**User-Centric Operational Decision-Making in Distributed Information Retrieval**

**Kartik Hosanagar**

**Online Appendix**

## I. Impact of Correlation across Servers

To illustrate the impact of correlations across servers, we consider simple low-dimensional examples with two or three servers. First, we consider a case in which the response times are correlated across servers but relevance scores are independent. Next, we consider a case in which server relevance scores are correlated but response times are independent. Throughout this section, we suppress subscript j for ease of notation but it is implied that all analysis is for an individual user or user segment.

Correlated Response Times and Independent Relevance Scores.

Case 1: Two Servers

If the servers are independent, then the expected surplus can be expressed as follows:

$$ES = q_1 F_1(T)\overline{U}_1 + q_2 F_2(T)\overline{U}_2 - \eta_1 q_1 - \eta_2 q_2 - \xi T$$

Let $\Pr(i1 \in \{0,1\}, i2 \in \{0,1\})$ denote the probability of observing a specific retrieval set if both servers are queried. That is Pr(0,1) indicates the probability that server 1 is not retrieved and server 2 is retrieved. Pr(0,0), Pr(1,0) and Pr(1,1) are defined in a similar fashion. Suppose the server response times are NOT independent. Then $\Pr(0,1) \neq (1 - F_1(T)) \cdot F_2(T)$. In other words, the probabilities can take arbitrary values as long as:

$\Pr(0,1) + \Pr(0,0) = \Pr(0,X) = (1\text{-}F_1(T))$

$\Pr(1,1) + \Pr(1,0) = \Pr(1,X) = F_1(T)$

$\Pr(1,0) + \Pr(0,0) = \Pr(X,0) = (1\text{-}F_2(T))$

$\Pr(1,1) + \Pr(0,1) = \Pr(X,1) = F_2(T)$

Then the expected surplus in this case of independent servers is:

$$ES = q_1(1-q_2)F_1(T)\overline{U}_1 + q_2(1-q_1)F_2(T)\overline{U}_2 + q_1 q_2\left(\Pr(0,0)\cdot 0 + \Pr(1,0)\overline{U}_1 + \Pr(0,1)\overline{U}_2 + \Pr(1,1)(\overline{U}_1 + \overline{U}_2)\right) - \eta_1 q_1 - \eta_2 q_2 - \xi T$$

Simplifying,

$$ES = q_1F_1(T)\overline{U}_1 + q_2F_2(T)\overline{U}_2 - q_1q_2(F_1(T)\overline{U}_1 + F_2(T)\overline{U}_2) +$$
$$q_1q_2\left((\Pr(1,0) + \Pr(1,1))\overline{U}_1 + (\Pr(0,1) + \Pr(1,1))\overline{U}_2\right) - \eta_1q_1 - \eta_2q_2 - \xi T$$

$$\Rightarrow ES = q_1F_1(T)\overline{U}_1 + q_2F_2(T)\overline{U}_2 - \eta_1q_1 - \eta_2q_2 - \xi T$$

Which is the same expression obtained assuming the servers are independent. In other words, correlation in response time does not affect the expression for the expected surplus in the case with two servers.

Case 2: Three Servers

If the servers are independent, then the expected surplus can be expressed as follows:

$$ES = q_1F_1(T)\overline{U}_1 + q_2F_2(T)\overline{U}_2 + q_3F_3(T)\overline{U}_3 - \eta_1q_1 - \eta_2q_2 - \eta_3q_3 - \xi T$$

Let $\Pr(i1 \in \{0,1\}, i2 \in \{0,1\}, i3 \in \{0,1\})$ denote, as before, the probability of observing a specific retrieval set. Suppose the server response times are NOT independent. Then the expected surplus in this case of independent servers is:

