A Decision-Theoretic Model for Decentralised Query Routing in Hierarchical Peer-To-Peer Networks

Henrik Nottelmann and Norbert Fuhr

Department of Informatics, University of Duisburg-Essen,
47048 Duisburg, Germany, norbert.fuhr@uni-due.de

Abstract. Efficient and effective routing of content-based queries is an emerging problem in peer-to-peer networks, and can be seen as an extension of the traditional "resource selection" problem. The decision-theoretic framework for resource selection aims, in contrast to other approaches, at minimising overall costs including e.g. monetary costs, time and retrieval quality. A variant of this framework has been successfully applied to hierarchical peer-to-peer networks (where peers are partitioned into DL peers and hubs), but that approach considers retrieval quality only. This paper proposes a new model which is capable of considering also the time costs of hubs (i.e., the number of hops in subsequent steps). The evaluation on a large test-bed shows that this approach dramatically reduces the overall retrieval costs.

1 Introduction

Peer-to-peer (P2P) networks have emerged recently as an alternative to centralised architectures. The major problem in such networks is query routing, i.e. deciding to which other peers the query has to be sent for high efficiency and effectiveness. In contrast to the traditional resource selection problem, this process is inherently decentralised in peer-to-peer networks and based on local knowledge.

The decision-theoretic framework (DTF) [7, 3] computes an optimum selection based on cost estimations. These cost estimations include several important factors like retrieval quality, time or money. A user can weight these cost sources for specifying her own selection policy, e.g. preferring cheap digital libraries (DLs), or high quality DLs. Resource descriptions, i.e. statistical aggregation of the DLs, are employed for estimating costs, in particular to approximate the retrieval quality.

[8] presents a heuristic extension of the DTF for hierarchical peer-to-peer networks. In such a P2P topology, peers are partitioned into low-end DL peers hosting the documents, and hubs which act as directory peers. Only hubs are responsible for routing; a DL receiving a query only returns result documents but does not forward the query to other peers. Costs for hubs are computed by simply aggregating the resource descriptions of all DLs in a certain neighbourhood of that hub, by assuming that these DLs are merged into a single virtual collection. This approach, however, does not allow to estimate time costs properly: Those costs depend on the peers a selected hub itself selects, and thus cannot be estimated via simple aggregated statistics.

To appear in: Proceedings ECIR '07.

In this paper, we present the first theoretical model for decentralised query routing in hierarchical P2P networks which considers time costs, and we give experimental results demonstrating the validity of this model. The basic idea is to estimate the costs of the DLs a neighbour hub would select in subsequent phases. This estimation is based on statistical aggregations of the DLs' content, as well as the distances of the DLs to the hub. The advantages are two-fold: Hub costs are only based on those DLs the hub potentially selects (i.e., can contribute to the final result). Second, this approach allows us to take the distances of selected DLs (and, thus, the associated time costs) into account.

This paper is structured as follows: Section 2 briefly describes DTF. An overview over important aspects of resource selection in peer-to-peer networks is given in section 3. Then, section 4 presents a new approach for estimating hub costs which inherently also considers the number of hubs associated with the selection of a hub. An evaluation of the proposed approach is shown in section 5. Section 6 summarises work related to resource selection in distributed IR.

2 The decision-theoretic framework for resource selection

Most resource selection approaches (e.g. CORI) only consider retrieval quality. As an approximation, they compute a similarity score of each DL to the query, and select a fixed number of top-ranked libraries. Other aspects like execution time of the DLs are neglected.

In contrast, the decision-theoretic framework (DTF) [7, 3] is capable of dealing with different selection criteria, which are unified under the notion of "costs". As the actual costs are unknown in advance, expected costs (for digital library DL_i when s_i documents are retrieved for query q) are regarded instead.

Different sources can be considered:

Effectiveness: Probably most important, a user is interested in getting many relevant documents. In a simple model, effectiveness costs are based on the expected number $s_i - E[r_i(s_i,q)]$ of non-relevant documents, where $E[r_i(s_i,q)]$ denotes the expected number of relevant documents among the s_i top-ranked documents.

Time: We assume uniform costs for transmitting a result document, thus these costs can be neglected for selection. As a consequence, the expected costs $EC_i^t(s_i)$ are based on the initial costs for contacting a DL.

Money: Monetary costs are important for some applications, but can be neglected in the context of this paper.

