User-Centric Operational Decision-Making in Distributed Information Retrieval

Kartik Hosanagar[+]

Abstract

Information specialists in enterprises regularly use Distributed Information Retrieval (DIR) systems that query a large number of Information Retrieval (IR) systems, merge the retrieved results and display them to users. There can be considerable heterogeneity in the quality of results returned by different IR servers. Further, since different servers handle collections of different sizes, have different processing and bandwidth capacities, there can be considerable heterogeneity in their response times. The broker in the distributed IR system has to decide which servers to query, how long to wait for responses and which retrieved results to display based on the benefits and costs imposed on users. The benefit of querying more servers and waiting longer is the ability to retrieve more documents. The costs may be in the form of access fees charged by IR servers or user's cost associated with waiting for the servers to respond. We formulate the broker's decision problem as a stochastic mixed integer program and present analytical results for the optimal query set and wait time. Using data gathered from Fedstats – a system that queries IR engines of several US federal agencies – we demonstrate that the technique can significantly increase the utility from DIR systems. Finally, we present a simulation-based optimization technique to solve the broker's decision problem under more complex decision environments. The technique is computationally efficient and can be used to generate decision rules for source selection and query termination that are relatively easy to implement.

Keywords: distributed Information Retrieval (IR); personalization; utility theory, optimal operational decisions; source selection, query termination, stochastic modeling

---

[+] Operations and Information Management, The Wharton School of the University of Pennsylvania; email: kartikh@wharton.upenn.edu

Acknowledgements: Luo Si provided valuable guidance regarding design of Fedstats' DIR system. In addition, I would like to thank Monique Guignard-Spielberg, Ramayya Krishnan, Gautam Pant, participants of the 2005 ACM SIGIR conference and anonymous referees for helpful discussions on prior versions of this work.

1. Introduction

The burgeoning of the information age has been accompanied by an explosive growth in the amount of information being generated and stored electronically. Access to this information is crucial to various information-intensive firms. Information specialists use multiple information sources to respond to information requests within these firms. For example, a patent metasearch system may be used to query several distributed patent databases such as USPTO[1] and WIPO[2]. Similarly, a portal called Fedstats[3] is often used to access statistics from over 100 US federal agencies including NIH, USDA and census bureau. Similar systems are also deployed in law firms and financial institutions to provide centralized access to multiple data collections. Other examples include library management systems that provide access to multiple distributed digital libraries and comparison shopping engines that may query multiple store websites in real time to gather price and product information[4]. These systems belong to a general class of Information Retrieval (IR) systems called Distributed IR (DIR) systems.

In a DIR system, a broker queries multiple distributed data sources to gather relevant information in response to a query. These distributed data sources may each be IR systems. Given a query, the goal of each of these IR systems is to identify and display the local documents most relevant to the query. The objective of the DIR system is to provide a user with unified access to all relevant resources on the network but give the impression of a single large IR database (Fuhr 1999).

Key operational issues that must be addressed during a distributed IR task include which data sources or IR servers to query, how long to wait for responses, and which results to display

---

[1] US Patent & Trademark Office http://www.uspto.gov/

[2] World Intellectual Property Organization www.wipo.int

[3] http://www.fedstats.gov/

[4] Some comparison shopping engines cache all the price and product information locally and only query a single local database, while others may query multiple store websites in real time. There are pros and cons with each approach. In this paper, we focus only on distributed Information Retrieval (IR) systems.

(Baeza-Yates and Ribiero-Neto 1999; Fuhr 1999; Montgomery et al 2004). Data sources need to be selected carefully because there can be considerable heterogeneity in the quality of results returned by different IR servers and the access fee charged by them. Furthermore, each of the IR servers has considerable processing to do locally in response to a query which can result in high response times. Hence, a broker may find it optimal to terminate a search even before all queried servers have responded if it believes that the user's benefit from waiting is outweighed by the cost of waiting. Finally, the broker must determine which of the retrieved results to display. These operational decisions can have a significant impact on user's utility. For example, a recent survey of users of patent metasearch systems identified comprehensive coverage and slow response times as two major issues with current systems.[5] These are both impacted by the broker's operational decisions.

In this paper, we address optimal operational decisions – which servers to query, how long to wait for responses and which retrieved results to display - by brokers in distributed IR by taking into account user preferences and historical performance of the distributed sources. We formulate the broker's decision problem as a stochastic mixed integer program and present an analytical solution. We illustrate its application using data from a real-world DIR context and find that the gains from the technique can be significant. Finally, we present an algorithmic solution technique to address more complex formulations in which the expected benefit from a server is a function of which other servers are queried. The simulation-based technique is computationally efficient and offers very good solutions in practice.

Our research contributes to two distinct streams– the design of distributed IR systems in the computer science community and user preference modeling in electronic environments in the

---

[5] Source: Patsnap Inc

IS/Marketing communities. While a number of interesting technical challenges in the design of DIR systems have been addressed by IR researchers, user models are often absent or not very sophisticated. Our research represents a novel application of Utility theory to IR and bridges utility-centric considerations commonly studied in management/marketing science with computational aspects commonly analyzed in IR research. By incorporating information on user preferences, the design of DIR systems can be considerably improved. In turn, this will increase user satisfaction and help increase usage of these systems.

The rest of this paper is organized as follows. In Section 2, we review related work. In Section 3, we develop a decision theoretic formulation to model the tradeoffs and present an analytical solution to the problem. In Section 4, we apply data from a real-world DIR application and evaluate the performance improvement that optimal decision-making can provide. Section 5 presents a simulation-based solution technique for more complex decision environments. Section 6 concludes the study and discusses future work.

2. Prior Work

The IR field has been highly interdisciplinary, drawing from library and information sciences, computer science, and statistics. Some areas of interest include IR models for locating and ranking relevant documents, distributed IR, human interaction, filtering, clustering, question answering, and multimedia IR. Distributed IR is specifically concerned with the challenges of retrieval from distributed data sources. Below, we review three streams of work most relevant to this paper – 1) Operational decisions in DIR with a special emphasis on decision-theoretic approaches 2) Management Science research on user preferences and costs and 3) Management Science applications in heterogeneous Information Systems.

Operational decisions in DIR: The process of determining the servers to query is termed source selection. Callan et al (1995) represent each server by its terms and document frequencies to rank order and select the servers. Other popular source selection techniques include gGIOSS resource ranking algorithm (Gravano and Garcia-Molina 1995) and ReDDE (Si and Callan 2003). Once results have been retrieved from different sources, they need to be merged. When the data sources are cooperative, the results can be merged based on the server-specific relevance scores and normalizing statistics provided by the servers. In non-cooperative environments, more sophisticated techniques including regression based techniques (Le Calve and Savoy 2000; Si and Callan 2002) and Bayesian models (Aslam et al 2001) are used. These techniques involve offline analysis of the IR servers during a resource representation phase. The analysis is used to develop decision rules for merging retrieved results in a fast manner.

The most relevant papers in this stream are the ones applying decision-theoretic approaches. These include work by Fuhr (1999) on a decision-theoretic approach to source selection and by Voorhees (1995) on an approach to select sources and merge results based on historical data. Etzioni et al. (1996) also study the optimal sequence in which to query information sources in a sequential query problem where the broker pays each information source in order to query it. Si and Callan (2004) propose a deterministic Dynamic Programming (DP) based algorithm for source selection.

Our paper complements this stream of work but introduces an important perspective. We develop a model of user preferences and introduce a utility-theoretic framework to guide the decisions. Models of user preferences have been largely absent in the IR literature. Montgomery et al. (2004) also integrate computational and behavioral considerations to study operational decisions made by shopbots. However, the solutions were derived for the case in which servers have i.i.d response time and i.i.d utilities which is a restrictive assumption for general DIR systems where servers can be highly heterogeneous. Further, the work did not focus on developing an operational algorithm for decision making and the results were specifically for the shopbot context and do not generalize to distributed IR. In contrast, our objective is to solve the decision problem for a broker in a DIR system, to account for server heterogeneity and to develop an operational algorithm that can be implemented in a computationally efficient manner.

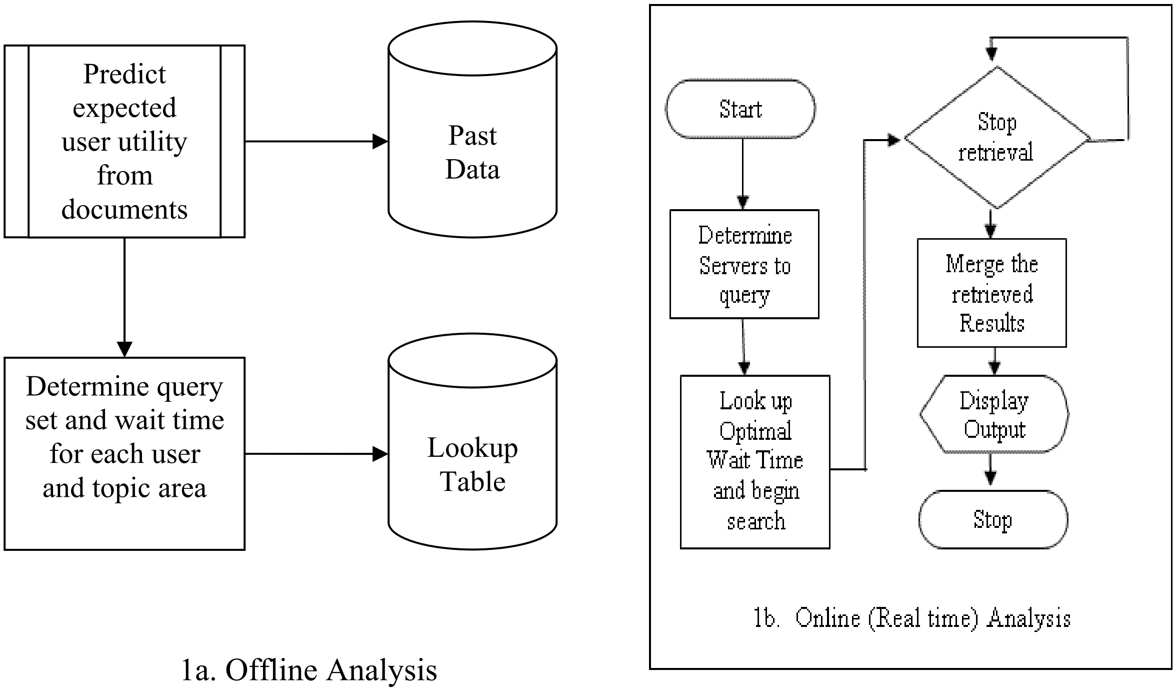
Management Science research on user preferences and costs: Management Science research has a lot to contribute in terms of modeling user preferences for distributed IR tasks. Research in marketing has studied the impact of waiting time on consumer perception of services (Hui and Tse 1996). User studies have shown that consumers incur costs in waiting for websites to respond (e.g., Dellaert and Kahn 1999; Ivory and Hearst 2002). Similarly, consumer research has identified that users incur costs in evaluating information and the cognitive resources needed to do so influence the amount of information users are able to process. Previous studies (Chase 1978; Johnson and Payne 1985) have tried to decompose the cognitive effort into units of elementary information processes and Shugan (1980) has proposed a metric for the cognitive cost based on these elementary processes. While this stream of work has identified and estimated waiting and cognitive costs that are highly relevant to web-based systems, there has been very limited work that incorporates these considerations into the operational decisions made by a system.

