Improving Web Catalog Design for Easy Product Search

I. Robert Chiang and Manuel A. Nunez

July 21, 2004

Department of Operations and Information Management
School of Business
University of Connecticut
2100 Hillside Road Unit 1041, Storrs, CT 06269-1041
robert.chiang@business.uconn.edu
manuel.nunez@business.uconn.edu
Improving Web Catalog Design for Easy Product Search

Abstract

Building intuitive web sites is important for online businesses as positive experiences toward the virtual storefront could translate to customer goodwill and repeat visits. Streamlining web site navigation is further motivated by the availability of comprehensive site visit traces such as shopping carts and click stream logs. The focus of this study is to incorporate such sales and browsing patterns for the autonomous design of web catalog. The objective is to reduce the expected click count it takes to find related items. We first model catalog design as a task of placing items onto catalog pages, leading to a quadratic assignment formulation. For a more extensive redesign, such as when the number and location of links on each page are also decision variables, we propose a two-stage optimization procedure to ensure that click count reduction is not achieved at the cost of excessive page cluttering. Our analysis reveals that an optimized catalog is robust against shifts in browsing behavior and that simplistic interfaces are preferred for users at either end of the experience spectrum (very low or highly experienced users). Using genetic algorithms, we found catalog designs that outperform the designs found with an integer programming heuristic for at least 12 percent.

KEY WORDS AND PHRASES: catalog design, network design, quadratic assignment problem, genetic algorithms, electronic commerce.
1 Introduction

1.1 Motivation

In a time following the “irrational exuberance” fuelled by speculations toward the commercialized Internet, many of the dubbed “dot goners” have turned out to be thriving and profitable enterprises [10, 20]. Many key indicators, including the volume of online B2B/B2C transactions [14] and the e-business’ contribution to the productivity gain [21], have either met or eclipsed bold pre-bubble predictions. With over ten percent of worldwide population having gained Internet access [28], the immediacy and ubiquity of the Internet could render electronic commerce an ever-increasing role in the global economy.

From the onset, many online retailers have experienced poor order fulfillments [24], overwhelmed or crashed web servers, and sloppy control of sensitive data. Previous research has thus proposed innovations such as supply chain alliances [4], server resource optimization [30], and data security enhancement [22]. While back-end blunders have subsequently dwindled, poor web design remains a recurring source of customer aggravation. Reportedly, poor web interface has resulted in missed business opportunities [31] and low browsing-to-sales conversions [16], as customers either were not aware of or had difficulty locating products or services they intended to evaluate or purchase.

Being the main conduit with prospects and customers, a virtual storefront should be engaging and intuitive to attract repeated visits and increase sales. Web site design involves psychological considerations such as content richness and aesthetics of individual pages [2, 15]. Another design consideration is the effective placement of hyperlinks - an issue especially critical for the web catalog [18] to facilitate product search. To showcase a company’s offerings, not only the catalog should enable customers to reach the pages of interest quickly; it should also provide smooth transition toward pages that are deemed “related.”

The basic means of online catalog navigation has been the category/subcategory hyperlinks on the side bar. For example, to find a digital camera would involve a click sequence similar to “Electronics > Camera & Photo > Digital Camera > 5 Megapixel or higher”. Alternatively, one can submit “6MP digital camera optical zoom” to the search engine for a list of recommendations. Relying on a category hierarchy or a search engine can be considered a “passive” navigation design, as both rely mainly on querying the product database with intrinsic attributes such as brand name, functionality, and price.

An obvious shortcoming of passive hyperlinks is that to view another item several “hops” may have to be traversed in between. As for using a site search engine, to ensure good coverage a search result may lead to a lengthy recommendation that discourages further exploration [29]. Using a search engine also requires interleaving two input devices, which potentially could disrupt the “flow” of catalog browsing [26].

To make navigation more intuitive, web sites have adopted the “Related Items” or
“You May Also Like” section in their catalog for a quick click-through. This feature is illustrated by a premier e-tailor’s catalog page shown in Figure 1. It appears that a wide variety of products can be considered in the related item section, ranging from substitutable (i.e. similar glassware within the product line) to complementary (i.e. coordinated napkin and dinnerware). The issue is thus to recognize and reflect how items are correlated with one another in the catalog design. While item placement could be a manual task relying on human expertise, going beyond a boutique operation would require a more systematic approach.

1.2 Objectives and contributions

The objective of this research is to incorporate consumer browsing and purchasing behaviors for the automation of web catalog design. Previously, the mining of business financial and operational data has been attributed to improved business decisions, such as on more precise expansion planning [27] and better logistics efficiency [5]. Browsing behavior has also been used for online advertisement design [17] and customer relationship management [23]. The navigation design issues studied here allow for cross-product recommendation found in collaborative filtering literature [3, 25] while in addition ensures that certain topological constraints are satisfied.

