ 1 個, 40 個, 80 個のときの入手失敗率を比較したものである.

べき乗則グラフ, Gnutella ネットワークでは, ノード記録, クエリ記録 (共有あり) の場合は, クエリ数によって検索失敗率に差がなかった. しかし, クエリ記録 (共有なし) の場合は, クエリ数を増やすと入手失敗率が大きく上昇した.

Facebook ネットワークでは, ノード記録, クエリ記録 (共有あり) の場合, クエリ数を増やすと検索失敗率が低下した. Facebook の平均次数の大きさにより, ノード記録では重複訪問する可能性が高い. そのため, より多くのクエリをネットワーク上にランダムに散布しランダムウォークさせることで, 重複訪問確率が緩和されると考えられる. クエリ記録では, べき乗則グラフ, Gnutella ネットワークと同様にクエリ数を増やすと入手失敗率が大きく上昇した.

クエリ記録 (共有なし) では, クエリ数を増やすと大きく入手失敗率が下がった. また, ノード記録, クエリ記録 (共有あり) では平均次数が大きいネットワークにおいて, クエリを増加させることが検索効率を高める効果が高まることがわかった.

第6章

まとめ

本論文では、有効なコンテンツ検索に関して理論的に考察し、ランダムウォーク検索において低次数ノードを優先することが検索効率を高めると予想し、シミュレーションにより評価した。

まず単一クエリで既訪問ノード回避アルゴリズムのノード記録型、クエリ記録型の入手失敗率をシミュレーションで評価した。ノード記録型では低次数優先型で入手失敗率が小さくなることがわかった。クエリ記録型では、ステップ数が増えると低次数優先型の入手失敗率が小さくなった。これにより、低次数優先型ではクエリ訪問確率の高次数ノードバイアスを緩和する効果が示された。

また、複数クエリの場合で、ノード記録型、クエリ記録型（共有あり）、クエリ記録型（共有なし）の入手失敗率をシミュレーションで評価した。クエリ記録（共有なし）の検索効率は、クエリ記録（共有あり）と比較して大きく劣化した。このことから、検索効率を担保するためには、既訪問ノード情報を共有する必要があることが示された。

またクエリの個数による比較を行った。クエリ記録（共有なし）では、クエリ数を増やすと大きく入手失敗率が下がった。また、ノード記録、クエリ記録（共有あり）では平均次数が大きいネットワークにおいて、クエリを増加により検索効率を高める効果が大きくなることがわかった。

本研究では、ネットワーク内の一つのノードがコンテンツを所持しているレアコンテンツを想定しているが、今後の課題として、コンテンツの存在確率を変えた場合の入手失敗率の評価が挙げられる。

参考文献

- [1] 佐藤一郎, “モバイルエージェント技術と動向,” NIIJournal, no.3, pp.53-66, 2001.
- [2] 森下美希, 木下和彦, 山井成良, 村上孝三, “異なる制限時間を持つエージェントのための公平なスケジューリング方式と その実装, ” 電子情報通信学会 ネットワークシステム研究会, NS2011-269, pp.511-516, 2012.
- [3] L.A. Adamic, R.M. Lukose, A.R. Puniyani, and B.A. Huberman, “Search in power-law networks,” Physical review E, vol.64, no.4, p.046135, 2001.
- [4] B. Yang and H. Garcia-Molina, “Improving search in peer-to-peer networks,” IEEE ICDCS, pp.5-14, 2002.
- [5] S. Jiang, L. Guo, and X. Zhang, “Lightflood: an efficient flooding scheme for file search in unstructured peer-to-peer systems,” Parallel Processing, 2003. Proceedings. 2003 International Conference on IEEE, pp.627-635 2003.
- [6] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, “Search and replication in unstructured peer-to-peer networks,” Proceedings of the 16th international conference on Supercomputing, pp.84-95, 2002.
- [7] S. Shioda, K. Ohtsuka, and T. Sato, “An efficient network-wide broadcasting based on hop-limited shortest-path trees,” Computer Networks, vol.52, no.17, pp.3284-3295, 2008.
- [8] P. Hieungmany and S. Shioda, “Characteristics of random walk search on embedded tree structure for unstructured p2ps,” IEEE ICPADS, pp.782-787, 2010.
- [9] S. Shioda and P. Hieungmany, “Random walk search on concatenated hop-limited trees embedded into unstructured p2ps,” International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), pp.43-50, 2011.
- [10] チュンマニプーヴィエン, 相馬知哉, 塩田茂雄, “非構造型 p2p システムにおける指向性ランダムウォーク検索, ” 電子情報通信学会論文誌 B, vol.98, no.2, pp.132-140, 2015.
- [11] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabasi, “Controllability of complex networks,” Nature, vol.473, no.12, pp.167-173, 2011.
- [12] H. Toyozumi, S. Tani, N. Miyoshi, and Y. Okamoto, “Reverse preferential spread in complex networks,” Physical Review E, vol.86, no.2, p.021103, 2012.
- [13] 中村和弘, 原田淳子, 塩田茂雄, “Havel-hakimi の定理を用いた 次数列を再現するネットワークの構成法とその特徴, ” 電子情報 通信学会論文誌 A, vol.J92-A, no.9, pp.604-612, 2009.
- [14] “Stanford large network dataset collection,” <http://snap.stanford.edu/data/>.

謝辞

今回、本研究を進めていくにあたって、丁寧にご指導いただいた塩田茂雄教授に深く感謝いたします。また、いつも暖かく接して下さった塩田研究室の方々にも深く感謝いたします。ありがとうございました。