Management Science research on heterogeneous Information Systems (IS): Prior work in IS has studied decision models to address operational issues in heterogeneous IS. Krishnan et al (2001) propose a cognitively-guided approach to query heterogeneous databases and propose mathematical models for optimal source identification. Dey et al (1998) present a decision-theoretic model for entity matching across heterogeneous databases, wherein the same entity may be represented differently in different databases. Dey (2003) extends that work by presenting a decision model and a heuristic solution approach to match entities in a data warehouse consisting of several distributed data sources. We complement this stream of work by presenting a decision model and analytical solution for optimal source selection and query termination in DIR.

3. Decision Theoretic Framework

To fix a context, we consider a DIR deployment in an enterprise setting wherein the IS manager is interested in maximizing the expected surplus from the system. Specifically, the broker makes operational decisions in order to maximize the expected surplus for any given query. Further, we assume that each of the individual IR systems index different collections with non-overlapping documents. Figure 1 (1a, 1b) illustrates the framework we propose for the broker's operational decisions. First, the broker analyzes past data on distribution of response times and relevance scores of documents retrieved from various servers and generates decision rules for source selection and query termination (Figure 1a). These rules identify the servers to query and the wait time for each query class and user. A query class is a topic area (e.g., "public policy") with query complexity information (e.g., a simple query may be defined as a non-boolean query with less than five terms per query). The user can be an individual user of the DIR system or a class of users that have been identified to be similar. The specific choice of whether individual-level or segment-level customization is done will depend on the amount of data available per user and computational costs associated with processing user information. We discuss this issue further in Section 6. When a query is received, the broker identifies the query class and user segment and then determines which servers to query and how long to wait for responses (Figure 1b). These decisions are guided by the decision rules from the prior offline analysis. Finally, the broker merges the results and displays the same to the user.

Figure 1: Decision process for the broker


Predict expected user utility from documents
Past Data
Determine query set and wait time for each user and topic area
Lookup Table
Start
Determine Servers to query
Look up Optimal Wait Time and begin search
Stop retrieval
Merge the retrieved Results
Display Output
Stop

1b. Online (Real time) Analysis

1a. Offline Analysis

3.1 Notation and Assumptions

Let N denote the total number of servers that can be queried. $\boldsymbol{q} = (q_1, q_2, ..., q_N)$ is a vector that denotes the servers queried, with q_i=1 if server i is queried and q_i=0 otherwise. t_i denotes the response time of server i. The response time of servers cannot be predicted precisely, i.e., t_i is a stochastic variable. $F_i()$ denotes the probability distribution function (i.e. cdf) for the response time of server i. T denotes the broker's wait time. The vector $\boldsymbol{r} = (r_1, r_2, ..., r_N)$ records the servers retrieved, with r_i=1 if q_i=1 and $t_i \leq T$. r_i=0 otherwise. For any given choice of q and T, there exists a probability distribution over $\boldsymbol{r}$. For example, when N=2 and both servers are queried (q = (1,1)), then $\boldsymbol{r}$ can have four possible values, i.e., $\boldsymbol{r} \in$ {(0,0), (0,1), (1,0), (1,1)}. The probability of retrieving a vector ***r*** with $r_i \le q_i, \forall i,$ (i.e., retrieval vectors that satisfy the condition that only queried servers are retrieved) is $\prod_i F_i(T)^{r_i}(1-F_i(T))^{1-r_i}$.

d_i denotes the number of documents returned by server *i* when it is retrieved. We model the retrieval of documents from individual servers as a batch process as is common with most IR systems. That is, either all d_i documents are retrieved or none are retrieved. The total number of servers queried is denoted *Q* and the total number of documents retrieved is denoted *D* ($\sum q_i = Q \le N$, $D = \sum r_i \cdot d_i$). The user derives some utility from the information but incurs costs associated with waiting for responses and evaluating the results. Further, the servers may impose a fee per query. We now introduce notation to model these benefits and costs.

Utility from Information: The user's utility from a document is a function of various attributes including relevance, novelty and credibility (Moenart and Souder 1996; Larcker and Lessig 1980). This is consistent with the design of real-world IR systems that compute the relevance score accounting for factors such as document relevance and credibility of the source. We therefore use the terms relevance score and utility interchangeably.

At the time the broker queries the servers the utility of documents that will be returned by a server are not known and are hence treated as random variables. Once documents are retrieved from the servers, the utility i.e. the relevance scores from the retrieved documents, can be computed by the broker. Accordingly, we assume that before the documents are retrieved, the broker only knows the probability distribution function $G_{ji}(\cdot)$ of user *j*'s utility from a document returned by server *i* ($g_{ji}(\cdot)$ is the corresponding pdf). These can be determined by the broker based on past queries. Once documents are retrieved, the relevance scores are known. Given *D* retrieved documents, $U_{j,k:D}$ is used to denote user *j*'s utility from the document ranked *k* in a list of documents sorted by relevance score. That is, $U_{j,D:D} \geq U_{j,D-1:D} \geq ... \geq U_{j,2:D} \geq U_{j,1:D}$. $\boldsymbol{U_j} = (U_{j,D:D}, U_{j,D-1:D}, \ldots, U_{j,1:D})$ denotes the relevance scores of the retrieved documents. We assume that the broker displays the documents in decreasing order of relevance score. This reflects a common practice in IR systems and is consistent with the Probability Ranking Principle (PRP) in IR (Robertson 1977). Finally, in order to evaluate the user's benefit from the displayed information, we need to understand the user's stopping criterion when evaluating the displayed results. Here, as in Fuhr (1999), we assume that the user views the top P documents sequentially. The utility to the user from the top P results given D documents are retrieved is given by $\sum_{k=1}^{P} U_{j,D-k+1:D}$. The sum of individual utilities specification is commonly assumed in the literature (Si and Callan 2004; Fuhr 1999). However, P is endogenously determined in our model, i.e., it depends on the quality of the documents displayed.

Cost of Waiting for Responses: The total waiting time for the user is primarily composed of the broker wait time and network latency. The network latency cannot be significantly influenced by the broker's operational decisions.[6] Thus, we drop network latency for the purposes of our decision model as it is a constant that does not influence our decision variables. Also, we ignore the time to merge retrieved results as most merging algorithms rely on offline analysis to generate decision rules for merging that are relatively fast in real time. The few that require the broker to download documents from the IR servers and process them in real time are considered too time consuming and inefficient (Si and Callan 2003). Thus, the user's waiting cost is modeled as $\xi_j T$, where ξ_j denotes the user's disutility of waiting 1 second and T is the broker's wait time. The above function models a linear waiting cost. Other models with linear delay costs

---

[6] Although the latency can increase with the number of results displayed, the actual influence of brief text-based metadata on latency is lower than that of other factors such as network conditions and speed of user's and broker's connections.

include Mendelson and Whang (1990) and Montgomery et al. (2004). Nonlinear cost functions can be incorporated into our framework in future work.

Cost of Evaluating Information: The user's cognitive cost associated with comparing *P* results, each with *A* attributes, is modeled as $\lambda_j AP$ where λ_j is the user's cost of evaluating one result along one attribute. This function is based on the metric for the cost of thinking proposed by Shugan (1980) that has previously been applied to measure cost of evaluating online information (e.g., Montgomery et al 2004).[7] The metric is based on the number of elementary information processes (EIPs) involved in processing information. Different users incur the same number of EIPs. Heterogeneity in user cognitive costs is captured by heterogeneity in λ_j. Our use of a cognitive cost function that is linear in *P* is due to its common use in marketing and the tractability it affords our analytical model. In Section 5, we consider more complex information evaluation criteria and nonlinear cognitive cost functions.

Server Querying Fee: Lastly, the cost incurred in querying the servers is given by $\sum_{i=1..N} \eta_i q_i$, where η_i is the cost of querying server *i*. Note that this cost can be zero (η_i =0) for one or more servers.[8] Even though the organization rather than the specific user incurs the query fee, we incorporate the server querying fee in the surplus function to capture the IS manager's objective of maximizing the net surplus from the DIR system.

---

[7] Shugan (1980) proposed the metric $C(P) = \lambda(A-1)(P-1)$ for the cognitive cost associated with comparing *P* alternatives, each with *A* attributes. He does not explicitly consider the option of not evaluating the information. Accounting for the additional alternative of not evaluating, the cost of thinking is better modeled in our context as $C(P) = \lambda AP$

[8] For example, the Consolidated Tape Association (CTA), the administrative body that oversees the distribution of financial market data in the US, recommends that vendors offer both per query and fixed pricing plans. Financial data providers like Nasdaqtrader and Amexdata offer per query pricing as also plans with fixed fees. Usage-based pricing is commonly preferred by firms.

Given these different terms, the net surplus (S) given the query set ($\boldsymbol{q}$), wait time (T) and documents evaluated (P) is

$$S_j = \left(\sum_{k=1..P} U_{j,D-k+1:D} \right) - \sum_{i=1..N} \eta_i q_i - \xi_j T - \lambda_j AP \qquad (1)$$

The surplus function assumes piecewise separability of the individual components (utility from information, costs of querying, waiting and of evaluating information). Given a query, the broker makes operational decisions to maximize the expected surplus.

