韓國經營科學會誌
第33권 第4號
2008年 12月

Hop 제약조건이 고려된 최적화 웹정보검색*

이우기**† · 김기백*** · 이화기****

Optimized Structures with Hop Constraints for Web Information Retrieval*

Wookey Lee** · Kibaek Kim*** · HwaKi Lee****

■ Abstract ■

The explosively growing attractiveness of the Web is commencing significant demands for a structuring analysis on various web objects. The larger the substantial number of web objects are available, the more difficult for the clients (i.e. common web users and web robots) and the servers (i.e. Web search engine) to retrieve what they really want. We have in mind focusing on the structure of web objects by introducing optimization models for more convenient and effective information retrieval. For this purpose, we represent web objects and hyperlinks as a directed graph from which the optimal structures are derived in terms of rooted directed spanning trees and Top-k trees. Computational experiments are executed for synthetic data as well as for real web sites' domains so that the Lagrangian Relaxation approaches have exploited the Top-k trees and Hop constraint resolutions. In the experiments, our methods outperformed the conventional approaches so that the complex web graph can successfully be converted into optimal-structured ones within a reasonable amount of computation time.

Keyword : Web Structuring, Binary Integer Linear Programming Model, Top-K Retrieval, Hop Constraints, Lagrangian Relaxation

논문접수일 : 2007년 10월 06일 논문게재확정일 : 2008년 11월 13일
논문수정일(1차 : 2008년 06월 10일)
* 본 연구는 지식경제부 및 정보통신연구진흥원의 대학 IT연구센터 육성 · 지원사업(IITA-2008-C1090-0801-0031)의 연구결과로 수행되었음.
** 인하대학교 산업공학과 교수
*** Texas A&M 석사과정
**** 인하대학교 산업공학과 교수
† 교신저자

1. 서 론

웹(World Wide Web)은 정보이다. 따라서 물리적 전달방법보다 논리적 처리방법 즉, 필요와 공급에 대한 이해, 효율적인 저장과 검색방식, 자원제약하의 효과적 분배 등과 관련된 개념에서의 개선이 훨씬 중요하다. 현상적으로 웹은 전 세계적 규모의 다양한 정보에 대해 실시간적 접근이 가능케 해주는 유일한 정보원천으로서 자리 잡았다. 양적 질적 측면에서 웹의 폭발적 확장을 부인하는 사람은 없다. 양적으로는 숨은 웹(Hidden Web)[1]을 제외하고도 웹 페이지 수는 현재 조 단위를 넘어서 지수적으로 팽창하고 있으며[21, 28], 질적으로는 단순한 HTML이나 XML 이외에 RFID, UCC, Blog, DMB 및 웹 서비스 등 다종다양한 내용이 계속 웹에 합류하고 있다[1, 7, 19]. 한편으로 또 다른 점은 기하급수적으로 커지는 웹에서 '작은 세상(small world)' 효과를 보이며 소수의 웹 사이트로 집중현상이 나타나고 있다[31, 34, 35]. 이는 정보가 소수의 노드로 결집하면서, 정보검색은 양적폭발에 비해서는 상대적으로 난이도가 심화되지 않고 있다는 것을 의미한다. 그러나 실제 검색을 하기위해 이러한 작은 세상 내부를 대상으로 정보검색을 시도해보면 여전히 매우 비효과적인 것으로 드러났다[19, 27, 28, 36]. 즉, 내부 검색 혹은 인트라넷 검색 시 일반적인 검색엔진에서 효과를 보이는 페이지랭크 기법 등을 적용할 경우, 웹 사이트의 많은 웹 페이지 중에서도 각종 홈페이지들이 최상위에 나타나는데, 이들은 사실 내부 사용자들에게는 무의미한 검색결과인 것이다. 또 다른 문제로서 작은 세상 즉, 웹 사이트 내부에 깊이 들어있는 중요 페이지는 거의 검색엔진이 접근하지 않고 있는데[27, 35], 그 이유는 검색엔진 및 관련 웹 로봇들이 웹 페이지 사이의 수많은 사이클 구조, 웹 노드들의 중복성 검사 등을 위한 처리능력에 한계가 있어서 주로 대표 페이지 및 그의 갱신 업로드 기능에 초점을 두고 있기 때문이다.

웹의 효과적 · 효율적인 저장, 검색 및 분석 문제의 최일선에는 일반사용자 이외에도 웹 로봇이 있다. 웹로봇은 크롤러(crawler), 에이전트(agent) 혹은 스파이더(spider) 등으로 불리는데, 이는 컴퓨터 프로그램의 일종으로 웹 서버에 접근하여 서버에서 페이지들을 자동으로 다운로드하고, 변경정보를 갱신하는 기능을 수행한다. 즉, 검색엔진은 웹 로봇을 사용하여 인터넷 상의 웹 페이지들을 수집하고, 내용물을 전송하고, 이를 인덱싱하고 있는 것이다. 이것이 성공적으로 수행되면 웹 로봇은 로컬 인덱스를 빠르게 검색하여 검색에 맞는 가장 합당한 결과를 찾아서 데이터베이스에게 제공하고, 검색이란 이 결과를 사용자들이 재활용하는 것이다.

```
<?xml version = '1.0' encoding = 'UTF-8'?>
<urlset xmlns = "http://www.sitemaps.org/
                 schemas/sitemap/0.9"
        <url>
            <loc> http://www.wookey.lee </loc>
            <lastmod> 2008-09-21 </lastmod>
                <changefreq> daily </changefreq>
                <priority> 0.8 </priority>
        </url>
</urlset>
```

[그림 1] 사이트맵 태그구조

현재 검색로봇의 검색방식은 신규페이지의 검색과 기존의 선호페이지의 재방문에 있어서 일반사용자의 검색과 유사한 패턴을 보인다. 즉, 기존의 웹 페이지에 포함된 링크를 따라 가거나, 혹은 무작위로 방문(random jump)하는 두 가지 방식으로 새로운 웹페이지들을 찾아내고 있으며, 그 양자의 비율은 구글봇의 경우 85 : 15정도 된다[9, 15]. 탐색에 있어서 검색엔진이 가진 페이지 평가기준 즉, 가중치를 기준으로 우선순위에 따라 탐색(navigation)을 수행한다. 일반 사용자의 경우 선호도라는 가중치를 기준으로 탐색하는 것이다. 또한 검색엔진 등을 이용한 random jump도 유사하다. 웹 페이지의 재방문 혹은 갱신에 있어서는 사이트맵(sitemaps)으로 명

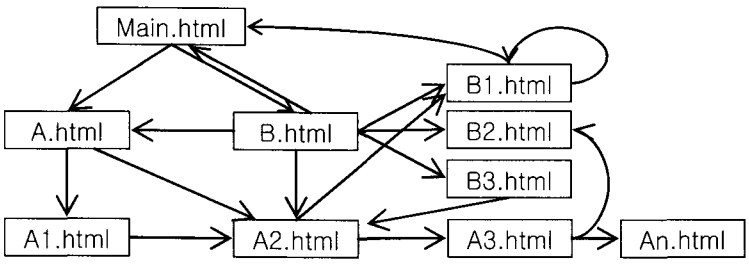
1) 웹 검색은 주로 표면적 웹(Surface Web)을 다루며, 숨은 웹은 스키마 매핑 등 데이터베이스 이슈이다.

