Low-degree-preferential Random Walk for Information Search

Shigeo Shioda*,
*Graduate School of Engineering, Chiba University
1-33 Yayoi, Inage, Chiba 263-8522, Japan
E-mail: shioda@faculty.chiba-u.jp

***Abstract*—A random-walk-based (RW-based) information search makes a single query or multiple queries walk around a network to search for the target information. Local navigation rules, based on which a query chooses an adjacent node to move from the current location, are important for increasing the search speed of the RW-based information search. In the present paper, we focus on degree-dependent navigation rules that use information on the degrees of nodes adjacent to the query's location. We find that a *low-degree-preferential* RW, where the query preferentially moves a step to lower-degree nodes, exhibits better search performance than a high-degree-preferential or an unbiased RW. In the present paper, we present several theoretical and numerical results in order to support this rather surprising finding.**

I. Introduction

The Internet is a huge database storing large amounts of various types of information. Due to the unstructured nature of the Internet, however, sometimes it is not easy to retrieve desired information. We often conduct information searches on the Internet by manually using search engines, such as Google or Yahoo, which rely on a centralized and structured database. Developing and maintaining such databases is extremely expensive because the Internet is still growing and information changes very frequently. In this sense, exploring other possible techniques for information search remains an important issue.

In the present paper, we are concerned with information searches on the Internet using *mobile agents* [1]. Mobile agents are software programs that can migrate from host to host in a network to perform a given job at times and places of their own choosing. Information search and retrieval using mobile agents have been investigated over the past decade and have been used in file-sharing systems based on unstructured P2Ps. For example, Gnutella uses a kind of mobile agent, called a query, for file searches. A Gnutella client (node) initiates file searches by broadcasting queries to its adjacent nodes. A query searches for the target file when it arrives at a node and, if it fails to find the target file there, it makes copies of itself and broadcasts them to adjacent nodes of its current location. This procedure is referred to as *flooding*. The flooding-based search is very powerful but requires a large amount of bandwidth and congests the network with query traffic. In order to avoid congestion, several extensions of the flooding-based search have been proposed and used in unstructured P2Ps [2], [3].

Instead of a flooding-based search, in the present paper, we explore the possibility of a random-walk-based (RW-based) search, where a single query or multiple queries walk around the network to search for information satisfying given conditions (e.g., including a given list of keywords) [4], [5]. Starting from a user node, a query visits adjacent nodes and searches for the target information there. If the query fails to find the target information, it moves a step to one of the nodes adjacent to its current location. The query continues this procedure until the target information is found or the number of visits exceeds a given limit. The load of the RW-based search is less than that of the flooding-based search but is likely to require a longer time to retrieve the target information than the flooding-based search.

In order to increase the search speed of the RW-based information search, in the present paper, we investigate local navigation rules, according to which a query chooses an adjacent node to move from the current location. In particular, we focus on degree-dependent navigation rules that use information on the degrees of adjacent nodes. Adamic *et al.* [4] found that intentionally moving a step to high-degree nodes increases the search speed when a query knows all pieces of information stored at its first and second neighbor locations. In actuality, however, it seems difficult for a query to know the information stored at neighbor nodes.

In the present paper, we examine the local navigation rule in the simplest and mostly common situation, in which a query does not know anything about information stored at nodes neighboring its location. In this situation, we are surprisingly driven to a totally opposite conclusion to that of Adamic *et al.*, namely, the *low-degree-preferential* RW, where the query preferentially moves a step to lower-degree nodes, exhibits better search performance than the high-degree-preferential (or unbiased) RW. In the present paper, we present some theoretical results as well as numerical evidence indicating the suitability of the low-degree-preferential RW for use in information search.

The remainder of the present paper is organized as follows. In Section II, we summarize research related to the search algorithm proposed for unstructured P2Ps. In Section III, we explain the concept of the low-degree-preferential RW and present theoretical results showing why it is promising for use in information searches. In Section IV, we present several simulation results to demonstrate the effectiveness of the low-degree-preferential RW for information search. Finally, we conclude the paper in Section V.

II. Related Research

The initial version of Gnutella broadcasted query messages over the network by flooding. The current version of Gnutella

978-1-5090-1525-2/17/$31.00 ©2017 IEEE

uses iterative deepening [3], whereby flooding is performed while increasing the time-to-live (TTL) constraint until the search is successful [6]. Jiang *et al.* [2] proposed *LightFlood*, which broadcasts over a preconstructed tree, and we previously proposed broadcasting over the concatenation of the hop-limited shortest-path trees [7]. These tree-based broadcasting methods are effective for preventing multiple copies of a message from arriving at any one node.

Some research has been conducted on searching with a random walk, in which an unduplicated query wanders around the network in order to reduce the bandwidth used in flooding [5], [4]. Lv *et al.* [5] proposed *k-random walk*, a search based on a random walk, where k random walkers are used to search for the requested file. Adamic *et al.* [4] proposed a random walk in which walkers are forwarded to nodes that are proportional to their degrees. They also proposed a self-avoiding random walk, in which walkers cannot return to nodes they have already visited. Flooding wastes resources, while random-walk searches or depth-first searches significantly increase the latency of the search.