$$ES = (1-q_3)\left(q_1F_1(T)\overline{U}_1 + q_2F_2(T)\overline{U}_2 + q_3F_3(T)\overline{U}_3\right) + q_3(1-q_1)(1-q_2)F_3(T)\overline{U}_3 +$$
$$q_3(q_1)(1-q_2)\left(\Pr(0X1)\overline{U}_3 + \Pr(1X0)\overline{U}_1 + \Pr(1X1)(\overline{U}_1 + \overline{U}_3)\right) +$$
$$q_3(1-q_1)(q_2)\left(\Pr(X01)\overline{U}_3 + \Pr(X10)\overline{U}_2 + \Pr(X11)(\overline{U}_2 + \overline{U}_3)\right) +$$
$$q_3q_1q_2\left(\begin{array}{l}\Pr(000)\cdot 0 + \Pr(100)\overline{U}_1 + \Pr(010)\overline{U}_2 + \Pr(001)\overline{U}_3 + \\ \Pr(110)(\overline{U}_1 + \overline{U}_2) + \Pr(011)(\overline{U}_2 + \overline{U}_3) + \Pr(101)(\overline{U}_1 + \overline{U}_3) + \Pr(111)(\overline{U}_1 + \overline{U}_2 + \overline{U}_3)\end{array}\right) -$$
$$\eta_1q_1 - \eta_2q_2 - \xi T$$

Noting that $\Pr(0X1) + \Pr(1X1) = \Pr(XX1) = F_3(T)$ and similarly for servers 1 and 2, we can simplify the above expression:

$$ES = (1-q_3)\left(q_1F_1(T)\overline{U}_1 + q_2F_2(T)\overline{U}_2 + q_3F_3(T)\overline{U}_3\right) + q_3(1-q_1)(1-q_2)F_3(T)\overline{U}_3 +$$
$$q_3(q_1)(1-q_2)\left(F_3(T)\overline{U}_3 + F_1(T)\overline{U}_1\right) +$$
$$q_3(1-q_1)(q_2)\left(F_3(T)\overline{U}_3 + F_2(T)\overline{U}_2\right) +$$
$$q_3q_1q_2\left(F_1(T)\overline{U}_1 + F_2(T)\overline{U}_2 + F_3(T)\overline{U}_3\right) -$$
$$\eta_1q_1 - \eta_2q_2 - \xi T$$

Summing together all terms with $F_i(T)\overline{U}_i$

$$ES = q_1F_1(T)\overline{U}_1 + q_2F_2(T)\overline{U}_2 + q_3F_3(T)\overline{U}_3 - \eta_1q_1 - \eta_2q_2 - \eta_3q_3 - \xi T$$

which is the same expression obtained assuming the servers are independent. In other words, correlation in response time does not affect the expression for the expected surplus in the case with three servers. Using a similar approach, the argument can be extended to higher dimensions.

Correlated Relevance Scores and Independent Response Times.

Case 1: Two Servers

If the servers are independent, then the expected surplus can be expressed as follows:

$$ES = q_1 F_1(T)\overline{U}_1 + q_2 F_2(T)\overline{U}_2 - \eta_1 q_1 - \eta_2 q_2 - \xi T$$

Suppose the document relevance scores are NOT independent across the two servers. As a result, the expected surplus from the evaluated documents of server 1 less the cognitive cost remains $\overline{U}_1$ but is no longer independent of $\overline{U}_2$, and vice-versa. Then the expected surplus can be expressed as follows,

$$\begin{aligned} ES = {} & q_1(1-q_2)F_1(T)\overline{U}_1 + (1-q_1)q_2 F_2(T)\overline{U}_2 + \\ & q_1 q_2 \left( \begin{array}{l} F_1(T)(1-F_2(T))\overline{U}_1 + (1-F_1(T))F_2(T)\overline{U}_2 \\ + F_1(T)F_2(T)\left( E\left[ \sum_{k=1..d_1} I_{\{U_{1,k} \geq \lambda A\}} \cdot (U_{1,k} - \lambda A) \right] + E\left[ \sum_{k=1..d_2} I_{\{U_{2,k} \geq \lambda A\}} \cdot (U_{2,k} - \lambda A) \right] \right) \end{array} \right) - \\ & \eta_1 q_1 - \eta_2 q_2 - \xi T \end{aligned}$$

Because the utility from documents is additive, the correlation in relevance score across servers does not affect the expected utility from any individual document or a group of documents. As a result,

$$\begin{aligned} ES = {} & q_1(1-q_2)F_1(T)\overline{U}_1 + (1-q_1)q_2 F_2(T)\overline{U}_2 + \\ & q_1 q_2 \left( \begin{array}{l} F_1(T)(1-F_2(T))\overline{U}_1 + (1-F_1(T))F_2(T)\overline{U}_2 \\ + F_1(T)F_2(T)\left(\overline{U}_1 + \overline{U}_2\right) \end{array} \right) - \\ & \eta_1 q_1 - \eta_2 q_2 - \xi T \end{aligned}$$

$$\Rightarrow ES = q_1 F_1(T)\overline{U}_1 + q_2 F_2(T)\overline{U}_2 - \eta_1 q_1 - \eta_2 q_2 - \xi T$$

which is the same expression for expected surplus obtained if the two servers have independent relevance scores.