명된 XML 파일을 통하여 페이지의 변경일자, 변화주기, 중요성 등의 메타정보를 수록하여 사이트 내 페이지 변화를 파악하는 방식이 적용되고 있다[2, 32]. 현재 각 검색엔진 및 쇼핑몰 등 웹 서비스 업체마다 개별적인 웹 로봇들을 도입하고 있으며 심지어 개인 웹 로봇들도 등장하고 있다. 그리고 이들은 웹 서버에 무단 체류하며 웹 서버 자원을 점유하고 해킹 등의 피해를 입히기도 하며 이는 크게 증가하는 추세이므로 이에 대한 대책으로서도 본 연구의 가치는 적지 아니하다 하겠다. 이는 웹 서버와 검색로봇 양자에 유익이 되기 때문이다. 우선 웹 서버의 경우 무단으로 퍼가는 웹 로봇들이 오랜 시간 웹서버를 점유하는 대신, 사전에 전체구조에서 관심 있는 정보만을 추출해서 가져가게 하거나, 갱신을 위한 재방문 시에는 변경된 정보만을 선별적으로 가져가게 할 경우 자원점유시간을 크게 절감할 수 있다. 이는 서버 증설의 부담을 덜어주고, 매출에 직접적인 영향을 줄 수 있는 일반사용자 서비스를 향상시킬 수 있는 효과가 있다. 웹로봇의 경우 적확한 정보선별이 가능해지므로 이러한 정보를 가지고 검색을 수행할 경우 하게 할 수 도 있다. 마찬가지로 일반 사용자에게는 기존의 정적이고 개별화되지 않은 사이트맵 대신 사용자의 선호도가 반영된 구조적 사이트맵을 제공할 수 있다. 본 연구에서는 이러한 사용자 및 웹로봇을 위한 정보검색 및 정보의 갱신방식에 최적화기법을 적용한다는 것이다.

예컨대, 사용자의 검색이 [그림 2]와 같은 웹 객체의 집합(이를 웹 사이트라고 인식해도 된다)에 대해 이루어진다고 할 경우, 검색대상이 되는 웹 사이트는 사각형과 화살표로 구성되는데 이는 각각 웹 객체와 하이퍼텍스트 링크를 의미한다.

근본적으로는 기존의 검색엔진 방식을 사용할 경우 검색의 결과는 웹 객체의 나열이 된다. [그림 2]와 같은 경우를 살펴보면 검색결과의 목록이 A2, Main, B1, A3, … 등과 같이 되며 이는 구글의 페이지랭크(PageRank)알고리즘[10, 15, 31]을 적용할 경우 각각 1.78, 1.40, 1.18, 1.01, … 등의 값을 가진다. 또한 A2, B2, B1, Main, … 의 순서로 나열될 수도 있는데 이는 HITS 알고리즘[23]에 따라 각각 2.10, 1.43, 1.43, 1.39, … 등의 권위도(Authority)값을 가지며, 그러한 평가 기준에 따라 검색순위가 매겨지게 된다.

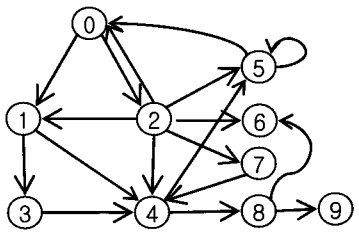
문제는 이러한 검색결과로서 표현되는 웹 객체들이 일차원적으로 나열된다는 점이다. 이는 웹 객체가 서로가 연결되어 있는 아크정보를 배제한 것이다. 즉, 웹이 가지고 있는 아크의 풍부한 표현능력을 제공하지 않으며, 그 하위에 어떤 웹 객체들이 연결되어 있는지를 배제한 결과이다. 그러므로 예컨대, 일련의 순서를 가진 웹 객체들([그림 1]에서는 A1, A2, …, An), 혹은 중요객체(예, authoritative pages)나 연결 객체(hub pages)들([그림 1]에서는 B, B1, B2, B3) 등이 모두 무시되고 1차원적으로 사용자에게 제공되고 있다. 그러므로 이러한 방식은 사용자에게 구조적 정보를 제공하지 못하며, 이로 인해 발생하는 문제는 다음과 같다. 사용자로 하여금 검색 중인 전체 웹 사이트에서 현재 자신이 어느 위치에 있는지, 얼마나 더 검색해야 하는지 단말노드(leaf node)인지 시작노드인지, 얼마나 연결된 자식노드가 있는지 등에 대한 정보가 없는 현상



[그림 2] 사례 웹 사이트

즉, 위치유실(lost in cyberspace)증세를 야기하기 쉽다. 또한 유사한 웹객체 주변을 계속 맴도는 사이클 즉, 방문경로의 비효율성을 낳을 가능성이 높다. 이로 인해 웹 로봇이 데이터를 가져올 때에 루트노드로부터 특정 깊이까지만 한정하는 것을 권고하거나[18], 이론전개에 약점을 노정하는 단말노드(dangling node)를 제외하기도 한다[11]. 이는 또한 웹 객체의 변경사항 및 인지향상에 취약하며, 특히 조작된 검색결과(Spamming)[15, 16, 21]에 취약하다. 즉, 조작된 다수의 웹 페이지에 대해 매번 검색하려고 시도할 경우 비효율이 야기되기 쉽다. 따라서 좀 더 이론적으로 강건한(robust) 모형 즉, 구조적 분석이 강력히 요구되는 것이다[15, 19, 28].

본 연구에서는 웹(WWW)을 다음과 같이 그래프 표현으로 정의한다. 즉, 웹은 유향 그래프(directed graph) $G = (N, E)$로 인식되며, 웹 노드의 집합 $N = \{0, 1, 2, \cdots, |N|\}$ 및 웹 아크는 아크 함수 $x_{ij} : N^k \rightarrow \{0, 1\}$ $\forall (i, j) \in E$ 유한 노드집합 N 및 유한 아크 집합 E 의 순서쌍으로 인식하여 (i, j)로 표현할 수 있고, $i, j = N$. 웹의 하위 집합으로 웹사이트를 보면 역시 노드와 아크로 이루어진 유향그래프로 간주할 수 있다[7, 33]. 이 때, 노드는 정보를 담고 있는 웹 페이지로, 아크는 웹 페이지 간의 하이퍼링크(hyperlink)로 볼 수 있다. 일반적으로 하나의 웹 사이트는 시작페이지를 가지고 있으며, 이 페이지를 통해 사이트 내 다른 웹 페이지로 이동할 수 있게 된다. 시작페이지로부터 특정 웹 페이지로의 이동 경로를 하나로 제한할 경우, 웹 사이트는 하나의 트리 구조로 인식될 수 있으며, 이 때 시작페이지는 루트노드가 된다.