These cost sources are weighted by user-specific parameters c^e (for effectiveness) and c^t (time) parameters. They allow a user to specify her own selection policy, e.g. good results vs. fast results. Thus, the expected costs (for digital library DL_i when s_i documents are retrieved for query q) are computed as:

$$EC_i(s_i,q) := c^e \cdot [s_i - E[r_i(s_i,q)]] + c^t \cdot EC_i^t(s_i) \ . \tag{1}$$

A user also specifies the total number n of documents to be retrieved out of m libraries, and the task is to compute an optimum solution (employing the algorithm presented in [3]):

$$s := \operatorname*{argmin}_{\sum_{i=1}^{m} s_i = n} \sum_{i=1}^{m} EC_i(s_i, q).$$

Relevance costs are computed in two steps:

1. First, the expected number $E(\mathrm{rel}|q,DL)$ of relevant documents in the library is computed based on statistical aggregations (called "resource description") of the DL.
2. Then, a linearly decreasing approximation of the recall-precision function is used for computing the expected number $E[r_i(s_i,q)]$ of relevant retrieved documents.

For the first step, the resource descriptions store the DL size $|DL|$ and the average (expectation) $\mu_t = E[w(d,t)|d \in DL]$ of the indexing weights $w(d,t)$ (for document d and term t). For a query with term weights $a(q,t)$ (summing up to one) and a linear retrieval model, the expected number $E(\mathrm{rel}|q,DL)$ of relevant documents in DL w. r. t. query q can be estimated as:

$$\begin{aligned} E(\mathrm{rel}|q,DL) &= \sum_{d \in DL} Pr(\mathrm{rel}|q,d) \approx \sum_{d \in DL} \sum_{t \in q} a(q,t) \cdot w(d,t) \\ &= |DL| \cdot \sum_{t \in q} a(q,t) \cdot \sum_{d \in DL} \frac{w(d,t)}{|DL|} \\ &= |DL| \cdot \sum_{t \in q} a(q,t) \cdot \mu_t \ , \end{aligned} \tag{2}$$

where $Pr(\mathrm{rel}|q,d)$ denotes the probability that document d is relevant.

In a second step, $E(\mathrm{rel}|q,DL)$ is mapped onto the expected number $E[r_i(s_i,q)]$ of relevant retrieved documents. Assuming a linearly decreasing recall-precision function $P:[0,1] \rightarrow [0,1]$, $P(R) := 1-R$, with expected precision $E[r_i(s_i,q)]/s_i$ and expected recall $E[r_i(s_i,q)]/E(\mathrm{rel}|q,DL_i)$, we can estimate the number of relevant documents when retrieving s_i documents:

$$E[r_i(s_i,q)] := \frac{E(\mathrm{rel}|q,DL_i) \cdot s_i}{E(\mathrm{rel}|q,DL_i) + s_i}. \tag{3}$$

For DTF, the libraries have to return the probabilities of relevance of the result documents, thus no further normalisation step is required.

3 Resource selection in peer-to-peer networks

Query routing (i.e., resource selection) is a crucial task in peer-to-peer networks, as contacting all connected peers does not scale [9]. This section introduces several competing approaches for resource selection in peer-to-peer networks.

3.1 Network topologies

A direct neighbour $P \in nb(P')$ of a peer P' is another peer P if and only if there is a (direct) connection link. The distance between two peers is the minimum number of hops (i.e., links) required to go from one peer to the other.

In this paper, we regard hierarchical peer-to-peer topologies, which are based on a partition of peers into DL peers (sometimes also called "leaves") and hubs. DLs are typically end-user machines which answer but do not forward queries, while hubs are responsible for routing and, thus, high-bandwidth computers which are nearly permanently online. Each DL peer is connected to at least one hub but not to other DL peers, which reduces the number of messages during query routing (i.e., resource selection). This results in a simple yet reasonable and efficient topology, called hierarchical peer-to-peer networks.

In this paper, we focus on HyperCube graphs (HyperCubes for short) [11] . A (binary) HyperCube is a regular d-dimensional structure, where each peer is connected to exactly d other peers (one per dimension). Messages arriving via a connection on dimension $k \in \{0, 1, \ldots, d-1\}$ can only be forwarded to peers on strictly higher dimensions $k' > k$. A consequence is that the dimensions define (starting from an arbitrary peer) a spanning tree on the network, which ensures that there is exactly one path from one peer to another peer. It also corresponds to a clearly defined partition of the whole network.

Positions of missing peers are filled with "virtual peers" (see [11] for details), which are then replaced by "shortcuts" to all original peers which can be contacted through virtual peers only. As a consequence, peers can be connected to more or less than d neighbours.