Table 1: Summary of Notation

N	Total number of servers
q_i	Variable that records if server i is queried (q_i=1) or not (q_i=0)
$\boldsymbol{q}$	Vector indicating which servers are queried ($\boldsymbol{q} = (q_1, q_2, ..., q_N)$)
Q	Total number of servers queried ($Q = \sum q_i$)
T	Broker's wait time
t_i	Response time of server i
$F_i(\cdot)$	cdf of the response time distribution of server i.
r_i	Variable that records if server i is retrieved (r_i=1) or not (r_i=0)
$\boldsymbol{r}$	Vector indicating which servers are retrieved ($\boldsymbol{r} = (r_1, r_2, ..., r_N)$)
d_i	Number of documents returned by server i assuming it is retrieved
D	Total number of documents retrieved ($D = \sum r_i \cdot d_i$)
$g_{ji}(\cdot)$	pdf of user j's utility from a document returned by server i
$U_{j,k:D}$	Utility from document ranked k among D documents sorted in increasing utility
$\boldsymbol{U_j}$	Vector recording utilities of retrieved documents ($\boldsymbol{U_j} = (U_{j,D:D}, U_{j,D-1:D}, \ldots, U_{j,1:D})$)
P	Number of results evaluated by the user
ξ_j	User j's disutility of waiting 1 second
λ_j	User j's cognitive cost of evaluating one result along one attribute
η_i	Cost of querying server i

3.2. Decision Problem

We now proceed to formulate the decision problem. We model it as a two-stage sequential process. In the first stage, the broker determines which servers to query (***q***) and how long to wait for responses (*T*). At the time these two decisions are made, the server response times (t_i) and document relevance scores are stochastic variables. The broker sorts the retrieved results in descending order of relevance scores. In the second stage, we determine the number of documents the user will evaluate which in turn influences the net surplus. At this decision time, the documents have already been retrieved and thus relevance scores of documents are known. We solve this sequential optimization problem in reverse order. That is, we first determine the number of documents evaluated by the user (*P*) given the set of retrieved results. Based on the user's response, we then determine ***q*** and *T*.

Stage 2: Determining Documents Evaluated by User

We now determine the number of documents (*P*) that will be evaluated by the user given the retrieved documents. It is important to determine *P* in order to compute the user's benefit from the information, which in turn influences the broker's choice of ***q*** and *T*. The user's decision problem in stage 2 (i.e. given ***r*** and ***U***$_j$) is

$$\max_P \left\{ \left(\sum_{k=1..P} U_{j,D-k+1:D} \right) - \lambda_j AP \,|\, \boldsymbol{r}, \boldsymbol{U_j} \right\} \qquad (2)$$

Note that the costs $\sum_{i=1..N} \eta_i q_i$ and $\xi_j T$ are already sunk at this stage and are not relevant to the user's decision *P*. The first term in equation (2) $\sum_{k=1..P} U_{j,D-k+1:D}$ is concave and monotonically increasing in *P*. The cost λAP is linear in *P*. Thus, equation 2 is concave in *P*. This leads to a very simple algorithm to determine the size of the evaluation set. We estimate *P* by first sorting the retrieved documents by relevance score. Starting with the document with the highest relevance score, we repeatedly add documents into the evaluation set as long as the documents offer utility greater than $\lambda_j A$.

The screening strategy above is also consistent with the *level cutoff* strategy studied by Feinberg and Huber (1996) in which only the alternatives that offer a minimal level of utility are evaluated. This screening strategy is an outcome of a linear cognitive cost function in our model. In Section 5, we study a compound stopping rule that models a nonlinear cost of evaluating the documents.

At the time of issuing a query, the broker does not know the utilities of documents that will be returned by the different servers and therefore does not know the exact documents that will be evaluated. The a priori estimate of the number of documents from server *i* that will eventually be evaluated by user *j* is $d_i(1-G_{ji}(\lambda_j A))$. Further if server i_1 stochastically dominates server i_2 ($G_{ji_1}(x) \leq G_{ji_2}(x), \forall x$) then the probability that a document from i_1 will be in the evaluation set given that i_1 has been retrieved is greater than the corresponding probability for i_2.

We now proceed to study how the broker can integrate the user's stage 2 decision into its operational decisions in stage 1 when the utility and response times are unknown.

Stage 1: Determining Servers to Query and Query Termination Time

There are clear tradeoffs in choosing the servers to query and the wait time. If the broker does not query a good server, the server is not retrieved and user surplus is unnecessarily reduced. Alternatively, if the broker queries irrelevant servers, access fees may be unnecessarily imposed. Similarly, the broker may decide to terminate a search but a highly relevant document may have been retrieved half a second later. Alternatively, the broker may choose to wait for a server's response, but may find that it ends up taking too long to respond or that the actual relevance of the documents is considerably lower than anticipated. We now formulate the problem of determining ***q*** and *T* to address these tradeoffs.

Given ***r*** and ***Uj***, the expected surplus of the user can be determined from the solution to equation 2. If $P_j(\boldsymbol{r},\boldsymbol{U_j})$ denotes that solution computed in stage 2, then the surplus given ***r*** and ***Uj*** is

$$S_j \mid \boldsymbol{r},\boldsymbol{U_j} = \left\{ \left(\sum_{k=1}^{P_j(\boldsymbol{r},\boldsymbol{U_j})} U_{j,D-k+1:D} \right) - \sum_{i=1..N} \eta_i q_i - \xi_j T - \lambda_j A P_j(\boldsymbol{r},\boldsymbol{U_j}) \right\} \qquad (3)$$

At the time the broker decides on the query set and wait time, neither ***r*** nor ***Uj*** are known due to the uncertainty in the response times, t_i and relevance scores $U_{j,k:D}$. The expected surplus can be computed by multiplying the probability of retrieving a vector ***r*** with the expected surplus from the associated evaluation set (evaluated over all possible ***Uj***) and then summing over all the 2^Q combinations of ***r***:

$$ES_j(\boldsymbol{q},T) = \left(\sum_{\boldsymbol{r}} \left(\prod_i F_i(T)^{r_i} (1 - F_i(T))^{1-r_i} \right) \cdot E_{\boldsymbol{U}|\boldsymbol{r}} \left[\sum_{k=1}^{P_j(\boldsymbol{r},\boldsymbol{U})} U_{j,D-k+1:D} - \lambda_j A P_j(\boldsymbol{r},\boldsymbol{U}) \right] \right) - \sum_{i=1..N} \eta_i q_i - \xi_j T$$

(4)

where $ES_j(\boldsymbol{q},T)$ denotes the expected surplus, $\prod_i F_i(T)^{r_i} (1 - F_i(T))^{1-r_i}$ is the probability of retrieving the random vector ***r*** given ***q*** and *T* and $E_{\boldsymbol{U_j}|\boldsymbol{r}}[\,]$ denotes an expectation over all possible values of ***Uj*** given the set of servers retrieved ***r***. Note that the cost of querying the servers and of waiting for responses are independent of the realized value of ***r*** and ***Uj*** and are hence not within the $\sum_{\boldsymbol{r}}$ expression. Thus, the optimization problem in stage 1 (when ***r*** and ***Uj*** are not known) is given by:

$$\max_{\boldsymbol{q},T} \left\{ ES_j(\boldsymbol{q},T) \right\} \qquad (5)$$

The optimization problem in (5) is a stochastic mixed integer program. In addition to the uncertainty with regard to the relevance scores and response times, the evaluation of (4) is complicated by the large number of combinations of ***r***. 30 servers imply 2^{30} (i.e., over a billion) combinations of ***r***. Thus it is important that any acceptable solution technique is able to solve (5) in a computationally fast manner.

By applying the assumption that the response times and document relevance scores are independent across servers, it is possible to simplify (4) and separate out the impact of each server. This yields the following decision problem (derivation is in online appendix A):

$$(\boldsymbol{q}^*, T^*) = \max_{\boldsymbol{q},T} \left\{ \left(\sum_{i=1..N} q_i F_i(T) \bar{U}_{ji} \right) - \sum_{i=1..N} \eta_i q_i - \xi_j T \right\} \tag{6}$$

In (6), q_i indicates whether server *i* is queried, $F_i(T)$ is the probability that the server is retrieved given that it has been queried and $\bar{U}_{ji} = d_i \left(\int_{\lambda_j A}^{\infty} (x - \lambda_j A) g_{ji}(x) dx \right)$ is the expected surplus from the d_i documents returned by server *i* given *i* is retrieved (recollect that only documents that offer utility greater than $\lambda_j A$ are evaluated). Because the servers are independent and the screening strategy in stage 2 does not involve inter-server interactions, equation (6) nicely separates out each server's net contribution to the overall surplus.

Computing the first order condition of (6) with respect to *T*, we get

$$\sum_{i=1..N} q_i f_i(T^*) \bar{U}_{ji} = \xi_j \tag{7}$$

At the same time, the expected benefit from querying server *i* is $F_i(T)\bar{U}_{ji}$ while the cost of querying it is η_i .Thus,

$$q_i^* = \begin{cases} 1 & \text{if } F_i(T)\bar{U}_{ji} \geq \eta_\text{i} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

The optimal query set and wait time can be determined by jointly solving (7) and (8). Note that the objective function need not be globally concave so it may not be straightforward to identify the optimal joint solution of (7) and (8). We investigate the concavity below and present an algorithm to determine the optimal decision variables.