[그림 3] 웹 유향 그래프

본 연구에서는 웹 사이트의 구조화에 대한 좀 더 실용적인 접근을 위하여, 기존의 웹 사이트의 구조화에 대해서 추가적으로 다음 세 가지 사항을 고려하였다.

- 첫째, 대상이 되는 웹은 표면적 웹(Surface Web)이다. 즉, 숨은 웹(관련 연구는 [2, 3, 10, 26] 참조)은 구조화의 대상에서 제외한다. 왜냐면, 숨은 웹은 실제 웹 페이지로 존재하는 것이 아니라, 내부의 데이터베이스에 존재하면서 사용자의 질의가 입력될 때 관련정보를 그 데이터베이스로부터 다만 인터페이스로써 형식만 웹을 통해 제공해주므로 구조화의 대상이 아니기 때문이다.
- 둘째, 최적화된 트리구조에서 노드의 깊이에 대한 조건인 Hop 제약조건을 고려하였다. 이 때 Hop이란 특정 노드로부터 하이퍼텍스트로 연결된 웹 페이지의 숫자를 의미한다. 큰 Hop은 제한 한다는 것은 이용자가 특정 깊이 이상으로는 확인하려고 하지 않는 현상을 반영한다는 의미이다. 예를 들어, 이용자는 처음의 결과 페이지에서 추가적으로 다른 정보를 찾기 위해 몇 번 이하로만 웹 페이지를 옮겨 다닐 것이라는 것이다. 그러므로 본 연구에서 그 특정 수 이하만큼의 깊이를 갖는 트리구조를 만들어야 한다는 접근법은 매우 현실적인 조건이 될 수 있다.
- 셋째, 이용자는 결과로부터 가장 좋은 몇 개의 페이지만을 확인할 것이라고 가정한다. 이는 현실적인 가정이다. 왜냐면 이용자들은 전체 웹페이지는 숫자가 방대하여 한 화면에 모두를 살펴볼 수 없다. 그러므로 웹 사이트의 모든 페이지를 구성하기 보다는 가장 좋은 몇 개의 페이지만을 구조화하는 것이 더 바람직하며, 이를 우리는 *Top-k* 페이지에 대한 구조화라고 부른다.

본 논문의 의의는 다음과 같다. 우선 최적화 기법

을 정보검색의 방법론으로 적용하였다는 점이다. 그 중에서도 네트워크 모델과 정수계획법을 이용하여 웹 정보의 최적 구조화를 시도하였다. 이는 웹페이지 및 하이퍼링크의 존재유무는 0/1으로 표현될 수 있고, 그리고 최적구조화의 다양한 장점들을 웹 검색에 최초로 적용한 의의가 있다. 둘째, 실제성을 높이기 위해, 웹 사용자(일반사용자 및 웹 로봇)가 특정 깊이 이상으로 접근 경로가 구성되지 않도록 제약을 하면서 이러한 웹의 최적구조를 유지한 상태에서 가장 좋은 k개의 페이지(*Top-k* page)에 대한 구조를 추가적으로 제시하였다. 따라서 웹 정보의 최적 구조화뿐만 아니라 실용적으로도 가장 좋은 k개의 페이지에 대하여 접근성을 높일 수 있도록 하는 것이다. 또한 이러한 정보는 메타정보 특히 사이트맵(Sitemap) 형식으로 사용자들에게 제공될 수 있다는 실용적 가능성이 있다. 그리고 특히 최적화 접근법의 다양한 강점이 웹 정보검색에 적용될 경우 웹의 변화에 대해 감도분석 등의 효과적 적용이 가능하다는 것을 보인다. 그리고 이 과정에서 우리는 이전 연구에서 노드들 간의 회로(circuit)를 제거하기 위해 추가했던 제약조건의 수를 흐름보존(flow conservation) 제약조건을 추가하여 줄임으로써 좀 더 현실적 모형을 만들 수 있음을 입증하고, 더욱이 라그랑지안 완화법(Lagrangian Relaxation)을 이용하여 제약조건을 완화한 해를 구하여 최적해의 유지와 제약식의 완화효과에 대한 비용분석을 하였다는 것이다. 또한 이러한 내용을 실제 구현한 사례로써 그 효용성을 입증하였다.

논문의 구성은 다음과 같다. 제 2장에서는 관련 연구에 대해 살펴보고, 제 3장에서는 웹 사이트 구조화의 문제를 네트워크 모델과 정수계획법을 이용하여 모델링하는 과정에 대해 설명한다. 제 4장에서는 라그랑지안 완화법을 이용하여 제약조건을 완화시켜 해를 구하는 과정을 설명하고, 제 5장에서는 예제를 통해 그 해가 계산비용을 낮춰준다는 것을 보여주고, 시스템구현 내역을 통해 실제 웹 검색에 어떤 의미를 가지는지를 보여주며, 마지막으로 제 6장에서는 결론 및 향후 연구방향에 대해 언급한다.

2. 관련 연구

웹의 구조화에 대한 연구를 그래프로서의 웹(Web-as-a-graph)접근법이라 부르는데, 이는 먼저 웹 전체의 크기를 추정하는데 도입되었으며[6, 31], 그 크기는 지수법칙(power-law)[34], 오각형 모형, Scale-free모형 등을 따른다고 제안되었다[23]. 전체 크기에 대한 추정과 별개로 웹 그래프 자체는 구조가 지나치게 복잡하여 웹 서버나 사용자 즉, 개별사용자 및 웹 로봇의 탐색에 적용하기 곤란하다. 그러한 복잡성 때문에 심지어 넓이우선탐색을 권장하고 그 결과로써 전체 웹 사이트에서 특정깊이 이상 탐색하지 않아도 무방하다는 제안[27]을 하는 등 근본적 한계를 노정하고 있다. 그러나 이러한 조야한 구조화 즉, 넓이우선 혹은 깊이우선 트리로의 구조화는 웹의 경우에는 상기 양자 모두 구조화로는 부적합한데, 전자의 경우 노드 및 링크의 숫자가 과다하므로 루트노드 주변에 대부분의 웹 페이지가 붙게 되는 경우가 많고, 후자의 경우 매우 긴 혹은 전체 웹페이지가 한 줄로 늘어서는 모양이 되는데, 이는 그 만큼의 마우스클릭 즉, 검색시간이 과다하게 되므로 이러한 형태는 구조화라는 의미가 없다.