Gkantsidis *et al.* [8] proposed a hybrid search scheme that combines flooding and random walks. In their scheme, a node initiating a search determines in advance the number of queries replicated in the search, referred to as the *budget*. When forwarding a query with budget k to d neighbor nodes, the node distributes the budget by selecting d integers $k_1, \ldots, k_d$ with $k_1 + \cdots + k_d = k$, and forwards the query to node i with a budget equal to k_i. Each neighbor reduces the budget by 1 and repeats the procedure. When $d = 1$, this scheme is equivalent to a random-walk search with hop limit k. When d is equal to the node degree, it is equivalent to flooding with a constraint on the number of queries replicated. Chen *et al.* [9] also proposed a hybrid scheme similar to that proposed by Gkantsidis *et al.* [8].

Only a few studies focused on the importance of low-degree nodes. Among these, Liu [10] showed that the fraction of *driver nodes*, which can offer full control over the network, is significantly higher among low-degree nodes than among the hub nodes. Toyoizumi *et al.* [11] proposed an information-spreading mechanism that distributes information with a probability that is inversely proportional to the degree of the receiving end node.

III. Proposal

A. Low-degree-preferential RW

Consider a random walk in which a single walker wanders on a nondirectional and connected graph (network). In the present paper, the random walk, in which the walker randomly selects an adjacent node to move to with equal probability is referred to as the *basic random walk (RW)*, and the information search based on the basic RW is referred to as the *basic-RW-based* search. The walker under the basic RW is known to have a preference for high-degree nodes. More precisely, the probability that a basic-RW-based walker stays at a given node is proportional to its degree in a stationary state. Because of this, the basic-RW-based search may require a long time to retrieve the desired information if it is stored in a low-degree node.

If the amount of information stored at a given node does not depend on its degree, the walker should equally visit all nodes regardless of their degrees. In order to mitigate the preference for high-degree nodes of the basic RW, we propose the *low-degree-preferential* RW, where the walker is likely to move a step to an adjacent node, the degree of which is lower than those of other adjacent nodes.

In order to explain in detail the algorithm of the low-degree-preferential RW, let $\mathscr{N}(i)$ be a set of adjacent nodes of node i, and let d_j be the degree of node $j \in \mathscr{N}(i)$. The walker locating node i moves to node $j \in \mathscr{N}(i)$ with probability $p_{i\to j}$, which is given below.

$$p_{i\to j} = \frac{d_j^c}{\sum_{k\in\mathscr{N}(i)} d_k^c}. \tag{1}$$

where c is referred to herein as the *preference parameter*. When $c = -1$, the walker moves to an adjacent node with a probability that is inversely proportional to the degree. In the limit of $c \to -\infty$, transition probability (1) yields the lowest-degree-preferential RW, where the walker moves to the lowest-degree node among adjacent nodes of its current location. The random walk governed by transition probability (1) is referred to herein as the *low-degree-preferential* RW if the preference parameter is less than 0.

B. Theoretical Considerations

1) Mathematical Model: Let us consider a mathematical model in which the walker can move from any node to any other node directly, regardless of the existence of the link between the two nodes. Although this model does not precisely represent actual random walks, it concisely reveals the reason why the low-degree-preferential RW is preferable for use in information search. Let N be the number of nodes. We assume that the transition probability from node i to node j depends only on j, denoted by π_j, and does not depend on i. Under this assumption, the stationary occupation probability of the walker is equal to $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$, i.e., the walker stays at node i with probability π_i in stationary states. Define

$$I_i(n) \overset{\text{def}}{=} \mathbf{1}(\text{the walker has visited node } i \text{ until } T_n).$$

where T_n is the time at which the walker makes the nth step, and $\mathbf{1}(A)$ is the indicator function that is equal to 1 (0) when A is true (false). The number of distinct nodes that the walker has visited until T_n, denoted by S_n, is given as

$$S_n = \sum_{i=1}^{N} I_i(n). \tag{2}$$

In order to obtain the expectation of S_n, note that

$$I_i(n) = \sum_{k=1}^{n} \mathbf{1}(\text{the walker first visits node } i \text{ at } T_k),$$

from which

$$\begin{aligned} E[I_i(n)] &= \sum_{k=1}^{n} P(\text{the walker first visits node } i \text{ at } T_k) \\ &= \sum_{k=1}^{n} \pi_i (1-\pi_i)^{k-1} \\ &= 1 - (1-\pi_i)^n. \end{aligned}$$

Substituting the above result into (2) yields

$$E[S_n] = \sum_{i=1}^{N} E[I_i(n)] = N - \sum_{i=1}^{N} (1-\pi_i)^n.$$

This model is closely related to the classical coupon-collection problem [12], [13], where coupons are collected one-by-one by purchasing coupon packages, each of which contains only one coupon. The type i coupon is contained in a purchased package with probability π_i, and the types of coupons contained in different packages are statistically independent. It is easy to see that S_n corresponds to the number of purchases necessary for acquiring n distinct types of coupons in the classical coupon-collection problem. In order to remind us of the dependence of S_n on $\boldsymbol{\pi}$, in the following we denote S_n by $S_n(\boldsymbol{\pi})$.

For later use, we introduce the definition of the usual stochastic order (Definition 1) as well as two lemmas (Lemmas 1 and 2), which shows the usual-stochastic-order relationship in the classical coupon collection problem.