Note that server *i* is queried if $F_i(T) \geq \eta_i / \bar{U}_{ji}$. Let $I_{\{F_i(T)\geq\eta_i/\bar{U}_{ji}\}}$ be an indicator variable that denotes whether *i* is queried. Then (6) may be rewritten as follows:

$$\max_T \left\{ \left(\sum_{i=1..N} I_{\{F_i(T)\geq\eta_i/\bar{U}_{ji}\}} F_i(T)\bar{U}_{ji} \right) - \sum_{i=1..N} \eta_i I_{\{F_i(T)\geq\eta_i/\bar{U}_{ji}\}} - \xi_j T \right\} \tag{9}$$

Consider the derivative of (9) with respect to *T*:

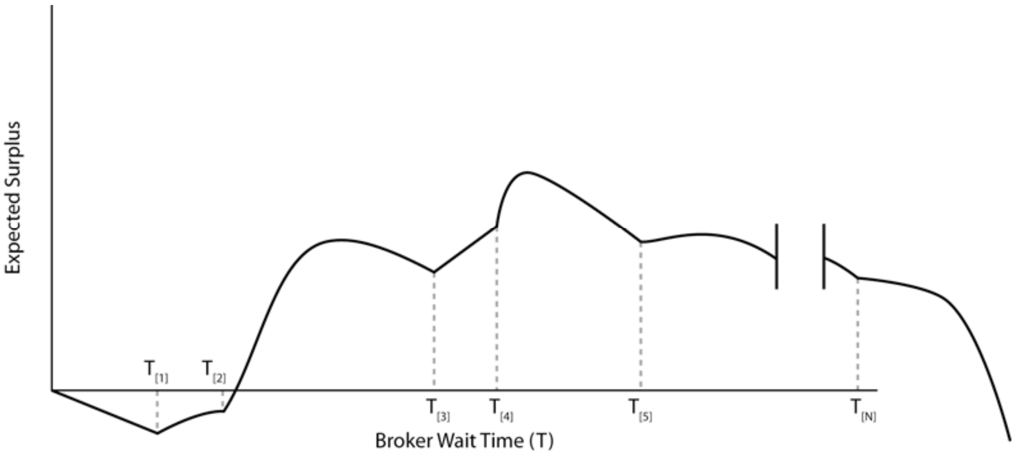
$$\left(\sum_{i=1..N} I_{\{F_i(T)\geq\eta_i/\bar{U}_{ji}\}} f_i(T)\bar{U}_{ji} \right) - \xi_j + \left(\sum_{i=1..N} \left[F_i(T)\bar{U}_{ji} - \eta_i \right] I'_{\{F_i(T)\geq\eta_i/\bar{U}_{ji}\}} \right) \tag{10}$$

Thus, a small increase in *T* can be associated with three effects. First, for the servers that are already in the query set ($I_{\{F_i(T)\geq\eta_i/\bar{U}_{ji}\}}$), it increases the probability that they will respond by $f_i(T)$ and thus there is a marginal benefit of of $f_i(T)\bar{U}_{ji}$ from each of these servers. Next, there is a cost $-\xi_j$ which is the user cost of waiting an additional time unit. Finally, a small increase in *T* can result in an additional server being added to the query set if there exists a server with $F_i(T)$ $= \eta_i / \bar{U}_{ji} - \Delta$. Otherwise, there is no change in the set of servers in the query set. That is, $I'_{\{F_i(T)\geq\eta_i/\bar{U}_{ji}\}}$ is generally zero except at $T = T_i = F_i^{-1}\left(\eta_i / \bar{U}_{ji}\right)$ when server *i* gets added into the query set. Note that even though a new server enters the query set at this *T* and $I'_{\{F_i(T)\geq\eta_i/\bar{U}_{ji}\}}$ is non-zero, that server makes no immediate contribution to the surplus because $F_i(T)$ $= \eta_i / \bar{U}_{ji}$

for that server and thus term 3 in (10) remains zero. However, the server is now in the query set and will help increase the marginal benefit of waiting (i.e., term 1 in (10)) for higher values of *T*.

We illustrate this effect in Figure 2. Initially when *T* is small, no server satisfies (8) and the query set is empty. Thus there is only a marginal cost of waiting ($-\xi_j$) but no marginal benefit. At some $T = T_{[1]}$, a server satisfies (8) and enters the query set resulting in a discontinuous change in the slope of the expected surplus ES_j. Specifically, the marginal benefit once the server has entered the query set is now given by the increase in the probability that the server responds multiplied by the expected surplus from the server (term 1 in equation (10)) and the marginal cost remains $-\xi_j$. If we continue to increase the broker wait time, then at some $T = T_{[2]} \geq T_{[1]}$, another server enters the query set. The marginal benefit from an increase in T now consists of the expected surplus times the change in the response probability for two servers. The marginal cost remains $-\xi_j$. As we increase *T*, this process repeats. Clearly, the derivative of the objective function with respect to *T* is not defined at the boundary points $\{T_{[1]}, T_{[2]}, ..., T_{[N]}\}$ and the objective function need not be locally concave either.

Figure 2: Expected Surplus against Broker Wait Time, T



Fortunately, it is possible to exploit some properties of the problem to formulate a computationally scalable algorithm to determine the optimal ***q*** and *T*. To do this, we first prove the following proposition in online appendix B.

Proposition 1: Suppose $T_i = F_i^{-1}\left(\eta_{\mathrm{i}} / \bar{U}_{ji}\right), \forall i \in \{1,2,...,N\}$. If server *l* is in the optimal query set, then all servers with $T_i \leq T_l$ are also in the optimal query set.

Corollary 1: The optimal query set is non-decreasing in the broker wait time *T*.

Based on Proposition 1, we first compute $T_i = F_i^{-1}\left(\eta_{\mathrm{i}} / \bar{U}_{ji}\right)$ for all servers and sort them in an ascending order. Let $T_{[l]}$ be the l^{th} lowest T_i for $l \in \{1,2,...,N\}$. For example, $T_{[1]} = \arg\min_{i=\{1,2,..N\}}\left\{F_i^{-1}\left(\eta_{\mathrm{i}} / \bar{U}_{ji}\right)\right\}$. For $T < T_{[1]}$, the query set is empty as no server satisfies (8). For $T \in [T_{[1]}, T_{[2]})$, the query set consists only of the server with the lowest T_i. For $T \in [T_{[2]}, T_{[3]})$, the query set consists of the two servers with $T_i < T_{[3]}$ and so on. This observation allows us to reduce the search space. For each value of *T*, we do not need to evaluate all 2^N query sets. Rather the specific query set associated with *T* is identified as described above.

Proposition 2: The maximum expected surplus *cannot* be realized at any of the boundary points $T_i = F_i^{-1}\left(\eta_{\mathrm{i}} / \bar{U}_{ji}\right)$.

The proof is in online appendix B. Based on Proposition 2, we now search for local maxima in each of these N regions. That is, in each of the regions $T \in [T_{[i]}, T_{[i+1]})$ wherein the expected surplus and its derivative are continuous in *T*, we identify local maxima that satisfy the following necessary and sufficient conditions,

$$\sum_{\{i|i\in\mathbb{N}, i\leq N, T_i\leq T\}} f_i(T^*)\bar{U}_{ji} = \xi_j \qquad \text{AND} \qquad \sum_{\{i|i\in\mathbb{N}, i\leq N, T_i\leq T\}} f_i'(T^*)\bar{U}_{ji} < 0 \tag{11}$$

The solution to (6) is given by computing the maximum among these local maxima. When the properties of $f_i()$ do not permit direct computation of the local maxima, one can use numerical techniques such as iterating through T with a small step size. An algorithm is provided in Figure 3. Notice that the technique requires the evaluation of N query sets rather than a search over all 2^N query sets. Thus, it scales rather well with the number of candidate servers.

Figure 3: Algorithm for determining the query set and wait time

```
Input $\eta_i, \bar{U}_{ji}, F_i$ for all $N$ servers and $\xi_j$ for user
Sort $N$ servers in ascending order of $T_i = F_i^{-1}\left(\eta_i / \bar{U}_{ji}\right)$. $T_{[l]}$ is $l^{th}$ lowest $T_i$. $T_{[N+1]} = T_{Max}$
Set Optimum_Surplus to 0 and q to (0,0,…,0)
Set Optimum_T to 0 and Optimum_q to (0,0,…,0)
Set $l$ to 1
While $T \le T_{[Max]}$ do
    Update q and set $q_i = 1$ for server with $l^{th}$ lowest $T_i$
    Set $T = T_{[l]}$
    While $T \le T_{[l+1]}$
        If $\left( \sum_{\{i|q_i=1\}} F_i(T)\bar{U}_{ji} \right) - \sum_{\{i|q_i=1\}} \eta_i - \xi_j T >$ Optimum_Surplus
            Set Optimum_Surplus to $\left( \sum_{\{i|q_i=1\}} F_i(T)\bar{U}_{ji} \right) - \sum_{\{i|q_i=1\}} \eta_i - \xi_j T$
            Set Optimum_T to $T$
            Set Optimum_q to q
        EndIf
        Set $T = T + \nabla T$
    EndWhile
    Set $l$ to $l + 1$
Endwhile
Output Optimum_q and Optimum_T
Halt
```

Proposition 3: If all $f_i(\cdot)$ are decreasing, then ES_j is locally concave in each of the regions $T \in [T_{[i]}, T_{[i+1]})$.

Response times of web servers often follow an exponential distribution which has a decreasing probability density function. Thus, it is possible to employ more efficient techniques for computing the local maxima in each of the regions $T \in [T_{[i]}, T_{[i+1]})$ given the local concavity.

3.3. Comparative Statics

Several additional properties of the optimal solution can be analytically derived. The most important property in order to derive the comparative statics is supermodularity.

Proposition 4: The broker's objective function is supermodular in its decisions ($\boldsymbol{q}$, T).

The proof is in online appendix B. Supermodularity implies complementarity between the decision variables. That is, having more of one variable increases the marginal returns to having more of the other. This is reflected in equations (7) and (8). Querying more servers increases the marginal return from waiting longer. Simultaneously, a longer wait time increases the returns from querying a server. Using the properties of supermodular functions, we can show the following results regarding the impact of exogenous variables on the optimal query set and wait time.

Proposition 5: The optimal query set and wait time are non-increasing in waiting cost, ξ.

Proposition 6: The optimal query set and wait time are non-increasing in user cognitive cost, λ.

Proposition 7: The optimal query set and wait time are non-increasing in access fee η_i, for all i.

All proofs are in online appendix B. An increase in ξ increases the marginal cost of waiting. This results in a decrease in the optimal wait time (setting the marginal benefit equal to the marginal cost as in Figure 4a). The decrease in the wait time reduces the expected benefit from querying each server. This can result in a decrease in the number of servers queried in accordance with (8). If fewer servers are queried, this in turn can result in a second order effect. A decrease in the number of servers queried reduces the marginal benefit of waiting as identified in (7). This can further reduce T as shown in Figure 4b and in turn affect the number of servers queried. Thus, both the first order and lower order effects act in the same direction such that the net effect is that the optimal query set and wait time are non-increasing in waiting cost ξ (Proposition 2). An increase in λ results in a decrease in the number of documents evaluated by a user. This reduces the marginal value of waiting as also that of querying the servers. Thus, the wait time and query set are non-increasing in λ (proposition 3). Similarly, an increase in any server's access fee can result in the elimination of that server from the broker's query set, which in turn would reduce the optimal wait time as highlighted above. Using properties of supermodular functions, comparative statics with respect to other parameters can be similarly derived.