웹의 구조화에 있어서 필수적인 요소가 아크의 가중치이다. 예컨대 어떤 계층적 구조이든, 심지어 완전그래프 일지라도 노드의 가중치 합은 동일하다는 등의 문제가 생긴다. 즉, 구조화의 결과물에 대한 평가 및 우열검증이 곤란해진다. 주지할 사실은 기존의 검색엔진들의 가중치는 노드 즉, 페이지의 랭크이지 아크의 가중치는 아니다[1, 16, 17]. 그 이유는 물론 검색결과물로 개별 페이지가 요구되기 때문이다. 그러나 본 연구에서와 같이 웹 검색 및 구조화라는 관점에서는 검색의 결과가 특정 한 페이지일 수도 있지만, 일련의 웹페이지들이 대상이 된다고 보는 좀 더 일반화된 시각이므로 상기 아크 가중치에 대한 분석은 불가결하다.

아크에 대한 분석으로 우선 거론될 수 있는 연구는 HITS[29]이다. 이는 웹에서 일련의 권위 페이지들(authorities)과 연결고리(hubs) 페이지들을 효과

적으로 찾는 알고리즘이다. 권위 웹 페이지는 질의하는 주제에 대해 좋은 내용을 담고 있는 페이지들을 의미하며, 좋은 연결고리 페이지들로부터 링크가 많이 걸리게 된다. 역으로 우수한 권위 페이지들과 링크를 많이 가진 페이지들이 또한 좋은 연결고리 페이지들이 된다. 이는 사회연결망(Social Network) 혹은 공동체망(community networks)관련연구의 기초가 되었으며[18, 24, 28], 일명 상호강화 관계성(mutually reinforcing relationships)라고 부르는 요소의 추출에 활용되고[18], 또한 기술망(technical networks), 혹은 논문의 인용정도(scientific and patent citations) 등에 사용되고 있다. 그러나 원래의 논문자체에서 사용한 정통적 문제 즉, 웹 정보의 검색에는 HITS가 뜻밖에도 적용되지 않고 있다. 그 이유는 첫째, 동일한 노드를 권위 페이지와 연결고리 페이지로 분리해석한 점에서 비현실성이 노정된 것과 둘째, 전이행렬(transition matrix)의 특성에 있어서 복수 최대 특성값(Dominant Eigenvalues)의 경우 등 알고리즘의 수렴성 및 안정성에 문제가 있기 때문이다.

한편 페이지랭크(PageRank) 알고리즘[21, 31]은 상기 약점들이 제거되어 현재 정보검색 및 검색엔진에 있어서 가장 폭넓게 적용되고 있다. 그러나 페이지랭크 역시 나가는 아크의 가중치는 모두 동일하다는 가정이 비현실적이다. 따라서 받는 노드와의 관련성에 따라 차등되는 아크 페이지랭크로의 변형이 필요하다. 따라서 본 연구에서는 실험부분에서는 페이지랭크 알고리즘뿐만 아니라 이를 보완한 가중치를 적용해보았다.

현재 웹에서 광범위하게 무시되고 있는 특성의 하나는 계층적 구조이다. 이는 도메인 이름체계(Domain Name System)와 웹 페이지들이 각 웹 사이트 내부에서 물리적으로 존재하는 방식이 루트노드로부터 개별 디렉터리 그리고 하위 디렉터리들 그리고 개별웹 페이지들로 만들어지는 이러한 구조가 웹 검색에 직접적으로 적용된 경우가 적다[36]. *Eiron* et al.[11]의 경우 웹 그래프 모델에서 링크구조를 통합하여 명백히 계층적 구조로 변형하는 방법을 제안하였고, 그 연구는 해당 조직의 구성과 관련 웹 사이트의 구조를 결합하는 방식이다.

한편 페이지 방문횟수(Page Popularity) 즉, 웹 로그 정보에 따라 웹 페이지의 배치를 재구성하는 시도도 있으나[13, 17], 이는 변화가 많은 웹의 환경에서는 구조가 지나치게 자주 바뀌거나 단기적인 방문기록 혹은 악의적인 방문공격(DoS : Denial of Service)에 취약할 우려가 있다. 본 연구에서는 웹 로그를 사용하지 않고 웹 페이지 및 웹 아크에 대한 전통적인 가중치들을 사용하여 문제의 일반성을 해치지 않도록 하였다.

그 밖에도 [20, 21] 및 [7]의 일반화된 웹 로봇을 포함한 웹 검색 방식에 대한 연구들도 연관성이 있다. 특히 [20]의 경우 웹 로봇의 검색 스케줄 정책에 대해 연구하였는데, 그들은 약 18만 개의 웹 페이지에 대해 다양한 웹 로봇 전략 즉, 넓이우선 정책, 백링크 숫자우선, 페이지랭크 우선 등을 적용하였다. 그 결과 높은 페이지랭크 우선 전략 다음으로 넓이우선 정책이 좋은 내용을 보였다는 점을 확인한 것은 의미 있는 발견이었으나, 단 하나의 도메인에 적용한 결과로 일반화하기는 어렵다. 다음으로 Najork와 Wiener의 경우[27] 3억 개 이상의 페이지에 대해 넓이우선 전략을 적용하였고, 결과적으로 의미 있는 페이지들을 가지고 오는 효과를 입증하였지만, 다른 전략들과 비교하지는 않았으며, 깊은데 위치한 페이지들의 중요성은 확인하지 못한 약점이 있다. Abiteboul 등의 연구[1]에서는 OPIC라는 알고리즘을 제시하여 각 페이지가 초기 '금액'을 각 노드에 한 번에 나누어주는 방식으로 가중치를 구했다. 이는 페이지랭크와 비슷한데 한 번의 연산으로 계산하기 때문에 아주 효율적으로 가중치를 구할 수 있으며, 10만 개의 페이지에 대해 수행해보았다. 그러나 이 역시 다른 전략들과 비교하지 않았다. 본 연구에서는 다른 전략들과 비교하였으며, 더 나아가 제안하는 웹 구조화 정보 즉, 메타정보는 일반사용자 및 웹 로봇에게 사이트맵으로 제공될 수 있다는 장점이 있다.

다음으로 웹의 구조화를 위한 노력은 웹사이트

설계에도 적용되고 있다. 사용자의 접근 패턴을 분석하여 사이트 구성과 인터페이스를 자동화된 방법으로 개선하는 적응적 웹 사이트를 제안한 접근법도 있다[32]. 이들은 사용자와의 상호작용에 근거하여 사이트 구조를 개선시키는 것을 목표로, 기존 구조를 건드리지 않고 링크를 추가하되, 기존 링크는 제거하지 않는 비파괴적인 방식의 변형을 시도하였다. 이 경우 알고리즘의 성능 분석에 그치고, 클러스터링 등의 효과가 제시되지 않는 단점이 있다. Botafodo et al.[4]의 연구에서는 웹 사이트를 설계할 당시 설계자가 의도했던 계층 구조를 찾아내고, 하이퍼텍스트 구조의 또 다른 특징을 설계자에게 보여 줌으로써, 사용자 인터페이스와 웹 사이트 구조를 개선하고자 하였다. Fu 등의 연구[12]는 페이지 클러스터를 생성하지 않고, 사용 패턴의 진화에 따라 웹사이트 구조와 구성이 변화하도록 허용하였다. 제안한 방법론의 유효성을 증명하기 위하여, 정확하게 분류된 페이지 비율로서 정확도 측정을 하였다. 이러한 접근법은 다소 지나친 재구성화 방식으로서, 실제로 서버 입장에서 볼 때 실용적인 방법론이라고 볼 수 없으며, 사용자 입장은 더욱 고려되지 않았다. 웹사이트 및 페이지 분석과 관련해서는 공영기관인 W3C에서 수행하는 유효성검사(validator.w3c.org) 및 링크구조분석 서비스가 있는데[39, 40] 이는 태그의 중첩성 분석 등에 지나지 않는다.