Definition 1 (Shaked and Shanthikumar [14]). *Let X_1 and X_2 be random variables on $\mathbb{R}$. Then, X_1 is called smaller than X_2 in terms of the usual stochastic order, denoted as $X_1 \leq_{\mathrm{st}} X_2$, if*

$$P(X_1 > x) \leq P(X_2 > x) \quad \text{for all } x.$$

Lemma 1 (Shaked and Shanthikumar [14]). *If $X_1 \leq_{\mathrm{st}} X_2$, then for all increasing function f*

$$E[f(X_1)] \leq E[f(X_2)],$$

provided the expectations exist.

Lemma 2 (Shioda [13]). *If $S_n(\boldsymbol{\pi}) \leq_{\mathrm{st}} S_n(\boldsymbol{\pi}_{ave})$, where*

$$\boldsymbol{\pi}_{ave} \stackrel{\text{def}}{=} (\bar{p}, \dots, \bar{p}), \quad \bar{p} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{i=1}^{N} \pi_i = \frac{1}{N}.$$

Remark 1. It follows from Lemma 1 and Lemma 2 that

$$E[S_n(\boldsymbol{\pi})] \leq E[S_n(\boldsymbol{\pi}_{ave})].$$

This can also be confirmed by the following direct calculation:

$$\begin{aligned} E[S_n(\boldsymbol{\pi})] &= N - \sum_{i=1}^{N} (1-\pi_i)^n \\ &= N - N \sum_{i=1}^{N} \frac{1}{N}(1-\pi_i)^n \\ &\leq N - N\left(1 - \frac{1}{N}\sum_{i=1}^{N} \pi_i\right)^n \\ &= N - N\left(1-\frac{1}{N}\right)^n = E[S_n(\boldsymbol{\pi}_{ave})] \end{aligned}$$

where their line follows from the fact that $f(x) = (1-x)^n$ is convex.

2) Information Placement with Constant Probability: Now suppose that the target information is stored at an arbitrary node with constant probability r. Let $P_f(n;\boldsymbol{\pi})$ be the probability that the walker has not found the target information until its nth visit when the stationary occupation probability of the walker is equal to $\boldsymbol{\pi}$. The following theorem shows that equally visiting all nodes is the best strategy for obtaining the target information within a given time limit.

Theorem 1. *$P_f(n;\boldsymbol{\pi})$ is minimized at $\boldsymbol{\pi} = \boldsymbol{\pi}_{ave}$. In other word, for all $\boldsymbol{\pi}$ and for all $n = 1, 2, \dots,$*

$$P_f(n;\boldsymbol{\pi}_{ave}) \leq P_f(n;\boldsymbol{\pi}).$$

Proof: Let A_n be the event whereby the walker has visited one of nodes storing the target information until T_n. Note that

$$E[\mathbf{1}(A_n)|S_n = S] = 1 - (1-r)^S.$$

Thus,

$$\begin{aligned} P_f(n;\boldsymbol{\pi}) &= 1 - E[\mathbf{1}(A_n)] \\ &= 1 - E[E[\mathbf{1}(A_n)|S_n(\boldsymbol{\pi}) = S]] \\ &= E[(1-r)^{S_n(\boldsymbol{\pi})}]. \end{aligned}$$

Since $f(x) = (1-r)^x$ is a decreasing function, it follows from Lemma 1 and Lemma 2 that

$$\begin{aligned} P_f(n;\boldsymbol{\pi}_{ave}) &= E[(1-r)^{S_n(\boldsymbol{\pi})}] \\ &\leq E[(1-r)^{S_n(\boldsymbol{\pi}_{ave})}] = P_f(n;\boldsymbol{\pi}), \end{aligned}$$

which completes the proof. ■

Let $T_s(\boldsymbol{\pi})$ be the number of steps that the walker takes until it finds the target information when the stationary occupation probability is equal to $\boldsymbol{\pi}$. Using $T_s(\boldsymbol{\pi})$, we define

$$R(n;\boldsymbol{\pi}) \stackrel{\text{def}}{=} \min\{T_s(\boldsymbol{\pi}), n\}. \tag{3}$$

Here, $R(n;\boldsymbol{\pi})$ corresponds to the response time of the information search when the walker needs to return the search result by the nth step. In other words, when the number of steps of the walker reaches n, the query needs to send the search result regardless of whether the search has succeeded or failed.