In this paper, we also ensure that each DL is connected to exactly one hub, so that (given the HyperCube) there is exactly one path from any hub to any DL in the network (i.e., cycles do not occur).

3.2 Centralised and decentralised selection

A simple selection strategy is to use the P2P network for a "cost estimation collection phase", where the query is flooded in a Gnutella-like way through the hub network. Each hub computes cost estimations of its neighbour DLs, and sends them to the hub starting the routing process. Then, a single central selection is computed, and the selected DL peers are notified directly. This centralised selection strategy yields a global optimum, but is rather inefficient (see [9] for Gnutella, and section 5.2 for hierarchical networks). HyperCubes can improve the cost estimation collection phase as each hub is connected exactly once (and not multiple times).

In contrast, decentralised selection computes a local optimum selection on every hub receiving the query, by considering locally available descriptions of all DLs and hubs in a predefined distance. Thus, a hub decides locally how many documents should be retrieved from neighbour DLs, and how many documents are to be delivered by neighbour hubs (which itself compute a local selection). This decentralised selection method produces an overhead as a cost estimation and selection has to be performed on every hub. On the other hand, this method cuts down the number of hubs that are traversed, and thus saves time and bandwidth. In HyperCubes, hub descriptions are based on disjoint sets of DLs, which should improve the selection accuracy. The following Section describes how resource descriptions for hubs can be computed and employed for decentralised selection.

4 Cost estimation for hubs

A hub description is a representative of the neighbourhood of a hub. Basically, its statistical characteristics are defined by combining the resource descriptions of a set of DL peers.[1]

A naive approach is to combine the documents of all DLs in a neighbourhood in a large "virtual" collection, and use the description of that collection as the hub description [8]. This, however, has two drawbacks: A hub is not a monolithic DL, its selection results in further selection steps which ignores most of the DLs in the neighbourhood. Additionally, time costs (i.e., the number of hops) are not considered in such a setting.

The basic idea presented in this paper is to approximate the selection step in a selected hub: When we estimate the costs of a hub, we assume that the hub selects the best DLs in the neighbourhood (but no hubs), and only consider these selected DLs in the cost estimation. This approach also allows us to estimate the time costs associated with selecting a hub.

4.1 Hub resource descriptions

In the traditional decision-theoretic framework (see section 2), the resource description contains the average indexing weight $\mu_t = E[w(d,t)|d \in DL]$ for each index term t. Time costs can easily be added for DLs by using a constant value (e.g., one) for the one hop to the neighbour DL, by setting $EC_i^t(s_i) = 1$ iff $s_i > 0$, and $= 0$, otherwise.

However, a hub is a representative of a sub-network (a "neighbourhood"), and its selection results in contacting further peers (DLs and hubs) with additional costs for those hops. In addition, a term t can occur in DL peers in different distances for a neighbour hub H, so constant time costs are not sufficient for hubs. In the following, we show how the content of a resource description is modified for hubs, so that time costs can be considered as well.

We start with a simple scenario, where a hub H is connected to m libraries $DL_1, \ldots, DL_m \in nb(H)$. Neighbour hubs $H' \in nb(H)$ are not considered so far, the approach is extended to this case in section 4.3. We further assume that resource descriptions are given for all DL_i, i.e. the average indexing weights $\mu_{t,i} = E(t \leftarrow d|d \in DL_i)$.

Each term t can be regarded as a single-term query. Then, the number $R_i(t) = E(rel|t, DL_i)$ of relevant documents in each DL_i can be easily estimated according to equation (2) as:

$$R_{t,i} = E(rel|t, DL_i) = |DL_i| \cdot \mu_{t,i} \ .$$

[1] In peer-to-peer networks, co-operative peers can be assumed, so query-based sampling is not considered here; each DL returns its description upon request.

The results are rounded to natural numbers, to ease the further processing steps.

For the hub H under consideration, the discrete empirical distribution $Pr(R_t)$ of the number of relevant documents is computed over all neighbour DLs. Let us assume a term t_1 for which 4 of the 6 neighbour DLs have 3 relevant documents for t_1 and the 2 other neighbour DLs have 5 relevant documents. Then, the resulting distribution is defined by $Pr(R_{t_1} = 3) = 4/6$ and $Pr(R_{t_1} = 5) = 2/6$.

This distribution $Pr(R_t)$ forms a compact representation of the content of the hub's neighbourhood, and is used as the resource description of hub H. In other words, for each term t a function $Pr(R_t = \cdot)$ is stored.