We model a web catalog as a network consisting of nodes (i.e., catalog pages) and edges (i.e. page hyperlinks). Under consideration are three key factors that affect catalog navigation: items correlation, catalog topology, and user experience. How items correlate with one another can be readily observed since web servers routinely capture site visit and transaction histories. Naturally, items that are often jointly purchased or evaluated (i.e., comparison-shopped in the same browsing session) should be placed “close” to each other.

Catalog topology describes where and how densely catalog pages are connected by the hyperlinks. To see how topology impacts navigation, consider a fully connected catalog. Although it renders one-click access to all other pages, it also results in severe cluttering that defeats the usefulness of “quick” links. Thus, if an item relates little with the rest, few links should stem from its page. On the other hand, if an item is often viewed or purchased with many others, more links may be preferred even if doing so increases local page cluttering.

As for user experience, it reveals the effect of limited (page and link) visibility of online catalog browsing. For example, a first-time visitor may find it more difficult to locate items than returned customers, thus incurring more page clicks. We propose two approaches to accommodate the difference in user experience. The first is a generic approach in which one catalog is designed to best match the average browsing behavior of all users. The second is a customized approach in which several catalogs are generated and dynamically assigned according to the experience level of each individual. The tradeoff between the two approaches depends on whether the improved ease of navigation justifies the cost of
generating/managing multiple catalogs.

One key contribution of this study is that we propose two optimization models for catalog navigation depending on whether or not the placement of hyperlinks is to be determined. The first model applies when the topology is highly structured, such as when a web site uses a fixed number of hyperlinks on each page. If so, navigation design entails finding the best assignment of items onto catalog pages, a task that can be modeled using a quadratic assignment problem (QAP) formulation. Since QAP is a well-known $NP$-hard problem, finding an exact solution could be computationally expensive. Hence, we use genetic algorithms and other heuristics to find approximated solutions. To evaluate approximation error as well as computational performance, we compare the GA solutions to the optimal solutions obtained from using complete enumeration over a sample of small size problems.

The second optimization model takes place when the catalog topology is also subject to change. Since the placement of hyperlinks needs to be determined before the pair-wise distances between pages can be known, a two-stage optimization is proposed: a GA heuristic is used to search for an optimal catalog topology; at the same time, a QAP algorithm is used by the GA as a subroutine to determine a near-optimal item assignment.

Figure 1: Typical layout of an online catalog page.
Numerical results for both models reveal that our solution procedures generate near-optimal catalog designs. The procedures also consistently outperform an IP-based, polynomial-time heuristic by at least 12% percent in terms of objective value. An interesting finding is that, in some cases, the catalog, once optimized, is quite robust against shifts in average user browsing behavior. This suggests that a small number of customized catalogs would be sufficient to serve a wide range of users. Also, we found that a more sparse catalog topology should be favored when interacting with novice users and that when page cluttering carries high penalty, the resulting catalog has a tree topology. Overall, our approach leads to an autonomous and efficient design process that could help generating customized or even personalized catalogs.

The rest of the paper is organized as follows. Section 2 defines the pair-wise distance between web pages, which is followed by the formulation of an assignment optimization model. The solution procedure and the numerical results are then presented. The optimization problem that aims to improving the overall catalog topology is shown in Section 3, followed by a discussion on a solution procedure and numerical results. Section 4 discusses the tradeoff between a generic catalog and a set of customized ones in further detail. The last section provides concluding remarks as well as possible directions for future research.

2 The assignment model

In this section we consider a situation in which some “design cues” are copied from the existing catalog. To be visually consistent, a site designer may not alter the number of thumbnails in the “Related Items” section when the catalog is undergoing changes. In such a case, we show that catalog navigation design can be modeled as a quadratic assignment problem. We start with a discussion on how the pair-wise distances between pages can be measured; a mathematical programming formulation is then proposed with the objective of reducing the expected click count to find related items.

2.1 Distance between pages

By analyzing logs from past visits, it is possible to recognize items that are frequently viewed or purchased within a session of user visits. Items that are “clustered” should be placed in close proximity to facilitate their search. It is thus necessary to measure the distance between pages before item placement could take place. We represent the catalog topology as an undirected network $G = (N, E)$ with $|N| = n$ nodes and $|E| = m$ edges (i.e., the hyperlinks). Figure 2 shows a simple catalog topology with $n = 7$ and $m = 6$.