Theorem 2. *For all $\boldsymbol{\pi}$ and $n = 1, 2, \dots,$*

$$E[R(n;\boldsymbol{\pi}_{ave})] \leq E[R(n;\boldsymbol{\pi})].$$

Proof: Let A_n^c be the event whereby the walker has not visited any node storing the target information by T_n. It follows that

$$\begin{aligned} E[R(n;\boldsymbol{\pi})] &= \sum_{k=0}^{n-1} k(\mathbf{1}(A_k^c) - \mathbf{1}(A_{k+1}^c)) + n\mathbf{1}(A_n^c) \\ &= \sum_{k=0}^{n} \mathbf{1}(A_k^c) \\ &= \sum_{k=0}^{n} P_f(k;\boldsymbol{\pi}). \end{aligned} \tag{4}$$

Thus, the desired conclusion readily follows from Theorem 1. ■

3) More General Placement of Information: Next, consider situations in which the target information is more generally placed in a network. Suppose that the number of nodes storing the target information follows a given probability distribution $\{p_s(k)\}_{k=0}^{N}$. The target information is stored at k nodes with probability $p_s(k)$. The nodes storing the target information are randomly located in the network when the number of nodes storing the target information is given. Note that Section III-B2

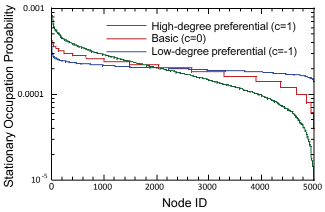

Fig. 1. Occupation Probability (Random Graph)

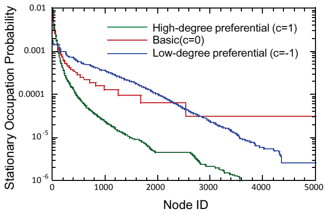

Fig. 2. Occupation Probability (Power-law Graph)

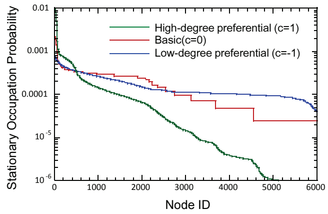

Fig. 3. Occupation Probability (Gnutella)

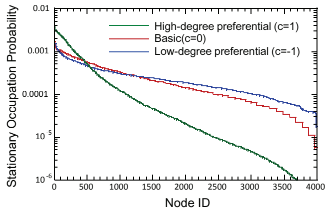

Fig. 4. Occupation Probability (Facebook)

is a special case of the above-mentioned situation, in which $p_s(k) = \binom{N}{k} r^k (1-r)^k$. The following theorem indicates that the same conclusion for Theorem 1 in Section III-B2 also holds in the abovementioned situations.

Theorem 3. *$P_f(n; \boldsymbol{\pi})$ is minimized at $\boldsymbol{\pi} = \boldsymbol{\pi}_{ave}$. In other words, for all $\boldsymbol{\pi}$ and $n = 1, 2, \ldots,$*

$$P_f(n; \boldsymbol{\pi}_{ave}) \le P_f(n; \boldsymbol{\pi}).$$

Proof: Let A_n be the event whereby the walker has visited one of the nodes storing the target information until T_n. Note that

$$E[\mathbf{1}(A_n)|S_n = S] = \sum_{k=1}^{N} p_s(k)\left(1 - \left(\frac{N-S}{N}\right)^k\right).$$

Thus,

$$\begin{aligned} P_f(n; \boldsymbol{\pi}) &= 1 - E[\mathbf{1}(A_n)] \\ &= 1 - E[E[\mathbf{1}(A_n)|S_n(\boldsymbol{\pi}) = S]] \\ &= p_s(0) + \sum_{k=1}^{N} p_s(k) E\left[\left(1 - \frac{S_n(\boldsymbol{\pi})}{N}\right)^k\right]. \end{aligned}$$

Since $f(x) = (1 - x/N)^k$ is a decreasing function, it follows from Lemma 1 and Lemma 2 that

$$\begin{aligned} P_f(n; \boldsymbol{\pi}_{ave}) &= p_s(0) + \sum_{k=1}^{N} p_s(k) E\left[\left(1 - \frac{S_n(\boldsymbol{\pi}_{ave})}{N}\right)^k\right] \\ &\le p_s(0) + \sum_{k=1}^{N} p_s(k) E\left[\left(1 - \frac{S_n(\boldsymbol{\pi})}{N}\right)^k\right] \\ &= P_f(n; \boldsymbol{\pi}), \end{aligned}$$

which completes the proof. ■

We also see that $E[R(n; \boldsymbol{\pi})]$ is minimized at $\boldsymbol{\pi} = \boldsymbol{\pi}_{ave}$ for the cases considered in this subsection.

C. Stationary Occupation Probability under Low-degree-preferential RW

The results in Section III-B suggest that the RW-based search would perform the best in terms of minimizing the search-failure ratio or minimizing the response time when the stationary occupation probability $\boldsymbol{\pi}$ is uniform. The walker visits all nodes in the network with equal probability. Although the stationary occupation probability is not uniform under the low-degree-preferential RW, we could expect that the low-degree-preferential RW makes the occupation probability more uniform than the basic or high-degree-preferential RW. Table I compares the difference between the largest and smallest stationary occupation probabilities of the walker, i.e.,

TABLE I. Difference between the largest and smallest stationary occupation probabilities

Graph	$c = 0$	$c = 1$	$c = -1$
Random	4.82×10^{-4}	1.10×10^{-3}	2.66×10^{-4}
Power law	3.55×10^{-2}	9.58×10^{-2}	2.16×10^{-3}
Gnutella	2.31×10^{-3}	1.62×10^{-2}	1.15×10^{-3}
Facebook	5.92×10^{-3}	2.78×10^{-2}	2.04×10^{-3}

$$\pi_{def} \stackrel{\text{def}}{=} \max_i \pi_i - \min_i \pi_i,$$

for three types of random walks (RWs): basic RW, high-degree-preferential RW ($c = 1$ in (1)), and low-degree-preferential RW ($c = -1$ in (1)). The stationary occupation probability is numerically obtained by solving the Markov chain describing the transition of the walker. We conduct the numerical experiments in four different networks: a random graph, a power-law graph, a Gnutella network, and a Facebook network. These networks will be described in detail in Section IV-A. Table I verifies our expectation. Namely, the low-degree-preferential RW exhibits the smallest π_{def} among the three RWs for all four of the networks.