4.2 Cost estimation

At query time, the resource description of hub H is employed for estimating retrieval costs. Costs for hubs are estimated in 6 subsequent phases, incorporating only neighbour DLs:

1. For the query, the distribution of relevant documents in all DLs is computed.
2. For all DLs, the number of relevant documents is estimated based on the distribution.
3. The DLs are ranked w. r. t. their number of relevant documents, and the best DLs are selected.
4. Costs are estimated based on the best selected DLs.
5. Minimum costs are computed.

In the first phase, the distribution $Pr(R_q)$ of the relevant documents $R_q = E(\text{rel}|q, DL)$ (for a query q and a randomly chosen library DL) is computed. Remember that with a linear retrieval function, we have:

$$R_q = E(\text{rel}|q, DL) = |DL| \cdot \sum_{t \in q} a(q,t) \cdot \mu_t = \sum_{t \in q} a(q,t) \cdot R_t \ .$$

Thus, the random variable R_q can be considered as the linear combination of the random variables R_t. The distribution $Pr(R_q)$ can thus be computed via convolution:

$$Pr(R_q) = \sum_{R_t :\ R_q = \sum_{t \in q} a(q,t) \cdot R_t} \ \prod_{t \in q} Pr(R_t).$$

Here, basically, the probabilities of all possible cases for the R_t which lead to a fixed value of R_q are summed up, assuming independence of the R_t. (Since the distributions $Pr(R_t)$ are very sparse, the convolution can be computed reasonably efficiently.) As an example, assume—in addition to the distribution for term t_1 (see section 4.1)—a second term t_2 with $Pr(R_{t_2} = 1) = Pr(R_{t_2} = 3) = 1/2$, and further assume $a_{t_1} = a_{t_2} = 1/2$. Then, the case $R_q = 2$ can only be caused by $R_{t_1} = 3$ and $R_{t_2} = 1$ with $Pr(R_q = 2) = 2/3 \cdot 1/2 = 1/3$. Similarly, the case $R_q = 3$ can be caused by either $R_{t_1} = R_{t_2} = 3$ or by $R_{t_1} = 5$ and $R_{t_2} = 1$, thus $Pr(R_q = 3) = 2/3 \cdot 1/2 + 1/3 \cdot 1/2 = 1/2$. Finally, $Pr(R_q = 4) = 2/3 \cdot 1/2 = 1/6$ is caused by the case $R_{t_1} = 5$ and $R_{t_2} = 3$.

In a second step, the frequencies $Pr(R_q)$ and the number m of neighbour DLs are used for estimating the number of relevant documents for the DLs. E.g., for $m = 6$ DLs in total, 2 DLs have 2 relevant document, 3 DL have 3 relevant documents, and the

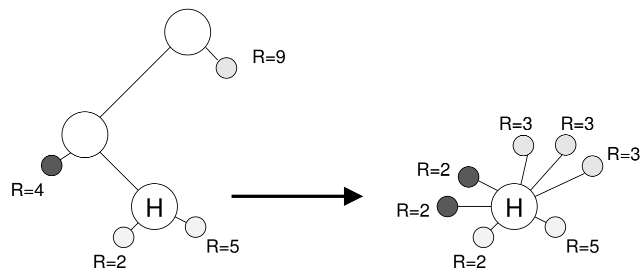

Fig. 1. Hub costs in larger neighbourhoods

sixth DL contains 4 relevant documents. Interpolation is used for computing a value for each DL in cases where the probability for a R_q value does not correspond to a natural number of occurrences.

Third, the DLs are ranked w. r. t. their number of relevant documents, i.e. $R_1 \geq R_2 \geq \ldots \geq R_m$. For $1 \leq l \leq m$, $R(l) := \sum_{i=1}^{l} R_i$ denotes the sum of the number of relevant documents in the top l DLs.

In a fourth step, we assume that a hub is the combination of all l selected DLs, i.e. a hub is considered as a single DL. With the recall-precision function and equation (3), we can compute the number $r(s,R(l))$ of relevant documents in the result set when retrieving s documents from the union of the l selected DLs. Following equation (1), the costs $EC(s,l,R(l))$ when l neighbour DLs are selected can be computed as:

$$EC_H(s,l,R(l)) := c^e \cdot [s - r(s,R(l))]] + c^t \cdot (l+1) \ .$$

Note that the number of hops equals the number of selected neighbour DLs (each one can be reached via one hop in a later phase) plus 1 hop for reaching the hub itself.