Let the distance between page $i$ and page $j$ be denoted $D_{ij}$. The distance can be measured in many ways. The best-case scenario corresponds to when a customer uses the shortest distance (i.e., the least number of clicks) between pages. The shortest distances can be obtained by performing a breadth-first search. For a catalog with $n$ nodes, the
algorithm incurs in a polynomial running time of $O(n^2)$ [1, 7]. We use the matrix $D^B$ to represent the minimal click counts between pages. For example, in the catalog described in Figure 2, we have

\[
D^B = \begin{bmatrix}
0 & 1 & 1 & 2 & 2 & 2 & 2 \\
1 & 0 & 2 & 1 & 1 & 3 & 3 \\
1 & 2 & 0 & 3 & 3 & 1 & 1 \\
2 & 1 & 3 & 0 & 2 & 4 & 4 \\
2 & 1 & 3 & 2 & 0 & 4 & 4 \\
2 & 3 & 1 & 4 & 4 & 0 & 2 \\
2 & 3 & 1 & 4 & 4 & 2 & 0
\end{bmatrix}.
\] (1)

The matrix (1) shows the minimum number of pages traversed from source to destination. For instance, the distance between page 1 and page 7 is 2, as shown on the upper right corner of $D^B$. We also assume zero distance between a page and itself, thus the zero values on the diagonal.

Since web interface permits only limited visibility, page traversing would likely incur in higher click counts than those prescribed by $D^B$. We assume the worst-case scenario occurs when the catalog is traversed by following a Markovian random walk. The resulting distance matrix, denoted $D^W$, is thus the first passage time from a source page to the destination. To compute $D^W$, let $P$ denote the matrix of transition probabilities for the corresponding Markov chain. If node $k$ in the network has $d$ ($d > 0$) outgoing hyperlinks, we have

\[
P_{kl} := \begin{cases}
\frac{1}{d} & \text{if } (k,l) \in E, \\
0 & \text{if } (k,l) \notin E,
\end{cases}
\] (2)
for all $k \in N$. For example, the matrix $P$ for the catalog in Figure 2 is

$$P = \begin{bmatrix}
0 & 1/2 & 1/2 & 0 & 0 & 0 & 0 \\
1/3 & 0 & 0 & 1/3 & 1/3 & 0 & 0 \\
1/3 & 0 & 0 & 0 & 0 & 1/3 & 1/3 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}.$$  \hspace{1cm} (3)

To compute the matrix $D^W$ we can solve a collection of $n$ linear systems for a total running time of $O(n^3)$ [13]. Notice that because of the definition of $P$, the Markov chain is irreducible. Let $P_{-l}$ be a matrix identical to $P$ except that zeros replace the $l$-th row, and let $e_{-l}$ be identical to the all-ones vector except that a zero replaces the $l$-th entry. If $D^W_l$ denotes the $l$-th column of the distance matrix $D^W$, $D^W_l$ can be computed by solving the following linear system

$$D^W_l = e_{-l} + P_{-l}D^W_l,$$

for all $l$. Using this procedure for the catalog in Figure 2 we have

$$D^W = \begin{bmatrix}
0 & 7 & 7 & 14 & 14 & 14 & 14 \\
5 & 0 & 12 & 9 & 9 & 19 & 19 \\
5 & 12 & 0 & 19 & 19 & 9 & 9 \\
6 & 1 & 13 & 0 & 10 & 20 & 20 \\
6 & 1 & 13 & 10 & 0 & 20 & 20 \\
6 & 13 & 1 & 20 & 20 & 0 & 10 \\
6 & 13 & 1 & 20 & 20 & 10 & 0
\end{bmatrix}.$$  \hspace{1cm} (4)

In practice the actual click count would likely fall in between those distances prescribed by $D^B$ and $D^W$. We also use a weighted distance matrix defined as

$$D(\theta) := \theta D^B + (1 - \theta) D^W.$$  \hspace{1cm} (5)

The choice of the value of $\theta$ would depend on a user’s experience with the web site as well as on the intended use of the catalog. For example, a B2B site getting routine purchases could expect a high value (up to 1) of $\theta$ since returned buyers browse and find items more easily. On the other hand, a catalog should be optimized for a lower $\theta$ value (toward 0) if it were designed for a general store encouraging visitors to explore the different “aisles.” We here refer $\theta$ as the experience index of a web site visitor. Section 4 provides further details on choosing an adequate value of $\theta$ for the model in practice.
2.2 Quadratic assignment problem

As discussed earlier, item assignments should be influenced by how they relate to each other. To capture such a correlation, we use a frequency matrix denoted by $F$ of size $n \times n$ to show past shopping patterns and browsing behaviors. An entry $F_{ij}$ ($\geq 0$ for all $i, j$) represents the proportion of occurrences that the pair of items $(i, j)$ appeared in the same shopping cart or were evaluated in the same comparison-shopping session. Summing over $i$ and $j$, we have

$$\sum_{i,j} F_{ij} = 1.$$ 

To record the page to which each item is assigned, we use an assignment matrix denoted by $X$ of size $n \times n$, where $X_{ik}$ is 1 if item $i$ is assigned to page $k$, and $X_{ik} = 0$ otherwise.