Figure 1 compares the stationary occupation probabilities for the three RWs on the random graph. The vertical axis shows the occupation probability, and the horizontal axis shows the node number, which is assigned in descending order of occupation probability ($\pi_i \ge \pi_{i+1}$ for all $i = 1, 2, \ldots$). The figure reveals that the occupation probability is more uniform under the low-degree-preferential RW than under the basic or high-degree-preferential RW. The same comparison is made in Figs. 2 (power-law graph), 3 (Gnutella), and 4 (Facebook), which also provide numerical support for our expectation.

D. Self-avoidance

If the walker moves a step without using the information on the past trajectory, the walker may wastefully visit the same node multiple times, which degrades the efficiency of the information search. In order to avoid multiple redundant visits to the same node, we herein consider a function called *self-avoidance*, which allows nodes to keep the information on the trajectory of the walker. In order to explain the algorithm of the self-avoidance function, consider the case in which the walker currently stays at node i. When the self-avoidance function is used, node i knows a set of adjacent nodes, denoted by $\mathcal{N}_m(i)$, to which the walker has already moved from node i. Using the knowledge on $\mathcal{N}_m(i)$, the walker selects node $j \in \mathcal{N}(i) \setminus \mathcal{N}_m(i)$

TABLE II. Networks Used in the Simulation

Types of graph	# of nodes	# of links
Random graph	5000	24894
Power-law graph	5000	15915
Gnutella	6299	20776
Facebook	4039	88234

to move with probability $p_{i \to j}$, which is given as follows:

$$p_{i \to j} = \frac{d_j^c}{\sum_{k \in \mathcal{N}(i) \setminus \mathcal{N}_m(i)} d_k^c}. \tag{5}$$

If $\mathcal{N}(i) = \mathcal{N}_m(i)$ ($\mathcal{N}(i) \setminus \mathcal{N}_m(i) = \emptyset$), the walker selects an adjacent node to move to with equal probability. We have numerically confirmed that the self-avoidance function significantly improves the search performance of the low-degree-preferential RW.

IV. Simulation Experiments

A. Simulation Conditions

We evaluated the search performance of the low-degree-preferential RW through simulation experiments. We used four different model networks listed in Table II. The random graph has 5,000 nodes and 24,894 links. Each pair of nodes is connected by a single link with probability 0.002. The power-law graph was obtained as follows. First, a degree sequence was produced from the power-law degree distribution with exponent 2.0, and a graph having the produced degree sequence was generated by the Havel-Hakimi algorithm [15]. Then, rewiring of the generated graph while maintaining its degree sequence was performed numerous times in order to remove the degree correlation, and the resultant power-law graph was used in the simulation. Gnutella and Facebook networks were generated based on datasets available at the site of Stanford University [16]. Note that Gnutella forms a directional network, and thus we generate a nondirectional network from the dataset of Gnutella by making links nondirectional.

B. Results 1: Rare Information Retrieval

First, we investigated the search performance of the low-degree-preferential RW when the target was rare information, which is stored at only a single node. This situation corresponds to the case of Section III-B3, where $p_s(1) = 1$ and $p_s(k) = 0$ for $k = 2, 3, \ldots, N$. We evaluated the search failure ratio $P_f(n)$, i.e., the probability that the target information has not been found before the nth step of the walker, by setting n to four different values ($N/2, N, 3N/2, 2N$), where N is the number of nodes. Note that parameter n corresponds to the response time limit. The results are shown in Figs. 5 (random graph), 6 (power-law graph), 7 (Gnutella), and 8 (Facebook). The horizontal axis of these figures shows the preference parameter c in (1). In the figures, $c = -\infty$ indicates the lowest-degree-preferential RW, in which the walker moves a step to the lowest degree node among adjacent nodes of the current location, and $c = \infty$ indicates the highest-degree-preferential RW, in which the walker moves a step to the highest degree node among adjacent nodes of the current location. The self-avoidance function was used in the simulation. Each figure shows an average search failure ratio, which was taken for all possible pairs of the target-storing node and the search initiation node.