The final cost estimations can be easily computed in a sixth step:

$$\begin{aligned} EC_H(s) &= \min\{EC_H(s,l,R(l)) | 1 \leq l \leq m\} \\ &= \min\{c^e \cdot [s - r(s,R(l))]] + c^t \cdot (l+1) | 1 \leq l \leq m\} \ . \end{aligned}$$

These cost estimations can be used in the usual selection process.

4.3 Considering a larger neighbourhood

So far, only neighbour DLs are considered for computing the resource description of a hub. However, hubs typically are connected to other (neighbour) hubs, which themselves have DLs (and, potentially, other hubs) attached. We apply a trick and conceptually replace hubs and their attached DLs by new virtual DLs directly connected to the hub. Thus, the network structure is “flattened”, and costs can be estimated for all DLs in the same way, regardless of their distance.

The horizon h defines the maximum distance between the hub H and the DLs to be considered for its hub description. In section 4.2, only neighbour DLs are considered, which equals to a horizon $h = 1$. For a horizon $h > 1$, the neighbourhood function nb is extended as follows:

$$nb^1(H) := nb(H),$$
$$nb^h(H) := \bigcup_{H' \in nb(H)} nb^{h-1}(H') \; .$$

In other words, $nb^h(H)$ describes all peers in a distance of exactly h hops (from H).

The key idea for considering remote DLs is the following: Costs remain untouched if such a library DL_i' is replaced by two "virtual" DLs connected directly to H, where the relevant documents are uniformly distributed over both DLs (i.e., with $R/2$ relevant documents): To obtain R relevant documents, two hops (one for each new virtual DL) are required. For implementing this scheme, the neighbouring hub H' has to send its cumulated statistics about $DL_i' \in nb(H')$ to hub H. Costs are then estimated as described in section 4.2, without a need for caring about the distance of DL peers. A similar approach is used for $h > 2$, were a DL in distance of $h' \leq h$ is replaced by h' DLs with R/h' relevant documents.

An example is shown in figure 1. Here, the DLs directly connected to the hub H remain untouched. The DL connected to the direct neighbour hub of H (with $R = 4$ relevant documents) is replaced by two virtual hubs with $R = 4/2 = 2$, while the DL with $R = 9$ connected to the neighbour hub of the neighbour hub (i.e., the DLs in $nb^3(H)$) is replaced by three virtual DLs with $R = 9/3 = 3$.

5 Evaluation

The proposed approach has been evaluated on a large test-bed. This sections describes the setup of the experiments and results in terms of efficiency, effectiveness and costs.

5.1 Experimental Setup

The WT10g collection is used as a basis for our experiments. The topology "cmu" (taken from [5]) employes a hierarchical P2P network, where the WT10g collection is divided into 11,485 collections according to the document URLs; 2,500 collections (containing in total 1,421,088 documents) were chosen randomly, each of them forming one DL peer. Hubs are formed by similarity of the DL peers, each hub is connected to 13-1,013 DLs (379.8 on average). Neighbour hubs are selected randomly so that each hub has 1-7 hub neighbours (3.8 on average).

The topology "hc-1" regarded in this paper is a HyperCube derived from the "cmu" topology in the following way: it consists of 25 hubs as in "cmu" (with dimension $d = 5$); each hub is connected to 4–14 other hubs (5.8 on average). Each DL is connected to exactly one hub, randomly chosen out of the "cmu" connections, so that each hub is connected to 3–254 DLs (100 on average). Thus, "hc-1" completely avoids cycles and yields disjoint hub neighbourhoods.

The WT10g collection only provides 100 topics with relevance judgements. For large P2P networks, more queries are required. Thus, we use 1,000 random queries (from the test set) generated from title fields of documents. Each query contains up to 6 terms, the average is 2.9 terms per query. In all experiments, $n = 50$ documents are requested. As these queries were created artificially, no relevance judgements are available. Pseudo-relevance judgements were obtained by combining all 2,500 collections into one centralised collection (using system-wide IDF values), and marking the 50 top-ranked documents[2] for each query as "relevant". Thus, the experiments measure how well distributed retrieval approximates the centralised collection. Document indexing weights are computed based on the BM25 formula [10].

For resource selection, we set $c^e = 1$ in all cases (for effectiveness costs, i.e. the number of non-relevant documents), and vary the parameter c^t to simulate different user preferences. Similarly, for computing costs after retrieval (the actual costs connected to the query) we set $c^e = 1$ and use varying parameters for the time component. Please note that for $c^t = c^e = 1$, one hop corresponds to one non-relevant document. For the same number of documents, selecting an additional DL thus can only be compensated if that DL returns an additional relevant document. Similarly, for $c^t = \frac{c^e}{4} = 0.25$, four hops correspond to one relevant document.