Given an arbitrary distance matrix $D$ and a frequency matrix $F$, we model the process of improving catalog navigation as finding the optimal assignment matrix $X^*$ that solves the following optimization problem:

\[
\text{(QAP)} \quad \min z(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} F_{ij} \sum_{k=1}^{n} \sum_{l=1}^{n} D_{kl} X_{ik} X_{jl}, \\
\text{s.t.} \\
\sum_{l=1}^{n} X_{il} = 1, \forall i, \\
\sum_{i=1}^{n} X_{il} = 1, \forall l, \\
X_{il} \in \{0, 1\}, \forall i, l.
\]

For a given frequency matrix $F$ and distance matrix $D$, how items are assigned to pages (i.e., the assignment matrix $X$) influences the objective function. In QAP, the objective function $z(X)$ represents the average number of clicks it takes to find a pair of items of interest for a given $X$. The constraints ensure that one and only one item is assigned to each page. A more compact representation is

\[
\text{(QAP)} \quad \min z(X) = \text{trace} \left( F X D X^T \right), \\
\text{s.t.} \\
x e = e, \quad x^T e = e, \quad X_{il} \in \{0, 1\}, \forall i, l;
\]

where $e$ denotes the $n$-dimensional all-ones vector.
To study the complexity of solving QAP, it is easy to see that any instance of the quadratic assignment problem [12] can be polynomially transformed to an instance of QAP. Since the quadratic assignment problem is \(NP\)-hard [11], it follows that QAP is also \(NP\)-hard. Hence, finding exact solutions to QAP could be computationally expensive. Instead, we propose GA-based heuristics to find approximate solutions. Next, Section 2.4 shows computational results.

2.3 Genetic algorithm and IP heuristic

Genetic algorithms [6, 19] have been widely used as a guided search heuristic for combinatorial optimization problems. Although it is fairly easy to apply GA in many unconstrained optimization situations, care must be taken for the case of constrained optimization, as it could be costly to repair infeasible solutions after standard GA operations such as crossover and mutation. We use a chromosome representation consisting of permutation vectors. In other words, every feasible solution to QAP is represented as a \(n\)-dimensional permutation vector \(v\), where entry \(v_i\) denotes the item assigned to page \(i\). For example, in the catalog described by Figure 2, a possible chromosome is \(v = (2,5,1,7,6,3,4)\). In this example \(v_3 = 1\), meaning that item 1 is assigned to page 3. Notice that the total number of possible chromosomes/feasible solutions is \(n!\), so the search space grows fast as a function of the number of items.

There are several standard permutation-preserving crossover operations available (see for instance IDX and LOX in [6]). We use a variant called median crossover (MDX) introduced in [8, 9] for its ability to combine crossover with local search for permutation chromosomes. Moreover, it has been shown that MDX provides excellent results for quadratic assignment problems [8]. In MDX, the network is divided into two cohesive parts according to the median distance of the network nodes with respect to a fixed pivot node. Then, given two parents to be crossed, the parents are split according to the median network parts, and the offsprings are generated by taking the segment under the median from one parent and the segment over the median from the other parent. Since a crossover depends on which node is chosen to be pivot, a local search is performed in which only the two best offsprings from all the possible crossovers are selected. As for the mutation operator, in order to keep mutated chromosomes feasible we use random swapping of the assignment of two items in a given chromosome vector. We also use elitism [19] to retain best candidate solutions across generations.

To compare the performance of our GA we use a simple polynomial-time integer programming heuristic to find approximate solutions to QAP. In our IP heuristic we solve two linear assignment problems. In the first problem items are assigned to pages solely based on the distance matrix and in the second problem items are assigned to pages solely based on the frequency matrix. Clearly both problems can be solved in polynomial time. Once the two problems are solved, we choose the solution with the lowest QAP objective.
2.4 Numerical results

We start with the task of re-designing a small catalog. A group of 12 items is extracted from the web site of a major home-improvement center. Figure 3 illustrates the original catalog topology. To populate the frequency matrix $F$ we first form clusters among items and then assign high frequency values for items that are in the same cluster. We also assign low frequency values for items across clusters. Table 1 describes the items (as numbered in Figure 3) as well as the clusters. For example, the frequency between a fish tape and a fish tape leader, $F_{10,11}$, is set to 10 because they are placed in the same cluster whereas the frequency between an indoor timer and an electrical tape, $F_{3,4}$, is set to 1 because they are placed in different clusters.

Table 2 compares the objective values obtained from different heuristics. The expected click-counts range from 2 to more than 8. In each case, a local search GA was able to identify the optimal item assignment by evaluating only 2000 (a population size of 20 times 100 generations) among the more than $12! \approx 4.7 \times 10^8$ possible assignments. The GA outperforms the IP heuristic by at least 12 percent. Using a desktop computer running on 3.2GHz Pentium 4 processor and 1GB RAM, both GA and IP heuristics take only several seconds whereas the complete enumeration registers a run time of more than four hours.