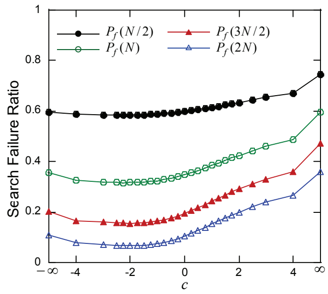

Fig. 5. Search Failure Ratio (Random Graph: rare information)

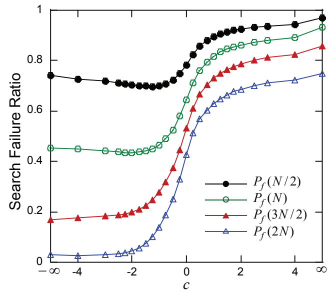

Fig. 6. Search Failure Ratio (Power-law Graph: rare information)

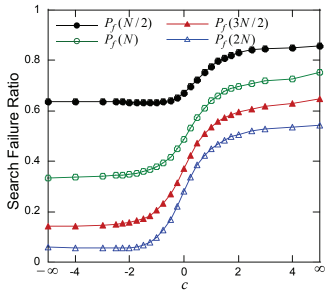

Fig. 7. Search Failure Ratio (Gnutella: rare information)

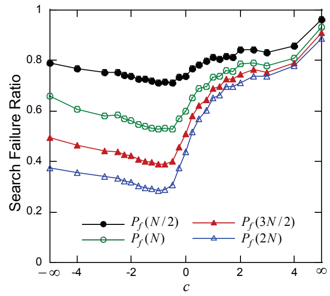

Fig. 8. Search Failure Ratio (Facebook: rare information)

All of the figures confirm that the search failure ratio decreases as preference parameter c decreases from ∞. In the random graph (Fig. 5), $P_f(2N)$ is minimized around $c = -2$, and, in the power-law graph (Fig. 6), $P_f(2N)$ is minimized around $c = -3$. For Gnutella (Fig. 7), $P_f(2N)$ is minimized around $c = -4$, and, for Facebook (Fig. 8), $P_f(2N)$ is minimized around $c = -1$. Based on these results, we see that the preference for low-degree nodes decreases the search failure ratio, but the excessive preference may degrade the search performance. The best value of the preference parameter depends on the structure of the graph. We also see that the gain obtained by the preference for low-degree nodes becomes large as n (response time limit) increases.

We also evaluated the average response time $E[R(n)]$, defined in (3), through simulation experiments by setting the response time limit n to four different values ($N/2, N, 3N/2, and 2N$). The results are shown in Figs. 9 (random graph), 10 (power-law graph), 11 (Gnutella), and 12 (Facebook). These figures also confirm that the low-degree-preferential RW outperforms the basic RW or the high-degree-preferential RW in terms of faster retrieval of the target information.

C. Results 2: Common Information Retrieval

Next, we investigated the search performance of the low-degree-preferential RW when the target information was more common and was stored at an arbitrary node with a probability of 0.002. For example, the target information was stored at 10 nodes on average in the random graph used in the simulation because it had 5,000 nodes.

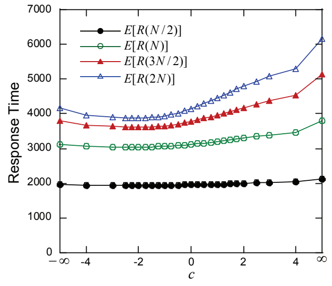

Fig. 9. Response Time (Random Graph: rare information)

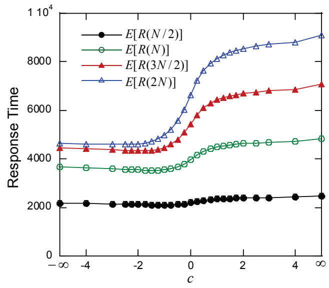

Fig. 10. Response Time (Power-law Graph: rare information)

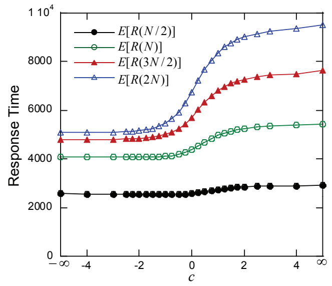

Fig. 11. Response Time (Gnutella: rare information)

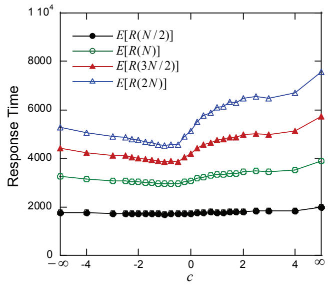

Fig. 12. Response Time (Facebook: rare information)

The search failure ratios $P_f(n)$ evaluated by setting n at four different values ($N/8$, $N/4$, $3N/8$, $N/2$) are summarized in Figs. 13 (random graph), 14 (power-law graph),15 (Gnutella), and 16 (Facebook). Since the target file is stored at multiple nodes, the search failure ratio is much smaller than in the rare-information retrieval case in Section IV-B. By focusing on $P_f(N/2)$, we find that the preference toward low-degree nodes is also effective for decreasing the search failure ratio. For example, in Fig. 13 (Random Graph), $P_f(N/2)$ is minimized around $c = -2.25$. In the power-law graph (Fig. 14) and for Gnutella (Fig. 15), $P_f(N/2)$ is minimized around $c = -1.25$, and, for Facebook (Fig. 16), $P_f(N/2)$ is minimized around $c = -1$.

Table III compares the search failure ratio $P_f(2N)$ at $c = 0$ with the minimum value of $P_f(2N)$ when the target is rare information. The minimum value of $P_f(2N)$ is obtained by adjusting the value of preference parameter c. Table IV also compares $P_f(2N)$ for $c = 0$ with the minimum value of $P_f(2N)$ when the target is common information. Based on these tables, the gain obtained by the low-degree preference is slightly smaller for common information retrieval than for rare information retrieval. When the target information is common, the query could find the target information with a small number of steps. This is the reason why the low-degree-preferential RW is less effective for common-information retrieval than for rare-information retrieval.