For result merging, we assume that a hub propagates the hub-local idf values[3] to its directly connected DLs, and then merge the returned ranking lists according to descending RSVs.

5.2 Results

Table 1 depicts the results for our HyperCube topology. First, the tables show that centralised selection ("cent.") outperforms decentralised variants ("dec.") in terms of effectiveness. Compared to $c^t = 0$ (ignoring time costs), precision in the top ranks decreases with increasing time costs c^t (for a single hop). Precision in lower ranks, mean average precision (MAP) as well as set-based precision and recall, however, increase up to $c^t = 0.5$ (for $h = 1$) or $c^t = 1$ (for $h = 3$ and the set-based values), before these values decrease again. As intended, efficiency dramatically increases (i.e., less hubs and DLs are selected) with increasing c^t. Both effects nearly balance so that the overall costs only slightly increase.

The table also reveals that for any fixed time cost user parameter c^t, effectiveness increases with a larger horizon h. In other words, a larger hub neighbourhood (with more DLs considered) improves the cost estimation process. This fact shows that our model makes good use of the knowledge provided. As also can be seen from these figures, decentralised selection with time costs considered outperforms the two other approaches.

As a summary, incorporating time costs in the selection process dramatically reduces the final costs w. r. t. the user's preference. Moreover, the approach is capable of adjusting to increasing time costs per hop c^t so that the final costs increase only marginally. In addition, broadening the horizon leads to increased retrieval quality and marginally lower costs.

[2] Documents are ranked using the same indexing weights and retrieval functions as all DLs.

[3] In order to reduce the experimental effort, we used system-wide idf values in our experiments, since earlier experiments [8] showed that the difference between system-wide and hub-local idf values is negligible.

(a) $h = 1$

	Dec., $c^t = 0$	Cent., $c^t = 0$	Dec., $c^t = 0.1$	Dec., $c^t = 0.25$	Dec., $c^t = 0.5$	Dec., $c^t = 1$
P@10	0.4385 / 0.0%	0.6586 / 50.2%	0.4241 / -3.3%	0.4044 / -7.8%	0.3527 / -19.6%	0.2981 / -32.0%
P@30	0.2087 / 0.0%	0.3774 / 80.8%	0.2171 / 4.0%	0.2206 / 5.7%	0.2072 / -0.7%	0.1856 / -11.1%
MAP	0.1307 / 0.0%	0.2565 / 96.3%	0.1387 / 6.1%	0.1450 / 10.9%	0.1399 / 7.0%	0.1299 / -0.6%
Precision	0.1571 / 0.0%	0.2688 / 71.1%	0.1677 / 6.7%	0.1751 / 11.4%	0.1700 / 8.2%	0.1597 / 1.6%
Recall	0.1354 / 0.0%	0.2576 / 90.3%	0.1435 / 6.0%	0.1493 / 10.3%	0.1438 / 6.2%	0.1325 / -2.1%
#Hops	40.9 / 0.0%	55.5 / 35.5%	23.6 / -42.4%	14.9 / -63.6%	9.8 / -76.0%	6.6 / -83.8%
Costs	35.78 / 0.0%	34.52 / -3.5%	37.54 / 4.9%	38.63 / 8.0%	40.00 / 11.8%	41.98 / 17.3%

(b) $h = 2$

	Dec., $c^t = 0$	Cent., $c^t = 0$	Dec., $c^t = 0.1$	Dec., $c^t = 0.25$	Dec., $c^t = 0.5$	Dec., $c^t = 1$
P@10	0.5694 / 0.0%	0.6586 / 15.7%	0.5560 / -2.4%	0.5278 / -7.3%	0.4797 / -15.8%	0.4198 / -26.3%
P@30	0.3141 / 0.0%	0.3774 / 20.2%	0.3300 / 5.1%	0.3301 / 5.1%	0.3167 / 0.8%	0.2920 / -7.0%
MAP	0.2066 / 0.0%	0.2565 / 24.2%	0.2206 / 6.8%	0.2232 / 8.0%	0.2194 / 6.2%	0.2079 / 0.6%
Precision	0.2360 / 0.0%	0.2688 / 13.9%	0.2520 / 6.8%	0.2559 / 8.4%	0.2536 / 7.4%	0.2448 / 3.7%
Recall	0.2139 / 0.0%	0.2576 / 20.5%	0.2278 / 6.5%	0.2295 / 7.3%	0.2255 / 5.4%	0.2136 / -0.1%
#Hops	45.2 / 0.0%	55.5 / 22.7%	23.3 / -48.4%	14.9 / -67.1%	10.1 / -77.7%	7.2 / -84.2%
Costs	34.10 / 0.0%	34.52 / 1.3%	35.73 / 4.8%	36.84 / 8.1%	38.24 / 12.2%	40.61 / 19.1%