적응적 웹사이트 분석을 위해 선형계획법을 적용한 연구[24]가 있는데, 여기서는 목적 함수로 모든 링크들의 빈도수합계를 최대화하는 것을 설정하였다. 링크 빈도수는 동시 발생 빈도수로 표현된다. 제약 조건으로는 노드의 외부 링크수(정보 부담)와 홈페이지로부터 각 페이지로의 최단 경로 길이 즉, 탐색 길이에 대한 제한을 두었다. 그리고 성능 효율을 개선하고자 2국면 휴리스틱을 개발하였다. 활용도가 낮은 링크 제거를 통한 웹 사이트 재구조화를 수행하였고, 적응적 웹 사이트는 사용자 특성에 따라 개별적으로 혹은 전체적으로 적응하는 웹사이트를 구성하자는 것이다. 이 연구에서의 큰 약점은 사용자 방문빈도를 "선형 독립"으로 적용한데 있다. 예컨대, 웹 페이지'7'에 대한 방문 경로가 0 → 2 → 7 일 경우, 2 → 7의 방문빈도(Hit ratio) 혹은 머문 시간(duration time)은 그에 선행하는 0 → 2에 독립적이라고 보았다. 그리고 이를 바탕으로 0-1정수 선형계획법을 적용한 것은 당연히 부적절하다고 볼 수 밖에 없다. 따라서 본 연구에서는 웹 로그 등에서 얻어지는 사용자 방문빈도로 접근하지 않고, 일반적인 키워드 정보를 바탕으로 수행했다. Lee[36] 및 Lee and Lim[37] 연구는 선형계획법을 정보검색에 적용한 최초의 시도였다. 그러나 여기서는 일반적 최적트리를 적용한 의의는 있으나, Hop제약조건이나 Top-k 등에 대해 고려하지 않았다. 이점에서 홉 제약이 있는 최소합 아보레슨스(HCMA : Hop Constrained Min-Sum Arborescence) 접근법은 주목할 만하다. 원래 이 문제는 네트워크 설계와 라우팅, 스케쥴링과 같은 분야에서 자주 다루어지는 것으로, 다양한 접근법이 있으나 웹을 대상으로 한 연구는 없었다. 기술적 관점에서 Gouveia[14]는 HCMA 문제를 정수계획법 문제로 모델링하고 이를 해결하기 위해서 라그랑지안 기반의 발견적 방법을 제안하였다. Kawatra[22]의 연구에서는 HCMA 문제를 풀기 위해 라그랑지안 완화법과 서브 그라디언트 최적화(Sub-gradient Optimization), 가지교환 휴리스틱(Branch Exchange Heuristic)을 이용하는 방법을 제안하였다. 이들 역시 웹 환경에 적용된 적이 없으며, 또한 아크나 페이지가 변화되는 일반적인 웹 검색 및 웹 구조화와는 거리가 멀다.

3. 최적화 모델

3.1 가중치의 계산

가중치(Weight)란 웹 객체 즉, 사용자의 질의(query) 및 그 대상이 되는 웹 페이지 혹은 아크 이들 사이의 유사도(Similarity) 혹은 거리(Distance)를 의미한다. 가중치는 글로벌 가중치와 로컬 가중치로 분류될 수 있는데, 이들은 상호 독립적이다.

즉, 로컬가중치는 하이퍼링크를 배제하고 키워드 및 질의어를 사용하여 가중치를 구하며, 한편 글로벌가중치는 개별적 검색 키워드를 고려하지 않고 하이퍼링크 정보만으로 구하기 때문이다. 현재 많이 쓰이는 글로벌가중치 방식에는 페이지랭크 방법[7, 9, 15, 31], HITS방법[23]등이 있다. 로컬가중치로서 전통적인 정보검색에서 사용하는 기법들로서 벡터공간모델(VSM : vector space model)에 기반하여 *tf-idf*(term frequency and inverse document frequency)[6], Cosine 가중치[16, 37], 2-Poisson Model[33], 다양한 통계모델[9, 28] 등이 있다.

글로벌 가중치 즉, 아크정보를 활용한 아크기반 노드가중치 평가방법이 최근 구글의 성공과 함께 주목받고 있다[2, 9, 11, 35]. 이 방법에서는 인접행렬(adjacency matrix) M에 대해 정의하기를 만일 아크 i에서 j로의 아크가 존재하면 $M[i, j] = 1$이고, 없으면 0으로 나타낸다. 이러한 인접행렬에 기초하여 여러 가지 가중치 계산 기법들이 개발되고 있다. 페이지랭크 기법도 그 중 하나이다. 페이지랭크의 가정은 그래프의 강연결(SCC : strongly coupled component)요소를 대상으로 하고 인접행렬이 ergodic[9, 17]하도록 일종의 감쇠요소(a damping factor)를 인접행렬에 추가해준다. ergodic이란 인접행렬 M이 마코프 전이될 때 비음(non-negative) 값으로 구성되고 비가역(irreducible)한 기본행렬변환(primitive transition) 및 비주기적(aperiodic) 행렬변환이 일어난다는 특성을 의미하고, 그 역도 성립한다고 증명되어있다[17]. 이러한 성질을 만족한다는 가정 위에 만들어진 페이지랭크는 구글의 핵심 알고리즘으로서 그래프의 아크 즉, 하이퍼링크의 숫자에 기반을 두어 계산된다.

페이지랭크 알고리즘에서 노드 i의 중요도(rank)는 앞서 언급한 인접행렬의 한 요소로서 그러한 요소로 구성된 행렬식 즉, 인접행렬에 대해 행렬전환을 해당노드의 노드의 중요도는 노드 i로부터의 출력아크의 수로 나눈 값을 주어 반복적(iterative)으로 수행하면서 이미 증명된 바와 같이 일정 횟수이상 반복되면 수렴하는 것이 증명되었으므로, 수렴되는 바로 그 해당 값들을 각 노드(웹페이지)에 대한 중요도 즉, 페이지랭크로 사용하고 있다[31].