The average response time $E[R(n)]$ evaluated for four different values ($N/8$, $N/4$, $3N/8$, and $N/2$) of n are also summarized in Figs. 17 (random graph), 18 (power-law graph), 23 (Gnutella), and 24 (Facebook). As in the case of rare information retrieval, the low-degree-preferential RW outperforms the basic RW and the high-degree-preferential RW in terms of faster retrieval of the target information. When the response time limit n is small (e.g., $n = N/8$), however, the gain obtained

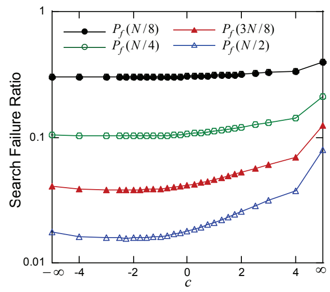

Fig. 13. Search Failure Ratio (Random Graph: common information)

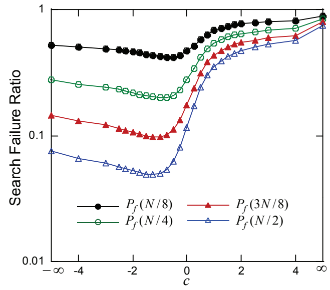

Fig. 14. Search Failure Ratio (Power-law Graph: common information)

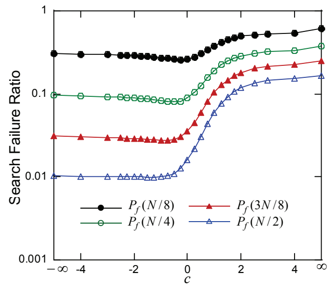

Fig. 15. Search Failure Ratio (Gnutella: common information)

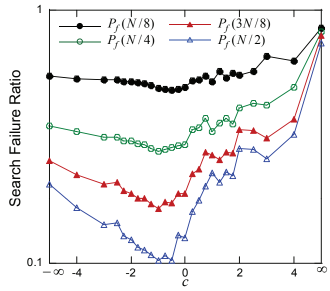

Fig. 16. Search Failure Ratio (Facebook: common information)

by the preference for low-degree nodes is not so significant.

TABLE III. GAIN OBTAINED BY THE LOW-DEGREE PREFERENCE (RARE INFORMATION).

Graph	$P_f(2N)$ at $c = 0$	$\min\{P_f(2N)\}$	ratio
Random	0.104821	0.067345 (c=-2)	0.642476
Power law	0.428333	0.025544 (c=-4)	0.059636
Gnutella	0.280931	0.054715 (c=-3)	0.194763
Facebook	0.439501	0.285184 (c=-1)	0.648881

TABLE IV. GAIN OBTAINED BY THE LOW-DEGREE PREFERENCE (COMMON INFORMATION).

Graph	$P_f(N/2)$ at $c = 0$	$\min\{P_f(N/2)\}$	ratio
Random	0.017938	0.015752 (c=-2.25)	0.878136
Power law	0.115476	0.048901 (c=-1.25)	0.423473
Gnutella	0.015932	0.009953 (c=-1.25)	0.624718
Facebook	0.126569	0.102893 (c=-1)	0.81294

D. Effect of Self-avoidance

Finally, in Figs. 21 (power-law graph) and 22 (Gnutella), we show the search failure ratio when the self-avoidance function is disabled. Comparison of these figures with Figs. 6 and 7 reveals that the search performance without using the self-avoidance function is much worse than the performance with the self-avoidance function. The improvement by the self-avoidance function is large especially when the preference parameter c takes a large negative value. Figures 21 (power-law graph) and 22 (Gnutella), which show the response time when the self-avoidance function is disabled, also support the same conclusion. When the preference parameter is set to a large negative value, the walker is likely to stay at the lowest-degree nodes, and thus the stationary occupation probabilities at these

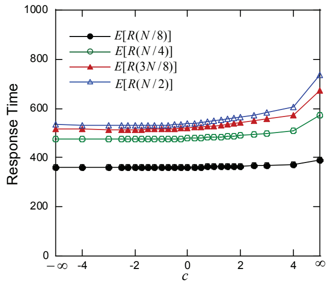

Fig. 17. Response Time (Random Graph: common information)

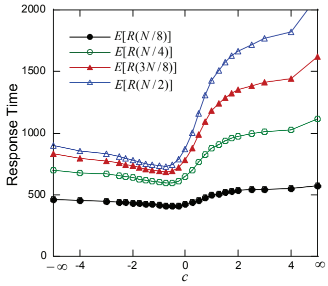

Fig. 18. Response Time (Power-law Graph: common information)

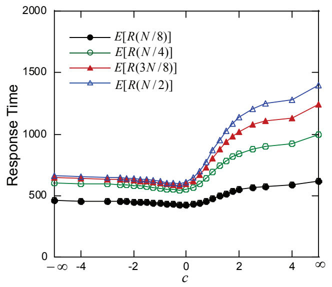

Fig. 19. Response Time (Gnutella: common information)