Figure 4a: Direct Effect of an Increase in Marginal Cost of Waiting (ξ)

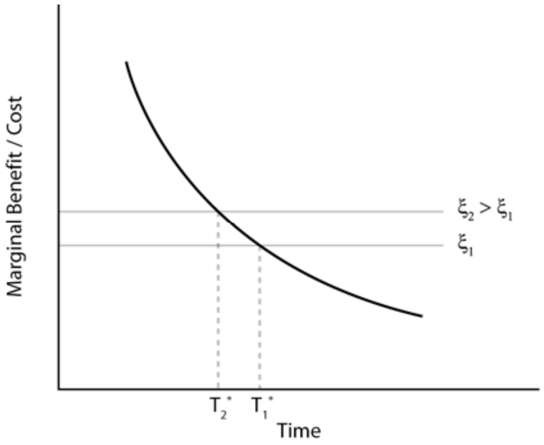
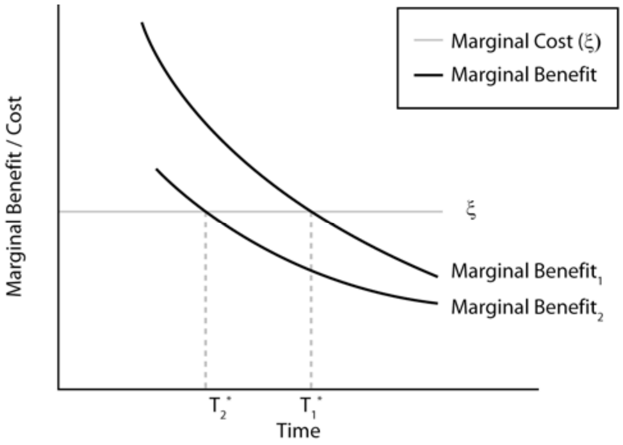


Figure 4b: Indirect Effect Due to a Decrease in the Marginal Benefit of Waiting



4. Empirical Illustration of Gain from Optimal Decision-Making

In this Section, we use simulations to measure gains from optimal decision-making. In order to instantiate the simulation parameters, we use data from FedStats which is a real-world DIR application. Our choice of this application context is due to its prior use in DIR research and by the availability of data.

4.1. Federated Search (FedStats)

Fedstats is a portal that provides unified access to information and statistics from over 100 federal agencies including NIH, USDA and census bureau. Fedstats was previously designed as a single-database architecture with information from all agencies replicated in a central database. Since 2003, Fedstats has adopted a distributed architecture wherein the broker forwards the query to IR servers of different agencies and merges the results. Avrahami et al (2006) provide a good review of the advantages of the DIR architecture.

In order to apply our techniques, we calibrated the relevance score and response time distributions using data gathered over 33 days in February/March 2006. Each day, our software agent queried the IR servers of 15 federal agencies and gathered server response times for 26 queries. The queries and IR servers are the same as in Avrahami et al (2006). The mean and standard deviation of the response time of the IR servers is in Table 2. The response time of the IR servers is modeled very well as a Gamma distribution, the parameters of which were estimated using Maximum Likelihood Estimation (MLE). We also extracted the top 20 documents that the servers returned in response to a query and computed the centralized relevance score of each document.[9] The centralized relevance score reflects the expected utility of a document. Since these are computed in a centralized manner, comparison of document relevance scores across IR servers is meaningful. In the following analysis, we only focus on relevance scores computed for the following queries: {crime rates, domestic violence, hate crime, homeless, suicide, unemployment rate} as they are broadly from the same topic area. The analysis for the entire query set is available upon request. Table 2 lists the mean and standard

---

[9] The scores were computed using the Lemur toolkit, an open source IR toolkit. The scores are not personalized based on any user characteristics but such personalized scores can be computed as in personalized IR systems.

deviation of the relevance scores of retrieved documents at the servers. The relevance scores for the top 6 servers are well described as Gamma distributions whereas the remaining servers have Normally distributed relevance scores. The parameters of these distributions were also estimated by MLE.

Table 2: Summary Statistics for Server Response Time and Document Relevance Scores

	Agency	**Response Time**		**Relevance Score**	
		Mean	**Std. Dev**	**Mean**	**Std Dev**
1	Bureau of Justice	0.41	0.81	0.2	0.12
2	Housing and Urban Development	1.8	4.00	0.17	0.07
3	ChildStats	1.7	4.50	0.2	0.04
4	Social Security Administration	0.27	1.09	0.1	0.07
5	National Science Foundation	1.14	0.33	0.06	0.06
6	Bureau of Economic Analysis	1.38	2.78	0.05	0.04
7	Economic Research Service	1.14	2.06	0.18	0.03
8	Bureau of Labor	0.39	1.39	0.15	0.02
9	National Institute of Drug Abuse	0.6	0.48	0.18	0.04
10	National Center for Education Stats	1.21	1.24	0.24	0.09
11	National Center for Health Stats	0.87	0.74	0.20	0.04
12	Environmental Protection Agency	0.4	1.77	0.17	0.04
13	Federal Reserve	1.48	0.95	0.14	0.03
14	National Inst. for Child Health & Development	0.61	0.38	0.16	0.03
15	Energy Information Administration	0.88	0.41	0.02	0.004

The heterogeneity among the servers is worth noting. Some servers such as the NSF IR server (server #5) respond fast and have low variance in the response time. The NSF server took greater than 5 seconds to respond in zero out of 858 searches. Unfortunately, the documents returned by NSF do not generally have high relevance scores for queries in our topic area. In contrast, some other IR servers such as those of Housing and Urban Development (HUD) and Bureau of Economic Analysis (BEA) take much longer to respond. The HUD server took more than 10 seconds to respond in 1.63% of the searches. However, HUD documents are generally very relevant. The BEA server took more than 10 seconds to respond in 3.2% of the searches. At the same time, BEA documents are not very relevant for queries in the chosen topic area. Hence, the broker may be better off not querying the BEA server.

4.2. Optimal Decisions

We now illustrate how to compute the optimal operational decisions using the above dataset. Even though server response time and relevance score distributions ($f_i(T)$, $g_{ji}(\cdot)$) are obtained from real-world data, servers' query fees and the user's waiting cost and cognitive cost need to be additionally specified. Our analytical model permits arbitrary values for these variables but for the purposes of the simulation we use some plausible values in our base case and conduct additional sensitivity analysis. In Section 6, we additionally discuss how these parameters can be estimated. For our base case, we assume that the cost of evaluating a document is two and a half times the cost of waiting a second ($\lambda = 2.5\xi$). This choice replicates the setting in Montgomery et al (2004). We also bootstrap the value of $\xi = 0.1$ so that the realized values of *P* in our simulations are typically between 5 and 25. Finally, we set the cost of querying the servers to 0.1 for all the servers in the base case (i.e., $\eta_i = 0.1$ for all *i*) which implies that the per-query fee charged by a server is of the same order of magnitude as the cost of waiting one second and the cost of evaluating one document.[10] Note that the querying fees are typically known a priori and the modeler can easily plug in appropriate values during implementation.

In Table 3, we compute the expected surplus $\bar{U}_{ji} = d_i\left(\int_{\lambda_j A}^{\infty}(x-\lambda_j A)g_{ji}(x)dx\right)$ from each of the servers if it is retrieved. In addition, we compute the waiting time ($T_i = F_i^{-1}\left(\eta_i / \bar{U}_{ji}\right)$) at

---

[10] It is also possible to express these values in dollar terms. For example, an annual wage of $70,000 translates to a value of time of approximately $0.01/second. Because we assume that $\xi = 0.1$, this implies that our unit above is approximately a tenth of a dollar. Correspondingly, our assumptions imply that the cost of evaluating a document is $0.025 and the cost of querying a server is $0.01 per query (vendors of Nasdaq data charge close to $0.005 per query so these are close to reality). Finally, the unit also suggests the value of documents in dollar terms. For example, the average value of a document returned by the Bureau of Justice in response to a query in our topic area is 0.2 units or approximately $0.02.

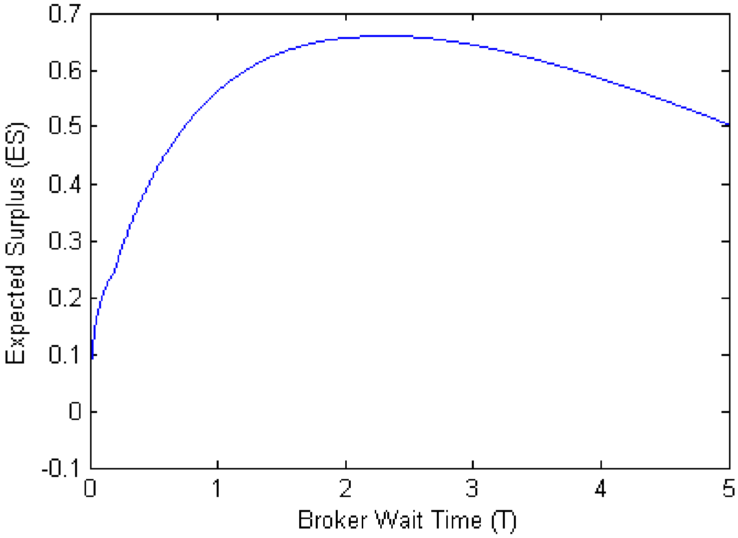
which it is optimal for the broker to add the server into the query set. Interestingly only three servers, namely those of the Bureau of Justice, Housing and Urban Development, and National Center for Educational Stats, have a finite T_i for the topic area and above parameters. For all other servers the expected surplus is not sufficient to offset the querying fee even if the broker wait time is high enough. If the query fees are reduced to $\eta_i = 0.025$, then servers 3, 4, and 11 may also be worth querying if the broker wait time is reasonably high (i.e. $\bar{U}_{ji} > 0.025$ for these servers). The table can be used to quickly identify the which servers to query for any arbitrary set of querying costs $\{\eta_1, \eta_2, ..., \eta_N\}$. In addition, the technique not only helps in identifying the optimal operational decisions but can also shed light on the vendor pricing plans that are acceptable to an IS manager.