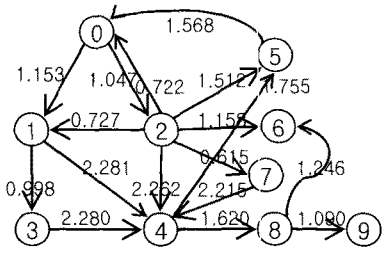
한편 로컬가중치(local weight) 즉, 웹페이지가 포함하고 있는 내용에 대한 가중치 평가방식 역시 필수적이다. 이는 전통적 정보검색에서 사용하는 벡터공간모델에 기반하고 있고, 그 중에서도 가장 많이 사용하는 방법의 하나가 *tf-idf*이다[6, 37]. 기본적으로 VSM에서는 사용자의 질의와 웹객체가 모두 키워드의 벡터로 표현된다. 만일 m개의 키워드 정보가 구해질 수 있다면, 노드 i는 m-차원의 정규화 벡터가 된다. 각각 키워드의 빈도(*tf*)요소와 역빈도(*idf*) 요소의 곱으로 표현하는 것이 *tf-idf*방법이다. 사용자의 질의 역시 m개의 벡터 $Q = <q_1, q_2, \cdots, q_m>$로 표현되고, 이때 만일 사용자 질의에 k번째 키워드가 포함되면 $q_k = 0$이고 아니면 $q_k = 0$이 된다. 주의할 부분은 이 방식에서는 아크 정보가 무시된다는 점이다.

그러므로 다양한 글로벌 및 로컬가중치의 조합이 가능하다. 본 연구에서는 기존의 페이지랭크 글로벌가중치를 적용해 우선 해당 노드를 선별한 다음, 개별아크에 다음 식 (0)과 같은 방식으로 노드간의 가중치를 계산하였다. 우선, m개의 특정 키워드가 웹페이지의 내용을 확인하고 사용자의 검색 질의어를 나타낸다고 가정하자. 이 때, 사용자의 검색 질의어가 $\delta(\leq m)$개의 키워드를 포함한다면, 두 노드 i, j 사이의 가중치는 다음과 같이 계산된다.

$$w_{ij} = \frac{D(i)^T \cdot D(j)}{\| D(i) \| \| D(j) \|} \qquad (0)$$

위 식에서, $D(i)$와 $D(j)$는 사용자의 검색 질의어에 포함된 각각의 키워드의 개수를 의미하는 한정벡터이다. 우리는 사용자의 검색 질의어에 포함된 키워드에 대응되는 요소만을 포함하는 모든 두 노드간의 아크에 대해 한정벡터를 구하였다. 즉, $|\delta|$는 한정벡터의 크기이고, $\| \cdot \|$는 위 식에서 벡터의 놈(Norm)이다. 가중치는 구조화된 검색 도메인에 의존적이다. 다시 말하면, 방향이 있는 아크의 가중

치는 구조화된 검색 도메인에 따라 다른 값을 가질 수 있다. 위 식에 근거하여 아크에 대한 가중치를 구하면 [그림 2] 및 [그림 3]의 예제구조는 [그림 4]와 같이 표현될 수 있다.



[그림 4] 가중치가 반영된 그래프

3.2 용어설명

위와 같은 아크가중치 정의를 이용하여 웹 사이트 구조화 문제를 같이 모델링하고자 한다. 우선 모델링에 사용된 용어 및 변수는 다음과 같다.

용어

N : 종점페이지의 집합

ω_{ij} : 아크(i, j)의 가중치, 식 (0) 참조.

h^t : 시작페이지와 종점페이지(t) 사이의 가능한 최대 아크의 수

결정 변수

X_{ij} : 최적해에 존재하는 경로에 페이지 i, j 간의 아크가 존재하면 1, 아니면 0을 갖는 이진변수

Y_{ij}^t : 시작페이지부터 종점페이지 t로 가는 경로에 페이지 i, j 간의 아크가 존재하면 1, 아니면 0을 갖는 이진변수

V_{ij} : 가장 좋은 k개의 페이지로의 연결이 존재하면 1, 아니면 0을 갖는 이진변수

3.3 최적 구조화 모델링

$$Z = max\left\{\sum_{i=1}^{N}\sum_{j=2}^{N} w_{ij}(X_{ij}+V_{ij})\right\}$$

subject to

$$\sum_{i=1}^{N} X_{i1} = 0 \tag{1}$$

$$\sum_{i=1}^{N} X_{ij} = 1, \quad for\ j \neq 1\ and\ \forall j \tag{2}$$

$$\sum_{i=1}^{N}\sum_{j=2}^{N} Y_{ij}^t \leq h^t \quad \forall t \tag{3}$$

$$\sum_{j=2}^{N} Y_{ij}^t - \sum_{j=1}^{N} Y_{ji}^t = \begin{cases} +1 & if\ i=1 \\ -1 & if\ i=t \\ 0 & otherwise \end{cases} \tag{4}$$

$$Y_{ij}^t \leq X_{ij} \quad \forall i, j, t \tag{5}$$

$$V_{it} \leq \sum_{j=2}^{N} Y_{ij}^t \quad \forall i, t \tag{6}$$

$$\sum_{i=1}^{N}\sum_{j=2}^{N} V_{ij} \leq k \tag{7}$$

$$\sum_{i=1}^{N} V_{ij} \geq V_{jk} \quad \text{for } 2 \leq j \leq |N|,\ 3 \leq k \leq |N| \tag{8}$$

$$\sum_{i=1}^{N} V_{ij} \leq 1, \quad \text{for } j \neq 1 \tag{9}$$

$$X_{ij} \in \{0, 1\} \quad \forall i, j \tag{10}$$

$$Y_{ij}^t \in \{0, 1\} \quad \forall i, j, t \tag{11}$$

$$V_{ij} \in \{0, 1\} \quad \forall i, j \tag{12}$$

제약조건 하에서 가중치를 최대로 하는 웹페이지의 최적구조화와 그 구조 안에서 가장 좋은 k개의 페이지로의 경로를 구하기 위해 우리는 정수계획법 문제로 모델링하였다. 이때, 목적함수 Z는 각 페이지 간의 아크와 가장 좋은 k개의 페이지로의 아크 간 가중치에 대해서 최대화하는 식이다. 식 (1)은 시작페이지로의 입력은 없어야 한다는 것이다. 즉, 시작페이지의 부모를 용납하지 않는다. 식 (2)는 트리를 위한 조건식이다. 즉, 부모페이지가 여러 개가 있을 때 그중 하나만 선택되어야 트리가 만들어 진다는 것이다. 식 (3)은 시작 페이지로부터 종점페이지까지의 아크의 개수가 미리 정의된 수(h^t) 이하이어야 함을 나타낸다. 종점페이지의 중요도에 따라 이 값은 미리 결정되어야 한다. 식 (2)와 식 (4)는 흐름보존 제약조건이며 다음절에서 자세히 설명된다. 식 (5)은 최적해 내에 아크(i, j)가 존재할 때만 시작 페이지로부터 종점페이지 t로의 경로에 아크 (i, j)가 존재할 수 있음을 의미한다. 식 (6)~식 (9)

는 가장 좋은 k개의 페이지에 대한 제약조건들이다. 식 (6)은 페이지 i로부터 종점페이지 t까지의 경로가 존재할 때만 아크(i, t)가 존재할 수 있음을 의미한다. 식 (7)은 가장 좋은 웹 페이지는 미리 정의된 k개 이하여야 함을 나타낸다. 식 (8)은 시작페이지를 제외한 가장 좋은 k개의 페이지를 위한 모든 경로는 부모페이지가 있어야 함을 의미한다. 식 (9)는 가장 좋은 k개의 페이지를 위한 경로들은 1개 이하의 부모페이지만을 가질 수 있음을 의미한다. 식 (10)~식 (12)은 이진 정수계획법이라는 의미이다.