(c) $h = 3$

	Dec., $c^t = 0$	Cent., $c^t = 0$	Dec., $c^t = 0.1$	Dec., $c^t = 0.25$	Dec., $c^t = 0.5$	Dec., $c^t = 1$
P@10	0.6307 / 0.0%	0.6586 / 4.4%	0.6284 / -0.4%	0.6048 / -4.1%	0.5618 / -10.9%	0.4987 / -20.9%
P@30	0.3757 / 0.0%	0.3774 / 0.5%	0.4012 / 6.8%	0.4045 / 7.7%	0.3939 / 4.8%	0.3652 / -2.8%
MAP	0.2541 / 0.0%	0.2565 / 0.9%	0.2765 / 8.8%	0.2822 / 11.1%	0.2816 / 10.8%	0.2688 / 5.8%
Precision	0.2810 / 0.0%	0.2688 / -4.4%	0.3056 / 8.8%	0.3125 / 11.2%	0.3142 / 11.8%	0.3036 / 8.0%
Recall	0.2625 / 0.0%	0.2576 / -1.8%	0.2844 / 8.3%	0.2891 / 10.1%	0.2892 / 10.2%	0.2752 / 4.8%
#Hops	45.0 / 0.0%	55.5 / 23.3%	22.0 / -51.0%	14.4 / -68.1%	10.1 / -77.6%	7.6 / -83.2%
Costs	33.08 / 0.0%	34.52 / 4.4%	34.13 / 3.2%	35.17 / 6.3%	36.62 / 10.7%	39.64 / 19.8%

Table 1. Results for centralised and decentralised resource selection

6 Related work

In contrast to the decision-theoretic framework (DTF) employed in this paper, most of the other selection algorithms compute a score for every library. Then, the top-ranked documents of the top-ranked libraries are retrieved and merged in a data fusion step.

The GlOSS system [4] is based on the vector space model and – thus – does not refer to the concept of relevance. For each library, a goodness measure is computed which is the sum of all scores (in the experiments reported, SMART scores) of all documents in this library w. r. t. the current query. Libraries are ranked according to the goodness values.

The state-of-the-art system CORI [1] uses the INQUERY retrieval system which is based on inference networks. The resource selection task is reduced to a document retrieval task, where a “document” is the concatenation of all documents of one library. The indexing weighting scheme is quite similar to one employed in DTF, but applied to libraries instead of documents. Thus, term frequencies are replaced by document frequencies, and document frequencies by collection frequencies. CORI also covers the data fusion problem, where the library score is used to normalise the document score. Experiments showed that CORI outperforms GlOSS [2].

Another ranking approach is based on language models [13]. Basically, the language model of the collection is smoothed with a collection-independent (system-wide) language model, and KL-divergence is used for ranking the DLs. The final document ranking is computed in a result merging step by using the original (collection-biased) document probabilities, the DL scores, a smoothing factor, and Bayesian inversion. The quality of this approach is slightly better than CORI.

The language model approach has been extended towards hierarchical peer-to-peer networks in [6] for ranking neighbour peers (leaves and hubs). Hubs are described by neighbourhood (which is not limited to the directly connected DLs), where the influence of term frequencies of distant DLs is exponentially decreased. DLs and hubs are selected separately, as DL and hub descriptions are not in the same order of magnitude. A fixed number of hubs is selected, while a modified version of the semi-supervised learning algorithm [12] is employed for computing a threshold for the number of selected leaves.

The decision-theoretic framework has been extended towards peer-to-peer networks in [8]. There, an extensive discussion of resource selection architectures for peer-to-peer networks is presented. The architectures are classified based on the underlying resource selection approach (DTF and CORI as a baseline), design choices like the locality of knowledge (e.g. IDF values) and selections (centralised vs. decentralised), as well as the network topology (hierarchical networks with DLs and hubs, distributed hash tables and HyperCubes). Time costs, however, are not regarded there. The evaluation shows that DTF slightly outperforms CORI in peer-to-peer networks. Centralised selection has higher effectiveness than decentralised selection, but has an expensive cost estimation collection phase. Distributed hash tables [14] and HyperCubes can reduce that effort.