Table 3: Expected Surplus and Minimum Wait Time to Consider Querying a Server

	Agency	Expected Surplus ($\bar{U}_{ji}$)	Minimum Wait Time Needed to Query ($T_i = F_i^{-1}\left(\eta_i / \bar{U}_{ji}\right)$)
1	Bureau of Justice	0.583	0.001
2	Housing and Urban Development	0.128	2.076
3	ChildStats	0.051	∞
4	Social Security Administration	0.045	∞
5	National Science Foundation	0.019	∞
6	Bureau of Economic Analysis	0.001	∞
7	Economic Research Service	0.002	∞
8	Bureau of Labor	0.000	∞
9	National Institute of Drug Abuse	0.013	∞
10	National Center for Education Stats	0.622	0.198
11	National Center for Health Stats	0.040	∞
12	Environmental Protection Agency	0.007	∞
13	Federal Reserve	0.000	∞
14	National Inst. for Child Health & Development	0.000	∞
15	Energy Information Administration	0.000	∞

Figure 5 plots the expected surplus against the wait time. For $T < 0.001$, the query set is empty and the expected surplus is negative. For $T \in [0.001, 0.198)$, the query set consists of only server 1. At T = 0.198, server 10 also enters the query set and we observe a sudden increase in the slope of the expected surplus function. Similarly, at T = 2.076, server 2 also enters the query set. The query set does not change subsequently. The expected surplus is maximized at T^* = 2.318 and the corresponding optimal query set consists of servers 1, 2 and 10. These operational decisions can be easily computed given data on past performance of the servers and the user parameters.

Figure 5: Expected Surplus against Broker Wait Time (assuming optimal query set $q^*(T)$)



We now conduct some sensitivity analysis to determine the impact of exogenous parameters on the expected surplus at the optimal operational decisions. Figure 6 presents the impact of the waiting cost and the query fee. First we compute the expected surplus obtained from the operational algorithm derived in Section 3 (this is labeled "Algorithm"). Simultaneously, we also compute the expected surplus obtained from a simple but reasonable heuristic in which we query all servers and wait for 5 seconds for the servers to respond (labeled "heuristic"). Clearly, an increase in the waiting cost or the query fee decreases the expected surplus even if the operational decisions are optimally adjusted. At the same time, we observe that the decay in the expected surplus under optimal operational decisions is not as drastic as that observed with the naïve heuristic. The algorithm is successful in responding to an increase the user waiting cost or server fees and ensures that the expected surplus continues to remain positive. In contrast, the naïve heuristic quickly deteriorates in performance when there are non-trivial costs.

Figure 7 presents a similar plot of the expected surplus against the cognitive cost of evaluating information and the user waiting cost. It is evident that the cognitive cost can have a significant impact on the expected surplus. This is because the parameter λ_j impacts the value realized from each and every document that is retrieved as opposed to the waiting cost (incurred once for the entire query) and the query fees (once per server). Yet again, the expected surplus under optimal operational decisions can be significantly higher than that under the naïve heuristic when the costs are non-trivial.

Figure 6: Impact of Waiting and Querying Costs on Expected Surplus

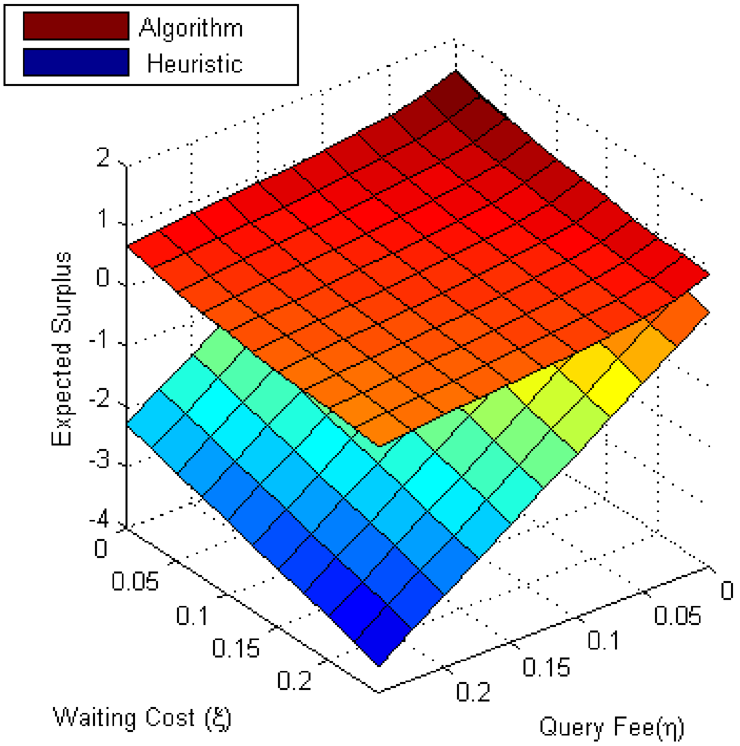
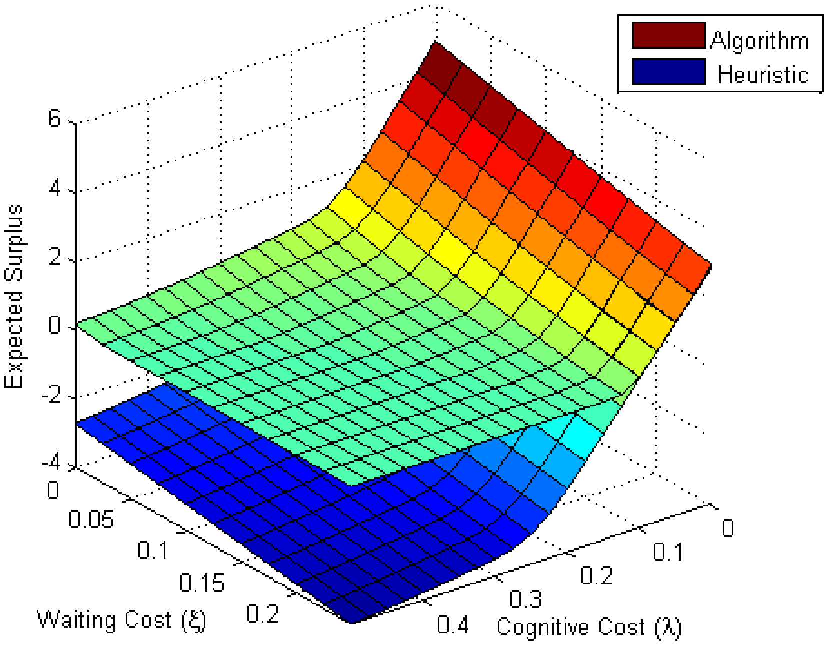


Figure 7: Impact of Waiting and Cognitive Costs on Expected Surplus under Optimal Operational Decisions and Naïve Heuristic



5. A Simulation Based Technique under Convex Cognitive Cost

In our model in Section 3, the expected surplus $\bar{U}_{ji}$ from server *i* was independent of the other servers queried. However, complex models of user preferences can generate inter-server dependencies even when server response time and relevance score distributions are independent. For example, consider the case in which user cognitive cost is convex in the number of documents evaluated (*P*). Under convex cognitive costs, the marginal cost of evaluating a document from server *i* depends on the rank of that document in the display set which in turn depends on the quality of documents returned by the other servers. As a result, the expected surplus from querying *i* ($\bar{U}_{ji}$) does not have a fixed value but is a function of the query set itself. The techniques of Section 3 are not directly applicable when a fixed $\bar{U}_{ji}$ cannot be computed for each of the candidate servers.

We develop a simulation based technique to determine the broker's optimal query set and wait time. April et al. (2001) and Glover et al. (1999) provide a useful primer on the merits of combining simulation and optimization in managing the complexity and uncertainty posed by many real-world problems. Our simulation-based technique builds on the results from Section 3 but additionally incorporates the notion that the expected surplus from a server ($\bar{U}_{ji}$) is a function of the query set.

To illustrate the use of simulations, we consider a compound stopping rule that generates inter-server dependencies. It has been suggested that even if there is an unlimited supply of relevant documents, users are unlikely process all of them. For example, Kraft and Buell (1984) suggest a fatigue stopping rule that assumes there is an upper bound (P_{MAX}) on P. Feinberg and Huber (1996) call this the quota cutoff criteria. We model this by assuming $C(P)=\begin{cases}\lambda AP & if\ P\le P_{Max}\\ \infty & if\ P>P_{Max}\end{cases}$. This compound stopping rule can be treated as an extreme case of the convexity in cognitive costs described above.

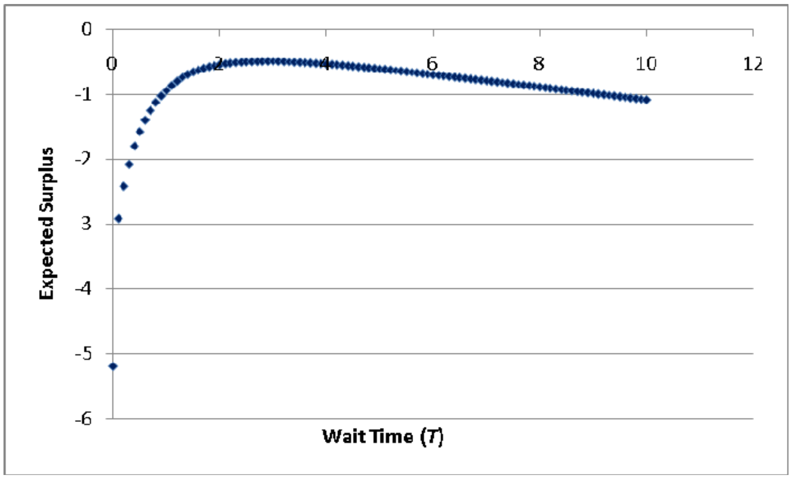
5.1 Determining Optimal *q* and *T*

First, we discuss determination of the optimal wait time given the query set ***q***. We then describe how to determine ***q***. Simulation parameters are based on the Fedstats dataset.

Optimal Query Termination Given Query Set

Given a query set ***q***, the expected surplus associated with any given choice of broker wait time can be determined using Monte Carlo simulations. In each run of the simulation, we draw the document relevance scores and server response times from distributions specified in Table 2. Next, we evaluate the expected surplus for a range of values of T selected from a grid (e.g., $T=\{0.1, 0.2, \ldots, 10\}$). Given the simulated relevance scores and response times and choice of T, we identify the servers retrieved and compute the surplus from the evaluated documents. Finally, the expected surplus associated with each T is obtained by averaging the surplus realized in 10,000 runs of the simulation. Figure 8 plots the expected surplus against the broker's wait time for the Fedstats data assuming all servers are queried (q_i=1, for all i), $\eta_i = 0.1$ for all i, $\xi = 0.1$, P_{MAX} = 15 and $\lambda = 0.25$. The expected surplus is maximized when T^* = 3.0 seconds.