To ensure that the computation time and solution quality for the GA procedure scales
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Cluster</th>
<th>Item</th>
<th>Description</th>
<th>Cluster</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Floodlight bulbs</td>
<td>A</td>
<td>7</td>
<td>Tape rolls</td>
<td>C</td>
</tr>
<tr>
<td>2</td>
<td>Bulb changer kit</td>
<td>B</td>
<td>8</td>
<td>Vinyl tape</td>
<td>C</td>
</tr>
<tr>
<td>3</td>
<td>Indoor timer</td>
<td>B</td>
<td>9</td>
<td>All purpose tool</td>
<td>D</td>
</tr>
<tr>
<td>4</td>
<td>Electrical tape</td>
<td>C</td>
<td>10</td>
<td>Fish tape</td>
<td>D</td>
</tr>
<tr>
<td>5</td>
<td>Wire connector kit</td>
<td>C</td>
<td>11</td>
<td>Fish tape leader</td>
<td>D</td>
</tr>
<tr>
<td>6</td>
<td>Cable tie kit</td>
<td>C</td>
<td>12</td>
<td>Scissors with stripping</td>
<td>D</td>
</tr>
</tbody>
</table>

Table 1: Items description for 12-node example.

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>Objective value ($Z$)</th>
<th>Relative error**</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enumeration</td>
<td>0.5</td>
<td>8.324*</td>
</tr>
<tr>
<td>0.9</td>
<td>2.775*</td>
<td>N.A.</td>
</tr>
<tr>
<td>0.95</td>
<td>2.082*</td>
<td>N.A.</td>
</tr>
<tr>
<td>GA</td>
<td>0.5</td>
<td>8.324</td>
</tr>
<tr>
<td>0.9</td>
<td>2.775</td>
<td>0.00%</td>
</tr>
<tr>
<td>0.95</td>
<td>2.082</td>
<td>0.00%</td>
</tr>
<tr>
<td>IP Heuristic</td>
<td>0.5</td>
<td>9.363</td>
</tr>
<tr>
<td>0.9</td>
<td>3.246</td>
<td>16.97%</td>
</tr>
<tr>
<td>0.95</td>
<td>2.482</td>
<td>19.21%</td>
</tr>
</tbody>
</table>

* Optimal objective value obtained by complete enumeration
** With respect to optimal solution

Table 2: Click count performance for the 12-node problem.

well with the problem size, we randomly generated a 120-node test problem. We assume 10 categories with 12 products in each category. Furthermore, we randomly assigned values of either 5 or 10 to pairs of elements within the same category in the frequency matrix. On the other hand, we assigned 1 to pairs of elements across categories. Finally, we assumed a three-link-per-page catalog topology. Since solving a 120-node problem is significantly more difficult than that of a 12-node problem, we increased the GA population size to 100.

Table 3 shows the relative performance of our GA and IP heuristics at this problem size. Notice that complete enumeration is not feasible within a reasonable time frame. From Table 3, the GA procedure on average generates an objective value that is 4.7% lower than the IP solution if the number of generations is set at 1000. If the number of generations reaches 5000, the GA solution on average is 16.5% lower than the IP heuristic, closely matching the improvement reported in Table 2 for the 12-node case. However the execution time for the GA increases with the number of generations. While the GA incurs
Table 3: Click count comparison for the 120-node problem.

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>Improvement over IP heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GA (1000 iterations)</td>
</tr>
<tr>
<td>0.5</td>
<td>4.85%</td>
</tr>
<tr>
<td>0.9</td>
<td>3.68%</td>
</tr>
<tr>
<td>0.95</td>
<td>5.79%</td>
</tr>
</tbody>
</table>

in a comparable execution time with the IP heuristic at several minutes when the number of generations is 1000, increasing the number of generations lengthens the execution time proportionally.

3 The design model

Although the assignment model discussed in the previous sections has shown to significantly improve catalog navigation, it operates under a restricted catalog topology. For some web sites, the number of hyperlinks is not fixed but depends on which item is assigned to the catalog page. When catalog topology is also a decision variable, the main concern is to ensure that page links would provide sufficient connectivity without incurring excessive cluttering.

3.1 Model description

To model how catalog pages are connected we use a node-node adjacency matrix, denoted by $Y$, to represent the placement of hyperlinks. We set $Y_{kl}$ to 1 if page $k$ and page $l$ are directly connected by a network edge, otherwise we set $Y_{kl}$ to 0. We assume that $Y$ is chosen such that the resulting network is connected. For a given $Y$, we define the weighted distance matrix similarly to (5) as

$$D(Y) := \theta D^B(Y) + (1 - \theta) D^W(Y),$$

where $D^B(Y)$ and $D^W(Y)$ are the best and worst case distance matrices induced by the topology $Y$.

In addition to placement, another issue with hyperlinks that affects navigation is the number of links. While too few links may result in parts of the catalog difficult to reach, having too many links also creates page cluttering and potentially makes the catalog confusing to traverse. We denote by $C(Y)$ the cluttering cost. For simplicity, we assume that
$C(Y)$ is proportional to the number of links in the network, or

$$C(Y) = c \sum_{k=1}^{n} \sum_{l=1}^{n} Y_{kl},$$

where $c$ is a positive constant.