Case 2: Three Servers

If the servers are independent, then the expected surplus can be expressed as follows:

$$ES = q_1 F_1(T)\overline{U}_1 + q_2 F_2(T)\overline{U}_2 + q_3 F_3(T)\overline{U}_3 - \eta_1 q_1 - \eta_2 q_2 - \eta_3 q_3 - \xi T$$

Suppose the document relevance scores are NOT independent across the servers. As a result, the expected surplus from the evaluated documents of server 1 less the cognitive cost remains $\overline{U}_1$ but is no longer independent of $\overline{U}_2$ or $\overline{U}_3$, and similarly for other servers. Then the expected surplus can be expressed as follows,

$$\begin{aligned}
ES = {} & q_1(1-q_2)(1-q_3)F_1(T)\overline{U}_1 + (1-q_1)q_2(1-q_3)F_2(T)\overline{U}_2 + (1-q_1)(1-q_2)q_3F_3(T)\overline{U}_3 + \\
& q_1q_2(1-q_3)\left(\begin{array}{l} F_1(T)(1-F_2(T))\overline{U}_1 + (1-F_1(T))F_2(T)\overline{U}_2 \\ + F_1(T)F_2(T)\left(E\left[\sum_{k=1..d_1} I_{\{U_{1,k}\geq\lambda A\}}\cdot(U_{1,k}-\lambda A)\right] + E\left[\sum_{k=1..d_2} I_{\{U_{2,k}\geq\lambda A\}}\cdot(U_{2,k}-\lambda A)\right]\right)\end{array}\right) + \\
& q_1(1-q_2)q_3\left(\begin{array}{l} F_1(T)(1-F_3(T))\overline{U}_1 + (1-F_1(T))F_3(T)\overline{U}_3 \\ + F_1(T)F_3(T)\left(E\left[\sum_{k=1..d_1} I_{\{U_{1,k}\geq\lambda A\}}\cdot(U_{1,k}-\lambda A)\right] + E\left[\sum_{k=1..d_3} I_{\{U_{3,k}\geq\lambda A\}}\cdot(U_{3,k}-\lambda A)\right]\right)\end{array}\right) + \\
& (1-q_1)q_2q_3\left(\begin{array}{l} F_2(T)(1-F_3(T))\overline{U}_2 + (1-F_2(T))F_3(T)\overline{U}_3 \\ + F_2(T)F_3(T)\left(E\left[\sum_{k=1..d_2} I_{\{U_{2,k}\geq\lambda A\}}\cdot(U_{2,k}-\lambda A)\right] + E\left[\sum_{k=1..d_3} I_{\{U_{3,k}\geq\lambda A\}}\cdot(U_{3,k}-\lambda A)\right]\right)\end{array}\right) - \\
& \eta_1 q_1 - \eta_2 q_2 - \eta_3 q_3 - \xi T
\end{aligned}$$

Once again, the additive utility function combined with a constant marginal cost of evaluation implies that the expected utility from any individual document or a group of documents is not affected. As a result,

$$\begin{aligned}
ES = {} & q_1(1-q_2)(1-q_3)F_1(T)\overline{U}_1 + (1-q_1)q_2(1-q_3)F_2(T)\overline{U}_2 + (1-q_1)(1-q_2)q_3F_3(T)\overline{U}_3 + \\
& q_1q_2(1-q_3)\left(F_1(T)\overline{U}_1 + F_2(T)\overline{U}_2\right) + \\
& q_1(1-q_2)q_3\left(F_1(T)\overline{U}_1 + F_3(T)\overline{U}_3\right) + \\
& (1-q_1)q_2q_3\left(F_2(T)\overline{U}_2 + F_3(T)\overline{U}_3\right) - \\
& \eta_1 q_1 - \eta_2 q_2 - \eta_3 q_3 - \xi T
\end{aligned}$$

$$\Rightarrow ES = q_1 F_1(T)\overline{U}_1 + q_2 F_2(T)\overline{U}_2 + q_3 F_3(T)\overline{U}_3 - \eta_1 q_1 - \eta_2 q_2 - \eta_3 q_3 - \xi T$$

which is the same expression for expected surplus obtained if the three servers have independent relevance scores.