Figure 8: Expected Surplus versus T (Q=15)



Determining the Query Set

The analysis in Section 3 indicated that the critical score to determine the query set under independence in the servers' expected contribution is $T_i = F_i^{-1}\left(\eta_i / \bar{U}_{ji}\right)$. Our simulation heuristic extends that insight while incorporating the fact that the expected surplus from a server evolves with the query set. The heuristic works as follows. Initially, all servers are queried in the first stage of the simulations (Q=N). The optimal wait time is computed as described above. Next, we compute the average contribution of each server to the user surplus. The contribution of a server in an individual simulation run is obtained by summing the utility from all those documents from the server that are evaluated by the user and subtracting the marginal cost of evaluating each document. The average contribution is simply the average over all simulation runs. We denote this contribution by $\bar{U}_{ji}(\boldsymbol{q})$. Unlike Section 3 the average contribution of a server depends on the query set $\boldsymbol{q}$. Next we compute $T_i = F_i^{-1}\left(\eta_i / \bar{U}_{ji}(\boldsymbol{q})\right)$ for all servers. We identify the server with the highest $F_i^{-1}\left(\eta_i / \bar{U}_{ji}(\boldsymbol{q})\right)$ and set the corresponding q_i=0. Next, with the new set of (N-1) servers, we again compute the optimal wait time and the average contributions of each of the servers. We again identify the server with the highest $F_i^{-1}\left(\eta_i / \bar{U}_{ji}(\boldsymbol{q})\right)$ and eliminate that server. We proceed in this manner until we are left with just one server. In this manner, we evaluate N possible choices for q. Finally, we select the option that yields the highest expected surplus among these N options.

In Table 4, we demonstrate this process for the 15 servers identified in Table 2. All parameter values are the same as the ones used to generate Figure 8. We begin by querying all 15 servers. Given this query set, the optimal wait time is 3.0s and the expected surplus is -0.49 units. Server 8 has the highest T_i and is eliminated.[11] In the next stage, we query the 14 remaining servers (second row of Table 4). The optimal wait time is 3.1s, associated expected surplus is -0.40 and the server with the highest T_i is server 15. Server 15 is now eliminated and we are left with 13 servers. This process repeats until we have evaluated all 15 combinations. In the last stage, server 1 is the only server that is queried. The optimal wait time is 2s and expected surplus is 0.29 units. Among the 15 combinations, the algorithm recommends querying 2 servers, namely servers 1 and 10 (i.e., IR servers of Bureau of justice and National center of educational statistics). The corresponding optimal wait time is 4.0s and the expected surplus under these decisions is 0.57 units. Note that the recommended servers are those that contain the most

---

[11] In case of ties, we eliminate the server with the lowest $\bar{U}_{ji}(\boldsymbol{q})/\eta_i$. Any additional ties are broken randomly. All values in Table 4 are rounded to two decimal places. Ties in $\bar{U}_{ji}(\boldsymbol{q})/\eta_i$ were rarely observed.

relevant documents and also highly likely to respond within the broker's waiting period. Unlike the results in Section 4, the optimal query set no longer includes server 2. This is because very few of server 2's documents appear among the top 15 documents as long as servers 1 and 10 are in the query set and therefore do not enter the evaluation set. This in turn reduces the contribution ($\bar{U}_{ji}(\boldsymbol{q})$) of server 2 and therefore increases its T_i. The net result is that it is no longer optimal to query server 2.

Table 4: Determining the Optimal Query Set (optimal solution shaded gray)

# of Servers	Optimal Wait Time	Expected Surplus		Sever														
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
15	3	-0.49	$U_{ji}(\boldsymbol{q})$	0.56	0.09	0.03	0.04	0.02	0.00	0.00	0.00	0.01	0.53	0.02	0.00	0.00	0.00	0.00
			T_i	0.00	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	0.24	Inf	Inf	Inf	Inf	Inf
14	3.1	-0.40	$U_{ji}(\boldsymbol{q})$	0.55	0.10	0.03	0.04	0.02	0.00	0.00	-	0.01	0.54	0.02	0.00	0.00	0.00	0.00
			T_i	0.00	Inf	Inf	Inf	Inf	Inf	Inf	-	Inf	0.23	Inf	Inf	Inf	Inf	Inf
13	3.1	-0.30	$U_{ji}(\boldsymbol{q})$	0.56	0.09	0.03	0.04	0.02	0.00	0.00	-	0.01	0.54	0.02	0.00	0.00	0.00	-
			T_i	0.00	Inf	Inf	Inf	Inf	Inf	Inf	-	Inf	0.23	Inf	Inf	Inf	Inf	-
12	2.9	-0.19	$U_{ji}(\boldsymbol{q})$	0.55	0.10	0.03	0.04	0.02	0.00	0.00	-	0.01	0.53	0.02	0.00	-	0.00	-
			T_i	0.00	Inf	Inf	Inf	Inf	Inf	Inf	-	Inf	0.24	Inf	Inf	-	Inf	-
11	3	-0.09	$U_{ji}(\boldsymbol{q})$	0.56	0.09	0.03	0.04	0.02	0.00	0.00	-	0.01	0.54	0.02	0.00	-	-	-
			T_i	0.00	Inf	Inf	Inf	Inf	Inf	Inf	-	Inf	0.23	Inf	Inf	-	-	-
10	2.9	0.02	$U_{ji}(\boldsymbol{q})$	0.56	0.09	0.03	0.04	0.02	0.00	-	-	0.01	0.54	0.02	0.00	-	-	-
			T_i	0.00	Inf	Inf	Inf	Inf	Inf	-	-	Inf	0.23	Inf	Inf	-	-	-
9	2.7	0.12	$U_{ji}(\boldsymbol{q})$	0.55	0.09	0.03	0.04	0.02	-	-	-	0.01	0.52	0.02	0.00	-	-	-
			T_i	0.00	Inf	Inf	Inf	Inf	-	-	-	Inf	0.24	Inf	Inf	-	-	-
8	2.9	0.21	$U_{ji}(\boldsymbol{q})$	0.55	0.09	0.03	0.04	0.02	-	-	-	0.01	0.54	0.02	-	-	-	-
			T_i	0.00	Inf	Inf	Inf	Inf	-	-	-	Inf	0.23	Inf	-	-	-	-
7	3.3	0.28	$U_{ji}(\boldsymbol{q})$	0.55	0.10	0.03	0.04	0.02	-	-	-	-	0.55	0.03	-	-	-	-
			T_i	0.00	Inf	Inf	Inf	Inf	-	-	-	-	0.23	Inf	-	-	-	-
6	3.1	0.38	$U_{ji}(\boldsymbol{q})$	0.55	0.10	0.03	0.04	-	-	-	-	-	0.54	0.03	-	-	-	-
			T_i	0.00	Inf	Inf	Inf	-	-	-	-	-	0.23	Inf	-	-	-	-
5	3.6	0.44	$U_{ji}(\boldsymbol{q})$	0.56	0.10	0.03	0.04	-	-	-	-	-	0.56	-	-	-	-	-
			T_i	0.00	Inf	Inf	Inf	-	-	-	-	-	0.22	-	-	-	-	-
4	3.9	0.49	$U_{ji}(\boldsymbol{q})$	0.57	0.10	-	0.04	-	-	-	-	-	0.58	-	-	-	-	-
			T_i	0.00	23.25	-	Inf	-	-	-	-	-	0.22	-	-	-	-	-
3	3.8	0.56	$U_{ji}(\boldsymbol{q})$	0.56	0.10	-	-	-	-	-	-	-	0.57	-	-	-	-	-
			T_i	0.00	15.47	-	-	-	-	-	-	-	0.22	-	-	-	-	-
2	4	0.57	$U_{ji}(\boldsymbol{q})$	0.58	-	-	-	-	-	-	-	-	0.59	-	-	-	-	-
			T_i	0.00	-	-	-	-	-	-	-	-	0.21	-	-	-	-	-
1	1.7	0.29	$U_{ji}(\boldsymbol{q})$	0.54	-	-	-	-	-	-	-	-	-	-	-	-	-	-
			T_i	0.00	-	-	-	-	-	-	-	-	-	-	-	-	-	-

The above algorithm requires the evaluation of N different query sets rather than 2^N combinations. The scheme is clearly very efficient in terms of reducing the search space. However, this scheme need not be optimal because the expected surplus from a server can change with the query set. The algorithm does not re-evaluate servers that have already been eliminated in previous stages even though their contribution can be different in a new query set.

Thus, it is possible that a reasonably good server is eliminated unnecessarily. In the next section, we evaluate the performance of the heuristic.

5.2 Evaluation

In order to assess whether the above heuristic to restrict the search space is reasonable, we compare the surplus from the algorithm with the surplus that is realized from an exhaustive search through all possible combinations of q. Clearly, it is not always feasible to evaluate all 2^N combinations of q. For example, Fedstats queries over 100 servers resulting in over 2^{100} combinations. Even with 30 servers, there are over a billion combinations of q. So, we only seek to verify that the algorithm performs well relative to the option of evaluating all 2^N combinations for relatively small values of N (specifically N=15). Using simulations, we compute a) the expected surplus under the query set suggested by our algorithm and b) the expected surplus associated with all possible values of q. We find that the optimal operational decisions (and consequently expected surplus) under our proposed algorithm is the same as the one identified through an evaluation of all possible q.[12]

We ran 30 additional evaluation experiments. In each experiment, we choose 8 servers randomly from the initial set of 15 servers. With these 8 servers, we determine the optimal query set and wait time using our proposed algorithm and by exhaustive evaluation of all 2^8 possible query sets. The results are in Table 5. The proposed algorithm recommended the same query set as the one obtained from evaluating all 2^8 combinations in 28 out of the 30 simulations. In the two experiments where the optimal decisions were different, there was no notable difference in the expected surplus. The expected surplus from the proposed algorithm averaged over all 30

---

[12] While it takes less than an hour to identify the optimal decisions under our proposed algorithm, exhaustive search required nearly 13 days on a machine with two 3.06 GHz processors and a 2 GB RAM. Also note that the search space under the latter strategy grows exponentially with the number of servers and will rarely be feasible with more servers.

experiments is 0.28, which is the same as under exhaustive evaluation. Thus, even though the algorithm does not re-evaluate servers once they are eliminated, its approach of identifying servers to eliminate is effective. Simultaneously, it helps significantly reduce the computational complexity of the evaluation by reducing the size of the search space. Thus, the algorithm has several desirable properties in terms of performance and ability to scale as the number of servers (*N*) increases. We conducted additional sensitivity analysis by varying the parameters ξ, λ, η_i and also considered convex cognitive cost functions of the form $C(P) = \lambda P^k$ where $k > 1$. The heuristic continued to perform well in these additional tests as well.