The design optimization model is described as follows:

$$(DP) \quad \min \quad z(X,Y) = \text{trace} \left( FXD(Y)X^T \right) + C(Y),$$

s.t.

$$Xe = e, \quad (6)$$

$$X^Te = e, \quad (7)$$

$$NV^k = b^k, \forall k, \quad (8)$$

$$\sum_{k=1}^{n} V^k \leq n(n-1)Y, \quad (9)$$

$$V^k \geq 0, X_{il}, Y_{kl} \in \{0,1\}, \forall i,k,l; \quad (10)$$

In DP, equations (6) and (7) are the assignment constraints similarly considered in problem QAP. Constraints (8) and (9) are to ensure connectivity, that is, each node can be reached by all others. This can be further explained as follows. Assume that there are $n - 1$ units of a unique commodity to be shipped from a node, and there are demands of 1 unit of such commodity at all other nodes. If the commodity originating from node $k$ is sent via the edge between page $i$ and page $j$, then $V^k_{ij}$ represents the flow of such a commodity through edge $(i,j)$. Constraint (8) corresponds to the flow-balance equations for each commodity, where the matrix $N$ is the node-arc incidence matrix, and the vector $b^k$ corresponds to the demand-supply vector at each of the nodes. Because the flow through each arc is at most $(n - 1)$ units for each commodity, it follows that $n(n-1)$ is an upper bound for the total flow of commodities through edge $(i,j)$. Hence, inequality (9) implies that $Y_{ij} = 1$ if arc $(i,j)$ is used to ship at least one unit of a commodity, and that there is no flow of commodities through arc $(i,j)$ if $Y_{ij} = 0$.

Because solving problem DP requires identifying the best item assignment for each topology being evaluated, the problem QAP becomes a subproblem of DP. It can be deduced that DP is at least an $NP$-hard problem and would require some heuristic to find approximate solutions in polynomial time. We propose a solution procedure that involves two GAs in the next section.

### 3.2 Genetic algorithm

To find approximate solutions to the design problem DP we use a GA to search in the space of connected network topologies. The chromosome representation of each feasible solution
in the search space consists of a binary matrix $Y$ representing the node-node adjacency matrix of a connected network. Similar to the assignment model, we use a tweaked median crossover MDX that has proven to be very good for network design problems [9]. We also use elitism across generations and a connectivity repair procedure for the mutation operator.

Since the constraints in DP are separable, we can re-write DP as

$$\min \left\{ C(Y) + F(Y) : \mathcal{N}V^k = b^k, \forall k, \sum_{k=1}^{n} V^k \leq n(n-1)Y, V^k \geq 0, Y_{kl} \in \{0,1\}, \forall k,l \right\},$$

where

$$F(Y) := \min \left\{ \text{trace} \left( FXD(Y)X^T \right) : Xe = e, X^Te = e, X_{il} \in \{0,1\}, \forall i,l \right\}.$$

Hence, the fitness/evaluation of a given chromosome $Y$ consists of the cluttering cost $C(Y)$ plus the click-count cost $F(Y)$ of the optimal assignment in the corresponding network. Because computing $F(Y)$ entails solving a QAP, we use a second GA (from Section 2.3) to find an approximate solution to the optimal assignment. Therefore, we apply a “master-subproblem” procedure in which a master GA for the design problem uses a QAP GA as a subroutine to estimate the fitness of each chromosome in a given population.

### 3.3 Numerical results

Due to the complexity in determining the optimal catalog topology and item assignment simultaneously, only the numerical results for the 12-node design problem are reported here. The population size and the number of generations are set at 20 and 100, respectively, for both the DP-GA and the QAP-GA. To be consistent, we populate the frequency matrix $F$ as described in Section 2.4.

<table>
<thead>
<tr>
<th>$\theta$</th>
<th>Cluttering cost $c = 1$</th>
<th></th>
<th>Cluttering cost $c = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>35.615</td>
<td>21.892</td>
<td>38.53%</td>
</tr>
<tr>
<td>0.9</td>
<td>27.464</td>
<td>15.055</td>
<td>45.18%</td>
</tr>
<tr>
<td>0.95</td>
<td>30.847</td>
<td>13.842</td>
<td>55.13%</td>
</tr>
</tbody>
</table>

Table 4: GA improvement in the objective value for the design problem.