3.4 회로제거에 대한 제약조건

복잡한 웹 사이트 구조로부터 트리형식의 최적화된 웹 사이트 구조를 구하는데 있어서, 웹 사이트 내부적으로 갖는 아크의 회로들은 항상 해결하기 어려운 문제였다. 그러나 본 연구에서는 웹 사이트 내의 회로들의 제거에 대한 제약조건을 흐름보존 제약조건을 추가함으로써 간단히 제거할 수 있었다. 또한, 이는 기존연구[22, 24]에서의 제약조건에 비교하여 보았을 때, 보다 적은 수의 제약조건 만으로 회로를 제거할 수 있었다. 이를 다음과 같이 증명하였다.

명제 1 : 흐름보존(Flow Conservation) 제약조건이 회로에 대한 제약을 한다.

증명 1 : 회로내의 페이지들은 자기 자신을 종점페이지로 갖는 아크를 갖는다. 즉, $i=t$라고 할 때, $Y_{ij}^t=1$을 갖게 된다. 이는 다음과 같은 제약조건들에 의해 제거할 수 있다.

$$\sum_{j=2}^{N} Y_{ij}^t = 0, \quad \forall i = t$$

이 식은 종점페이지에서는 어느 페이지로도 아크가 생성되지 않아야 한다는 것을 의미한다. 또한 다음의 식으로 종점페이지로는 어느 한 페이지에서 아크가 생성되어야 함을 제약할 수 있다.

$$\sum_{j=1}^{N} Y_{ji}^t = 1, \quad \forall i = t$$

위 두 식을 다음과 같이 하나로 표현하였다.

$$\sum_{j=2}^{N} Y_{ij}^t - \sum_{j=1}^{N} Y_{ji}^t = -1, \quad \text{for } i = t$$

즉, 종점페이지에서는 어느 페이지로도 아크가 생성되지 않으므로 회로내의 페이지들이 갖는 조건을 만족시키지 않는다.

명제 2 : 흐름보존 제약조건은 회로 제거에 대한 제약조건의 개수를 줄인다.

증명 2 : 기존의 제약조건[37],

$$X_{i j_1} + \sum_{k=1}^{m-1} X_{j_k j_{k+1}} + X_{j_m i} \le m, \quad \text{for } 2 \le m \le N$$

은 n = N개의 웹페이지에 대하여,

$\sum_{k=1}^{n} \frac{n!}{(n-k)!}$개의 제약조건을 생성한다. 예를 들어, 10개의 페이지에 대해서 위의 제약조건은 최대 6,235,300개의 제약조건을 생성한다. 반면, 여기서 제안하는 흐름보존 제약조건,

$$\sum_{j=2}^{N} Y_{ij}^t - \sum_{j=1}^{N} Y_{ji}^t = \begin{cases} +1 & if\ i=1 \\ -1 & if\ i=t \\ 0 & otherwise \end{cases}$$

은 N개의 페이지에 대하여, N^2개의 제약조건을 생성한다. 즉, 변수에 따른 다항(Polynomial) 개수의 제약조건의 생성을 보장한다.

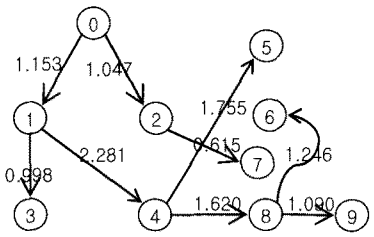
3.5 Top-k개의 페이지를 위한 제약조건

가장 좋은 k개(*Top*-k)의 페이지란 이용자가 원하는 가장 높은 가중치를 가지는 k개의 페이지를 말한다. 우리는 가중치를 이용하여 얻어진 웹사이트의 최적화된 구조를 유지하면서 가장 좋은 k개의 페이지와 그에 따른 구조를 추가로 구할 수 있도록 몇 가지 제약조건을 추가하였다. 웹 사이트의 구조

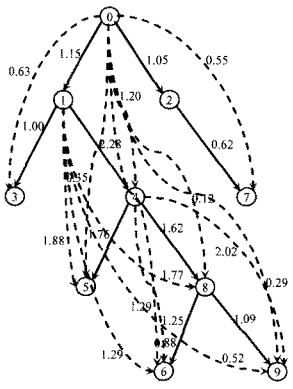
를 유지하면서 가장 좋은 k개의 페이지를 구하기 위해서는 다음과 같은 몇 가지 조건을 만족시키도록 가정하였다.

1. 가장 좋은 k개의 페이지들 간의 아크는 시작 페이지 혹은 같은 가지 내의 부모에서 가장 좋은 k개의 페이지만 가능하다.
2. 아크를 구성하는데 있어서 시작페이지보다 같은 가지 내의 부모에서 가장 좋은 k개의 페이지가 우선한다.
3. 페이지간의 아크 가중치가 가장 좋은 k개의 페이지를 만족시키지 않는 경우 다른 아크를 선택하지 않는다.

이 문제를 정수계획법으로 표현하기 위해 우리는 웹 사이트의 트리구조에서 각각의 가지 내에 있는 노드에서 상위의 모든 부모페이지로의 가상아크를 고려하여 시작페이지로부터 가장 좋은 k개를 찾도록 하였다.

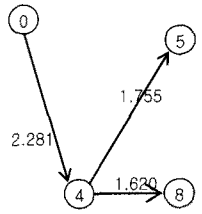


[그림 5] 예제에 따른 최적트리



[그림 6] Top-k트리를 위한 가상아크

[그림 5]에서는 최적화된 트리구조를 나타냈고, [그림 6]에서는 최적트리구조에서 동일 가지 내에서 상위의 모든 부모 페이지로의 가상아크를 나타내었다. 이와 같이 모든 페이지에 관한 가상아크 중에서 가장 좋은 k개의 페이지를 찾아 트리를 재구성한다. [그림 7]이 [그림 6]의 트리구조에서 루트노드에 연결된 가장 좋은 3개 즉, Top-3 페이지에 대한 결과를 나타낸다. 이 같이 k개의 가장 좋은 페이지에 대해 아크를 구하기 위한 제약조건이 모델의 (6)~(9)와 (12) 제약조건이다. 또한, 해 V_{ij}는 [그림 6]에서와 같은 가상아크를 나타내준다.