Table 5: Comparison with exhaustive evaluation

# Simulations with Matching Decisions	Exhaustive Search		Proposed Algorithm	
	Range of Exp Surplus	Average Exp Surplus	Range of Exp Surplus	Average Exp Surplus
28	0.00-0.57	0.28	0.00-0.57	0.28

6. Discussion and Conclusions

In this paper, we formulated the decision problem for a broker in distributed IR, analytically derived a solution that can be implemented in a computationally efficient manner and extended the approach to more complex decision environments. We demonstrated that the net surplus can be significantly enhanced by using the approach. Improved user modeling can help IS managers in designing and deploying DIR systems that generate greater user satisfaction. Various corporations spend large amounts in acquiring and providing centralized access to large distributed data repositories in order to empower their information workers. The design of intelligent information systems such as intelligent DIR systems will contribute to increased adoption of systems by their users and will help generate better return on investment from enterprise IS.

We now discuss implementation challenges with the proposed approach and conclude by discussing future directions for research. Our model assumes that it is possible to estimate the user utility (i.e., estimate U_{jk}, ξ_j, λ_j). This raises two important questions tied to implementation. The first relates to techniques that can be employed to estimate these parameters and the granularity at which these parameters can be estimated. A second question relates to the impact of uncertainty in the estimated parameters.

With regard to techniques for estimating user preferences, there are three approaches that can be used in implementing the model. In the IR community, a recent focus has been the design of personalized IR systems that personalize search results using models of user interests based on previously issued queries and previously visited webpages (Teevan et al. 2005). These techniques allow the computation of user-specific relevance scores. An additional approach available is the use of econometric models that estimate utility weights using prior choice/clicks data. Smith et al. (2001) estimate aggregate utility weights for users at a shopbot. Rossi et al. (1996) propose an individual-level multinomial probit model to estimate utility weights for each user. Estimating individual-level parameters requires a lot more data on past user activity. When such data are not available, segment-level estimation may be more appropriate. In an enterprise setting, a segment may be users within a division. In cases where segments are not easily specified in advance, latent segments can be identified and estimated (see Kamakura and Russell (1989) and Andrews et al. (2002)). Finally, another highly appealing option available is the use of conjoint analysis. In conjoint analysis, respondents are presented with options that simultaneously vary two or more attributes and are asked to indicate their preferences among these options (e.g., one option may entail waiting for an additional second and another may entail evaluating an additional document). Respondents' preference orderings are then used to estimate the utility part-worths. The technique has been widely adopted by marketing researchers and practitioners (see Greene et al. (2001) for a detailed survey). A conjoint task can be designed for users of a DIR system within an enterprise setting to estimate how users trade off the benefit from a document with cognitive and waiting costs.

Another important issue is that of uncertainty in the estimates. That is, what is the impact of errors in U_{jk}, ξ_j or λ_j. If the errors are iid with zero mean, then the optimal decisions need not change as long as the expected surplus function in Section 3 is additive and piecewise separable. However, it may be possible to exploit the error structure under some circumstances. Furthermore, it would be most useful to conduct sensitivity analysis to measure the impact of small changes in parameters on the optimal decisions. This can help determine whether to strictly implement the decisions generated by the model or to use the solution as an indicator of the neighborhood in which the optimal solution may lie.

There are several interesting avenues for future research. In this paper, we considered an IR broker that is interested in maximizing net surplus without any other constraints. There may be other objective functions worth exploring and resource constraints worth modeling. Another interesting extension will be the study of the algorithms in environments where there is considerable overlap in results across servers. Our analysis focuses on federated search where the overlap is minimal. Simulations that incorporate the possibility of overlap suggest that the techniques described in Section 5 are promising even in the presence of some overlap across servers. However, it may be feasible to exploit any knowledge of overlap patterns and develop more efficient algorithms for DIR environments with significant overlap. Finally, we considered a static wait time for the broker. The approach does not account for information a broker may gather in real time during a specific retrieval. For example, during a particular search, a broker may have retrieved the top few documents early on but will end up waiting until the recommended waiting period has elapsed or all servers have responded. In this situation, the broker may be better off terminating the search early since it knows that the servers expected to be most relevant have already responded. Hosanagar (2005) presents an adaptive approach to allow the broker to adjust the decisions in real time. Other adaptive techniques and metaheuristics to evaluate these alternatives can also prove useful.

References

R. L. Andrews, A. Ainslie, and I. S. Currim, "An Empirical Comparison of Logit Choice Models with Discrete Versus Continuous Representations of Heterogeneity," *Journal of Marketing Research*, 39, 479–87, 2002.

April, J., F. Glover, J. Kelly and M. Laguna, "Simulation/Optimization Using "Real-World" Applications," Proceedings of the Winter Simulation Conference, 2001.

J. A. Aslam, M. Montague. Models for Metasearch. In Proc. of the 23rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval, 2001.

T. T. Avrahami, L. Yao, L. Si and J. Callan, "The FedLemur project: Federated search in the real world", *Journal of the American Society for Information Science and Technology*, 57(3), 2006.

Baeza-Yates, R., Ribeiro-Neto, B. (eds.), Modern Information Retreival, ACM Press, New York, 1999.

J. P. Callan, Z. Lu, and W. B. Croft, "Searching distributed collections with inference networks," In Proceedings of the 18th Annual SIGIR Conference, Seattle, WA, 1995.

Chase, W.G. Elementary information processes. In W.K. Estes (Ed.), Handbook of Learning and Cognitive Processes. Volume 5. Human Information Processing. Hillsdale, NJ: Erlbaum, 1978.

B. Dellaert and B. Kahn. How Tolerable is Delay? Consumers' Evaluations of Internet Web Sites after Waiting, *Journal of Interactive Marketing,* 13(1), 1999, 41-54.

Dey, D., Sarkar, S., and De, P., "A Probabilistic Decision Model for Entity Matching in Heterogeneous Databases," *Management Science*, Vol. 44, No. 10, pp. 1379–1395 (1998).

Dey, D., "Record Matching in Data Warehouses: A Decision Model for Data Consolidation," *Operations Research*, Vol. 51, No. 2, pp. 240 – 254 (2003).

O. Etzioni, S. Hanks, T. Jiang, R. Karp, O Madani, and O. Waarts, “Efficient information gathering on the internet,” In Foundations of Computer Science (FOCS), pages 234-243, 1996.

F. M. Feinberg, and J. Huber, “A Theory of Cutoff Formation under Imperfect Information,” *Management Science*, 42(1), 65-84, 1996.

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

F. Glover, J. Kelly and M. Laguna. “New Advances for Wedding Optimization and Simulation," Proceedings of the 1999 Winter Simulation Conference, 1999.

L. Gravano and H. Garcia-Molina. Generalizing GloSS to Vector-Space Databases and Broker Hierarchies. In Proceedings of the 21st International Conference on Very Large Databases (VLDB), 1995.

Johnson and Payne, Effort and accuracy in choice. *Management Science*. 31 (4). 395-414, 1985

K. Hosanagar. A Utility Theoretic Approach to Determining Optimal Wait Times in Distributed Information Retrieval. In Proceedings of the ACM SIGIR conference, 2005.

M. Hui and D. K. Tse, “What to Tell Consumers in Waits of Different Lengths: An Integrative Model of Service Evaluation,” *Journal of Marketing*, 60, 81-90, 1996.

Ivory, M. Y. and Hearst, M. A., “Improving Web Site Design,” *IEEE Internet Computing* 6, 2, March 2002.

W. A. Kamakura and G. J. Russell, “A Probabilistic Choice Model for Market Segmentation and Elasticity Structure,” *Journal of Marketing Research*, 26, 379–90, 1989.

D. H. Kraft, D. A. Buell, Advances in a Bayesian Decision Model of User Stopping Behavior for Scanning the Output of an Information Retrieval System. *SIGIR 1984*: 421-433.

R. Krishnan, X. Li, D. Steier, and L. Zhao, On Heterogeneous Database Retrieval: A Cognitively Guided Approach, *Information Systems Research* 12, No. 3 (2001) 286-301.

D. Larcker and V. Lessig, “Perceived Usefulness of Information: A Psychometric Examination,” *Decision Sciences*, 11, 1, 1980.

A. Le Calve, J. Savoy. Database Merging Strategy Based on Logistic Regression. Information Processing & Management, 36(3), 2000.

R. Moenaert and W. Souder, Context and antecedents of information utility at the R&D/marketing interface, *Management Science*, v.42 n.11, p.1592-1610, Nov. 1996.

A. Montgomery, K. Hosanagar, R. Krishnan and K. Clay (2004), “Designing a Better Shopbot,” *Management Science*, Vol 50, No. 2.

S. E. Robertson. The probability ranking principle in IR. *Journal of Documentation*, 33:294--304, 1977.

S. Shugan, “The Cost of Thinking”, *Journal of Consumer Research*, Vol. 7, September, 1980.

M. D. Smith and E. Brynjolfsson. “Customer Decision Making at an Internet Shopbot: Brand Still Matters,” *The Journal of Industrial Economics*, 49(4) 541-558, 2001.

L. Si and J. Callan, “Using Sampled Data and Regression to Merge Search Engine Results”, In Proceedings of ACM SIGIR conference, Finland, 2002.

L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In Proc. ACM SIGIR 2003, July-August 2003.

L. Si and J. Callan, "Unified Utility Maximization Framework for Resource Selection" In Proc. of the 13th International Conference on Information and Knowledge Management, Washington D.C, 2004.

J. Teevan, Dumais, S. T., and Horvitz, E. Personalizing search via automated analysis of interests and activities. In Proc. of the ACM SIGIR conference, Salvador, Brazil, August 2005.

D. Topkis, *Supermodularity and Complementarity*, Princeton University Press, 1998.

E. Voorhees, N. K. Gupta, and B. Johnson-Laird, “Learning Collection Fusion Strategies,” In Proc. of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1995.