Table 4 shows a significant difference in objective values between the initial and final GA populations. It can be inferred that catalog navigation could deteriorate severely by a poor choice on the catalog topology. Not surprisingly, the GA objective value increases as the experience index $\theta$ decreases or as the cluttering cost coefficient $c$ increases.
Figures 4(a) and 4(b) show the best catalog designs at two different levels of cluttering cost. When the penalty for page cluttering is less severe, having more hyperlinks is preferred, leading to the 15-link topology shown in Figure 4(a). Due to the nature of the frequency matrix $F$, it is clear that items are also perfectly clustered (i.e. items inside each cluster are directly connected). On the other hand, when $c$ becomes larger, Figure 4(b) shows that the catalog topology is reduced to a tree structure with a 27% reduction in the number of hyperlinks. Also, items are no longer perfectly clustered with large $c$, potentially due to the highly constrained nature of a tree topology. As for the computational time, unlike solving QAP for a 12-node problem that takes only a few seconds, it takes on average 30 minutes to solve the corresponding DP.

$$\begin{array}{|c|c|} \hline \theta & \text{Number of links} \\ \hline 0.05 & 11 \\ 0.1 & 14 \\ 0.5 & 13 \\ 0.9 & 11 \\ 0.95 & 11 \\ \hline \end{array}$$

Table 5: Topology complexity as $\theta$ varies.

Finally, Table 5 shows the complexity of the topology designs (for a fixed clustering cost of 1) as $\theta$ varies. Complexity is measured as the number of links. Notice that for extreme experience factors (close to 0 or to 1) the number of links is reduced. Intuitively, for
inexperienced users the greater the number of links the higher the probability of “getting lost” in the random walk, thus yielding higher expected click-counts. On the other hand, for extremely experienced users, having more links adds to the page clutter yet helps little to reduce the click count, thus the more simplistic interface.

4 Generic versus customized catalogs

As mentioned in Section 1.2, one of the key factors that affects catalog design is user experience. User experience affects the choice of distance matrix in the objective functions of both the assignment and design models. In some cases it is possible to compile statistical data to determine average click-counts between pages for a given catalog topology. Let \( \hat{D} \) denote the matrix of observed average click-counts, that is, \( \hat{D}_{ij} \) represents the average click-count from page \( i \) to page \( j \) when items shown on the two pages were purchased or evaluated together by multiple users. When the catalog topology is fixed, one approach is simply to incorporate the observed \( \hat{D} \) and \( F \) in solving QAP. Alternatively, an experience factor \( \theta^* \) can be found to minimize the difference between \( \hat{D} \) and \( D(\theta) \), and then solve QAP for \( \theta^* \). The advantage of deriving \( \theta^* \) is that the catalog can be optimized even if a minor topological change is made (e.g. changing from three links per page to four links per page). We call such an approach the “generic catalog approach.” The generic approach makes sense when the web site visitors are more or less homogeneous, thus using one catalog (with one “optimized” \( \theta \)) is sufficient.

Another approach when \( \hat{D} \) is unavailable, or the designer wishes to have more flexibility in dealing with different levels of user experience, is to parametrically solve QAP (or DP) for multiple experience factors \( \theta \) and then, dynamically choose which optimal catalog topology to use according to a visitor’s profile. We call such an approach the “customized catalog approach.” In this section we discuss the implementation details of the generic and customized catalogs.

4.1 Generic catalog approach

To determine a value of \( \theta \) such that the weighted matrix \( D(\theta) = \theta D^B + (1 - \theta) D^W \) is as close to the observed distance matrix \( \hat{D} \) as possible, we solve the following optimization problem

\[
(FT) \quad \min \left\{ \|D(\theta) - \hat{D}\| : 0 \leq \theta \leq 1 \right\}.
\]

In defining FT we use the Frobenius matrix norm. In other words, for a matrix \( M \) we have

\[
\|M\| := \text{trace}(M^T M)^{1/2}.
\]
Let $\theta^*$ denote the solution to FT. Since problem FT is a simple one-dimensional constrained quadratic problem, it is easy to find its solution explicitly. If

$$\lambda := \frac{\text{trace} \left( (D^B - D^W)^T (D^W - D^\hat{D}) \right)}{\|D^B - D^W\|^2} \in [0, 1],$$

then set $\theta^* = \lambda$. Otherwise set

$$\theta^* = \begin{cases} 0 & \text{if } \|D^W - D^\hat{D}\| \leq \|D^B - D^\hat{D}\|, \\ 1 & \text{if } \|D^W - D^\hat{D}\| > \|D^B - D^\hat{D}\|. \end{cases}$$

### 4.2 Customized catalog approach

In this approach we parametrically solve QAP (or DP) and determine a set of optimal catalog topologies for various values of $\theta$. Multiple versions of the catalog are kept and one of the versions will be used to interact with a visitor depending on the visitor’s profile. When a visitor visits the web site for the first time, an experience factor $\theta$ close to zero is assigned to the visitor and the topology based on the distance matrix $D(\theta)$ (closer to $D^W$) is used. The web site would keep track of the frequency of visits of the visitor. Once the number of rapid clicks (before reaching the next page of interest) decreases, the visitor’s experience factor $\theta$ has probably improved, thus a different topology (closer to $D^B$) could be used.