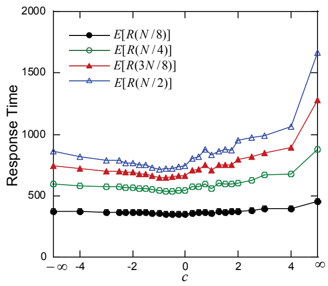

Fig. 20. Response Time (Facebook: common information)

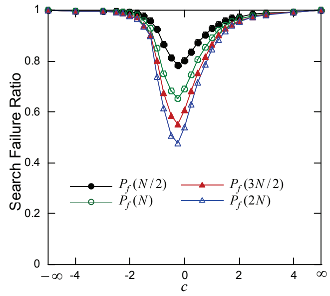

Fig. 21. Search Failure Ratio (Power-law Graph: self-avoidance disabled)

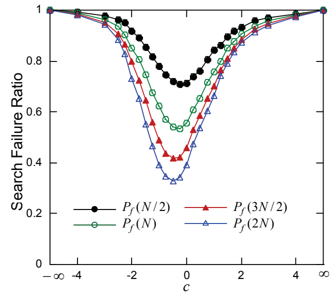

Fig. 22. Search Failure Ratio (Gnutella: self-avoidance disabled)

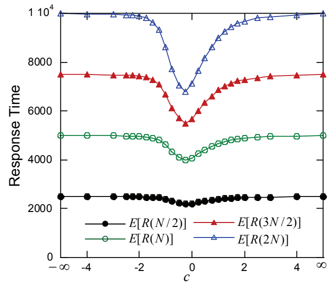

Fig. 23. Response Time (Power-law Graph: self-avoidance disabled)

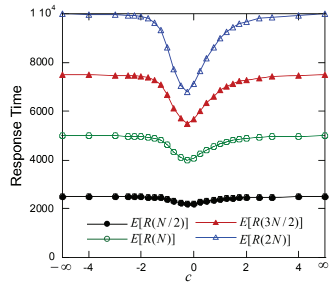

Fig. 24. Response Time (Gnutella: self-avoidance disabled)

nodes would be very large. The self-avoidance function is very effective in mitigating the bias of the occupation probability on the lowest-degree nodes.

V. Conclusion

In the present paper, we propose a low-degree-preferential RW in order to realize efficient information searches. The low-degree-preferential RW visits all nodes more equally than the basic RW and the high-degree-preferential RW. Because of this, faster information retrieval and a smaller search failure ratio are attained by the low-degree-preferential RW, especially in the case of rare-information retrieval. In the present paper, we focus only on the single-walker case, but in actual situations multiple walkers would be used. In multiple-walker cases, we could use a combination of both low-degree- and high-degree-preferential RWs. The local navigation rule for multiple walkers remains a subject for future research.

References

[1] R. Glitho, E. Olougouna, and S. Pierre, "Mobile agents and their use for information retrieval: a brief overview and an elaborate case study," *IEEE Network*, vol. 16, no. 1, pp. 34 – 41, 2002.

[2] S. Jiang, L. Guo, X. Zhang, and H. Wang, "LightFlood: minimizing redundant messages and maximizing the scope of peer-to-peer search," *IEEE Trans. Prallel and Distributed Systems*, vol. 19, no. 5, pp. 601–614, 2008.

[3] B. Yang and H. Garcia-Molina, "Improving search in peer-to-peer networks," in *IEEE ICDCS*, 2002, pp. 5–14.

[4] L. Adamic, R. Lukose, A. Puniyani, and B. Huberman, "Search in power-law networks," *Physical Review*, vol. E64 46135, 2001.

[5] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *The 16th ACM International Conference on Supercomputing*, 2002.

[6] Gnutella, http://www.gnutella.com/.

[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] C. Gkantsidis, M. Mihail, and A. Saberi, "Hybrid search schemes for unstructured peer-to-peer networks," in *IEEE INFOCOM*, 2005.

[9] S. Chen, Z. Zhang, S. Chen, and B. Shi, "Efficient file search in non-DHT P2P networks," *Computer Communications*, vol. 31, pp. 304–317, 2008.

[10] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabasi, "Controllability of complex networks," *Nature*, vol. 473, no. 12, pp. 167–173, 2011.

[11] H. Toyoizumi, S. Tani, N. Miyoshi, and Y. Okamoto, "Reverse preferential spread in complex networks," *Physical Review E*, vol. 86, no. 2, p. 021103, 2012.

[12] W. Stadje, "The collector's problem with group drawing," *Adv. Appl. Prob.*, vol. 22, pp. 866–874, 1990.

[13] S. Shioda, "Some upper and lower bounds on the coupon collector problem," *Journal of Computational and Applied Mathematics*, vol. 200, no. 1, pp. 154–167, 2007.

[14] M. Shaked and J. Shanthikumar, *Stochastic Orders*. Springer, 2016.

[15] K. Nakamura, J. Harada, and S. Shioda, "Topological properties of networks with a given degree sequence generated by havel-hakimi algorithm," *IEICE Trans. Fundamentals of Electronics, Communications and Computer Sciences*, vol. J92-A, no. 9, pp. 604–612, 2009.

[16] "Stanford large network dataset collection," http://snap.stanford.edu/data/.