In summary, the additive nature of the utility function combined with the constant marginal cost of evaluation implies that expected contribution of each server can be

independently computed without worrying about correlations. This changes, for example, if the cognitive cost function is non-linear. That is the subject of the next section of this appendix.

**II. 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. 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 ($P_{Max_j}$) on *P*. Feinberg and Huber (1996) call this the quota cutoff criteria. We model this by assuming $C_j(P) = \begin{cases} \lambda_j AP & if\ P \le P_{Max_j} \\ \infty & if\ P > P_{Max_j} \end{cases}$. This compound stopping rule can be treated as an extreme case of the convexity in cognitive costs described earlier. As before, we drop the subscript *j* and it is implied that all analysis is at the user or segment level.

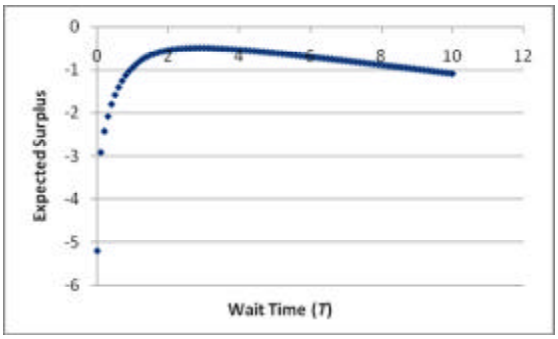
**5.1 Determining Optimal *q* and *T***

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. Figure OA1 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 associated with each $T$ is obtained by averaging the surplus realized in 10,000 runs with the same parameters. The expected surplus is maximized when $T^*$ = 3.0 seconds.

**Figure OA1: 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 OA1, 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 OA1. 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.[1] 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 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.

[1] 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.

**Table OA1: 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 | $U_{jk}(\boldsymbol{q})$ | 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 | $U_{jk}(\boldsymbol{q})$ | 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 | $U_{jk}(\boldsymbol{q})$ | 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 | $U_{jk}(\boldsymbol{q})$ | 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 | $U_{jk}(\boldsymbol{q})$ | 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 | $U_{jk}(\boldsymbol{q})$ | 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 | $U_{jk}(\boldsymbol{q})$ | 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 | $U_{jk}(\boldsymbol{q})$ | 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 | $U_{jk}(\boldsymbol{q})$ | 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 | $U_{jk}(\boldsymbol{q})$ | 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 | $U_{jk}(\boldsymbol{q})$ | 0.56 | 0.10 | 0.03 | 0.04 | - | - | - | - | - | 0.56 | - | - | - | - | - |
| | | | $T_i$ | 0.00 | Inf | Inf | Inf | - | - | - | - | - | 0.22 | - | - | - | - | - |
| 4 | 3.9 | 0.49 | $U_{jk}(\boldsymbol{q})$ | 0.57 | 0.10 | - | 0.04 | - | - | - | - | - | 0.58 | - | - | - | - | - |
| | | | $T_i$ | 0.00 | 23.25 | - | Inf | - | - | - | - | - | 0.22 | - | - | - | - | - |
| 3 | 3.8 | 0.56 | $U_{jk}(\boldsymbol{q})$ | 0.56 | 0.10 | - | - | - | - | - | - | - | 0.57 | - | - | - | - | - |
| | | | $T_i$ | 0.00 | 15.47 | - | - | - | - | - | - | - | 0.22 | - | - | - | - | - |
| 2 | 4 | 0.57 | $U_{jk}(\boldsymbol{q})$ | 0.58 | - | - | - | - | - | - | - | - | 0.59 | - | - | - | - | - |
| | | | $T_i$ | 0.00 | - | - | - | - | - | - | - | - | 0.21 | - | - | - | - | - |
| 1 | 1.7 | 0.29 | $U_{jk}(\boldsymbol{q})$ | 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, it 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**

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 $q$.[2]

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 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 appears to be effective. Simultaneously, it helps significantly reduce computational complexity 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 $\xi, \lambda, \eta_i$ and also considered convex cognitive cost functions of the form $C(P) = \lambda P^k$ where $k > 1$. The heuristic continues to perform well in these additional tests as well. The results in this Section suggest that the analytical results from Section 3 can be adapted to more complex environments.

**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 |

[2] 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.

References

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

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

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

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. Proc. Of *SIGIR 1984*: 421-433.