In particular it is interesting to study the shape of the objective function of QAP as a function of $\theta$. Let $z(\theta)$ denote the optimal objective value of QAP as a function of the experience factor $\theta$. Consider the set $Q$ representing the feasible region for QAP, that is,

$$Q := \{ X : Xe = e, X^Te = e, X_{il} \in \{0, 1\}, \forall i, l \}.$$ 

Notice that $|Q| = n!$. Let $X \in Q$ be fixed. Since $D^W \succeq D^B$, we have

$$\text{trace} \left( FX D^B X^T \right) - \text{trace} \left( FX D^W X^T \right) \leq 0,$$

and so, $\theta \left( \text{trace} \left( FX D^B X^T \right) - \text{trace} \left( FX D^W X^T \right) \right) + \text{trace} \left( FX D^B X^T \right)$ is a linear function on $\theta$ with negative slope, that is, it is linearly decreasing in $\theta$ for $X$ fixed.

Therefore, the function $z(\theta)$ is the minimum of $n!$ decreasing linear functions evaluated at $\theta$, so that $z(\theta)$ must be a piecewise linear concave decreasing function of $\theta$. As such, the lowest value of $z(\theta)$ is attained at $\theta = 1$, that is, when the user has perfect visibility (perhaps by memorizing) and uses the shortest path to search for the next item. On the other hand, when $\theta = 0$, $z(\theta)$ achieves its maximum value, corresponding to the totally random search for the next item. Moreover, the optimal assignment $X(\theta)^*$ remains constant inside the interval segments where $z(\theta)$ is linear with respect to $\theta$. In other words, the optimal solution is unstable only at the points where $z(\theta)$ changes its slope.
When the number of linear pieces in \( z(\theta) \) is small, the interval \([0, 1]\) can be partitioned into a small number of disjoint subintervals such that there is a unique optimal catalog topology associated with each subinterval. Hence, the web catalog designer only needs to keep a small number of catalog topologies to deal with a wide range of different types of user experience. Computational experience shows that when the frequency matrix is symmetric and homogeneous, then the number of linear pieces in \( z(\theta) \) is often small. Similar remarks apply to the design model.

<table>
<thead>
<tr>
<th>Item</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>10</td>
<td>16</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>100</td>
<td>0</td>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>C</td>
<td>100</td>
<td>200</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>D</td>
<td>500</td>
<td>500</td>
<td>200</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 6: Frequency matrix for a 4-node design problem.

For example, consider a 4-node problem with frequency matrix given in Table 6. Figure 5 shows the optimal objective value of problem DP as a function of \( \theta \). As mentioned before, the objective values determine a piecewise linear concave decreasing function with roughly three linear segments. For each segment there is a unique optimal topology and assignment as illustrated in Figure 6. Even though there are 912 possible solutions (4! permutations for each of the 38 possible connected topologies) only three are optimal and need to be

Figure 5: DP optimal objectives as \( \theta \) varies.
5 Concluding remarks

In this paper, we proposed two optimization models to improve web catalog navigation. When the catalog topology is highly structured, we show that catalog browsing can be improved by solving a quadratic assignment problem. On the other hand, if the topology is also subject to change, a two-stage optimization procedure is needed to identify improved page connectivity and item assignment. A recently proposed genetic algorithm that combines standard GA operations with local search is used as the optimization heuristic. By comparing to the optimal solution of small-size problems, we found that the GA was able to obtain quality solutions efficiently. When the problem size increases, the GA heuristic produced catalog designs that are up to 19% better than those produced by a polynomial-time IP heuristic with comparable computational times.

While our models exploit only the pair-wise correlation among items, they still are effective in capturing higher order correlations. For example, if three arbitrary items, A, B, and C, are popular and highly correlated, the frequency matrix will show high values between pairs (A, B), (A, C), (B, C), and so forth; which likely will result in all three items being clustered together in the optimal design.

Furthermore, although items in the same cluster might be highly correlated, the optimal topology is not necessarily a fully-connected network within the cluster because the cluttering costs outpace the benefit of click count reduction. For example, in the 12-node topology shown in Figure 4(a) there exists a star-topology within the cluster for items 9, 10, 11, and 12, with item 10 as the center of the star.

We have assumed that the frequency matrix is obtained from actual purchasing or browsing data. However, it is possible to rely on human expertise to populate the matrix, e.g. when launching a new e-commerce site. Similarly, during the period of product promotion and bundling the designer could use a “supplementary” frequency matrix to overlay
the frequency matrix resulting from actual visits, thus boosting correlations among targeted items.

As for future research, the current work focuses on improving web catalog navigation by optimizing the placement of hyperlinks. A potentially fruitful extension is to consider a combined design of hyperlinks and the category hierarchy (i.e., the side bar). The goal is to separate items into categories or hierarchy layers, and then to use our methodology to find the optimal design for each category or layer. Work is underway on this direction.

References