7 Conclusion and outlook

This paper presents the first theoretical model for decentralised query routing in hierarchical peer-to-peer networks, which also incorporates time costs (in addition to traditional retrieval quality measures). For this, the decision-theoretic framework has been extended to estimate the costs of DLs a neighbour hub would select in subsequent phases. This estimation is based on statistical aggregations of the DLs' content, as well as the distance of the DLs to the hub. The advantages are two-fold: Hub costs are only based on those DLs which the hub potentially selects (and, thus, can contribute to the final result). Second, this approach allows us to take the distance of selected DLs (and, thus, the associated time costs) into account.

The evaluation shows that the new P2P variant of the decision-theoretic framework is capable to optimise the selection quality when time costs are considered. The final costs (w. r. t. the user's preference) are dramatically reduced. Moreover, the approach can adjust to increasing time costs c^t per hop so that the final costs increase only marginally. Furthermore, broadening the horizon leads to increased retrieval quality and marginally lower costs.

Here we have tested our model under optimum conditions, in order to demonstrate its general validity. Future work will concentrate on the development of approximations for less favourable settings. First, we will replace the empirical term distributions $Pr(R_t)$ by appropriate theoretical distributions. Another issue is the reduction of the required knowledge about the neighbourhood when constructing resource descriptions. Currently, each hub has to provide separate statistics of all DLs in distance 1, for all DLs in distance 2, and so on. As an alternative, approximate aggregated descriptions will be investigated. In a similar way, we will work on modifications of the approach for effectively dealing with cycles in the network.

References

[1] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In E. A. Fox, P. Ingwersen, and R. Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, New York, 1995. ACM. ISBN 0-89791-714-6.

[2] J. French, A. Powell, J. Callan, C. Viles, T. Emmitt, K. Prey, and Y. Mou. Comparing the performance of database selection algorithms. In *Proceedings of the 22nd International Conference on Research and Development in Information Retrieval*, pages 238–245, New York, 1999. ACM.

[3] N. Fuhr. A decision-theoretic approach to database selection in networked IR. *ACM Transactions on Information Systems*, 17(3):229–249, 1999.

[4] L. Gravano and H. Garcia-Molina. Generalizing GIOSS to vector-space databases and broker hierarchies. In U. Dayal, P. M. D. Gray, and S. Nishio, editors, *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases*, pages 78–89, Los Altos, California, 1995. Morgan Kaufman.

[5] J. Lu and J. Callan. Content-based retrieval in hybrid peer-to-peer networks. In D. Kraft, O. Frieder, J. Hammer, S. Qureshi, and L. Seligman, editors, *Proceedings of the 12th International Conference on Information and Knowledge Management*, New York, 2003. ACM.

[6] J. Lu and J. Callan. Federated search of text-based digital libraries in hierarchical peer-to-peer networks. In J. Callan, N. Fuhr, and W. Nejdl, editors, *SIGIR Workshop on Peer-to-Peer Information Retrieval*, 2004.

[7] H. Nottelmann and N. Fuhr. Evaluating different methods of estimating retrieval quality for resource selection. In J. Callan, G. Cormack, C. Clarke, D. Hawking, and A. Smeaton, editors, *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, 2003. ACM.

[8] H. Nottelmann and N. Fuhr. Comparing different architectures for query routing in peer-to-peer networks. In *ECIR*, pages 253–264. Springer, 2006.

[9] J. Ritter. Why Gnutella can't scale. No, really., 2001. `http://www.darkridge.com/~jpr5/doc/gnutella.html`.

[10] S. E. Robertson, S. Walker, M. Hancock-Beaulieu, A. Gull, and M. Lau. Okapi at TREC. In *Text REtrieval Conference*, pages 21–30, 1992.

[11] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. Digital libraries. In *1st Workshop on Agents and P2P Computing*, 2005.

[12] L. Si and J. Callan. A semi-supervised learning method to merge search engine results. *ACM Transactions on Information Systems*, 24:457–49, 2003.

[13] L. Si, R. Jin, J. Callan, and P. Ogilvie. Language modeling framework for resource selection and results merging. In C. Nicholas, D. Grossman, K. Kalpakis, S. Qureshi, H. van Dissel, and L. Seligman, editors, *Proceedings of the 11th International Conference on Information and Knowledge Management*, New York, 2002. ACM. `http://www-2.cs.cmu.edu/~callan/Papers/cikm02-lsi.pdf`.

[14] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM*, 2001. `http://www.acm.org/sigcomm/sigcomm2001/p12-stoica.pdf`.