[그림 7] Top-3 트리

〈표 1〉 목적함수에 대한 감도분석 결과

Variable name	Reduced cost	Objective sensitivity ranges		
		Lower bound	Current	Upper bound
X_{01}	0	0.73	1.15	+ ∞
X_{02}	1.05	0	1.05	+ ∞
X_{13}	1	0	1	+ ∞
X_{14}	0	2.26	2.26	+ ∞
X_{20}	- 0.85	- ∞	0.72	1.57
X_{21}	- 0.42	- ∞	0.73	1.15
X_{24}	0	- ∞	2.26	2.26
X_{25}	0	- ∞	1.51	1.76
X_{26}	1.16	0	1.16	+ ∞
X_{27}	0.62	0	0.62	+ ∞
X_{34}	- 0.06	- ∞	2.2	2.26
X_{45}	0.25	1.51	1.76	+ ∞
X_{48}	1.62	0	1.62	+ ∞
X_{50}	0	0.72	1.57	+ ∞
X_{74}	- 0.04	- ∞	2.22	2.26
X_{86}	0.997	0	0.997	+ ∞
X_{89}	1	0	1	+ ∞

3.6 감도분석을 통한 최적구조화의 활용

최적화 접근법의 장점의 하나는 감도분석과 같은 잘 입증된 다양한 분석방법론을 적용할 수 있다는 점이다. 예컨대, 위의 예제에서 얻는 최적해에 대해 목적함수에 대한 감도분석을 수행한 결과가 표에 제시되어 있다. 우선 테이블의 첫 번째 행에서 알 수 있는 바는, 목적함수의 값 즉 w_{01}의 어떠한 변화에 대해서도 만일 그 변화의 범위가 0.73이상이기만 하면 최적해의 변경이 불필요하다는 점이다. 또한 다음 변수들의 변화 즉, $0 < w_{02} < +\infty$, $2.26 < w_{14} < +\infty$, $-\infty < w_{34} < 2.26$ 등과 같다는 것을 위의 결과에서 알 수 있다. 이것은 다음과 같이 활용될 수 있다. (1) 최적해가 변하지 않는다는 것은 웹 검색에 있어서 서버입장에서는 재계산이 불필요하다는 점이다. 웹은 변화가 많은 환경인데, 약간의 웹 페이지 변화에 대해 일일이 전체 구조를 매번 재계산을 하는 등의 수고를 덜어준다는 효과가 있다. 그리고 (2) 경우에 따라서는 최적해의 범위를 넘어서는 변화에 대해서도 다음으로 포함될 해가 무엇인지 예측할 수 있게 해준다. 예컨대, $X_{14} < 2.26$이 되는 경우 다음 해는 X_{24}가 되는 것이다. 이 점 역시 웹 서버의 관점에서는 재계산의 필요가 없어지므로 수행도에 있어서 매우 큰 혜택이 될 수 있다.

4. 라그랑지안 완화법

본 연구에서 최적구조화를 더 전개하여 라그랑지안 완화법을 적용하려는 것은 다음과 같은 실제적인 이유 때문이다. 상기 모델을 통해 최적해를 얻는 과정은 다소 시간이 적게 걸리는 등 손쉬운 과정이 될 수 있지만, 그 결과로 나온 최적구조는 실제 웹 사이트에서는 적용하는데 무리가 따른다. 왜냐면 첫째, 전체 구조화는 대개의 경우 웹 페이지 숫자가 과도하여 검색 시 별로 도움이 되지 않는다. 그러므로 앞서의 과정과 같은 Top-k 알고리즘이 필요하다. 둘째로, 상기 모형의 단점은 아크구조의 한계이다. 즉, 매우 좋은 웹 페이지가 구조상 아주 깊은 곳에 위치할 경우 몇 단계를 뛰어넘어 이를 사용자에게 제시할 수 있어야 한다. 그러므로 존재하지 않는 링크를 만들어 사용자에게 메타정보로 제공해 줄 수 있는 방법이 필요한 것이다. 이를 위해 본 연구에서는 다음과 같은 라그랑지안 완화법을 수행하고자 한다. 특히, Hop 제약조건과 가장 좋은 k개의 페이지를 구하기 위한 제약조건들은 해를 구하는데 많은 비용이 드는 것을 완화해줄 주 있는 방법이 필요한 것이다. 이를 확인해보는 절차로써 다음과 같이 제약조건에 따른 계산비용을 측정하기 위해 임의의 25개의 노드에 대하여 간단히 실험을 하였다. ILOG CPLEX8.1을 이용하여 모델에 대한 해를 구할 때까지의 시간을 비교하여 제약조건들의 계산비용을 판단하였다.

〈표 2〉 제약조건에 따른 계산비용

	시간(초)
모든 제약조건 포함	48.98
Hop제약 제외	2.56
*Top-k*제약 제외	23.48
Hop과 *Top-k*제약 제외	0.75

<표 2>에서 보면 모든 제약조건을 포함한 경우 가장 긴 시간이 걸렸고, 그 밖에 제약조건들을 제외함으로써 해를 구하는 시간이 크게 줄어드는 것을 알 수 있었다. 우리는 이 제약조건들을 라그랑지안 완화법의 대상으로 하였다. 우리는 각각의 결정변수에 대해서 독립적으로 문제를 풀 수 있도록 제약조건 (5)와 (6)을 완화하고, 추가로 많은 계산비용을 발생시키는 제약조건 (3)을 완화하였다. 먼저 라그랑지안 완화법을 적용하기 위해 목적식을 최소화문제로 바꿔준 후, 각각의 완화할 제약조건에 라그랑지안 승수를 곱하여 목적식에 더해준다. 위의 라그랑지안 완화법 문제는 다음과 같이 각각의 결정변수 X_{ij}, Y_{ij}^t, V_{ij}에 대한 세 개 서브문제, *SP1, SP2, SP3*로 나눌 수 있다.

[서브문제 1]

$$SP1(\mu_{ijt}) = \min\left\{ -\sum_{i=1}^{N}\sum_{j=2}^{N} w_{ij}X_{ij} - \sum_{i=1}^{N}\sum_{j=2}^{N}\sum_{t=2}^{N} \mu_{ijt}X_{ij} \right\}$$

subject to (1), (2), (10) and

$$\mu_{ijt} \geq 0, \quad \forall i, j, t$$

[서브문제 2]

$$SP2(\lambda_t, \mu_{ijt}, \nu_{it}) = \min\left\{ \sum_{i=1}^{N}\sum_{j=2}^{N}\sum_{t=2}^{N} (\lambda_t + \mu_{ijt} - \nu_{it}) Y_{ij}^t \right\}$$

subject to (4), (11) and

$$\lambda_t \geq 0, \quad \forall t$$

$$\mu_{ijt} \geq 0, \quad \forall i, j, t$$

$$\nu_{it} \geq 0, \quad \forall i, t$$

[서브문제 3]

$$SP3(\nu_{ij}) = \min\left\{ \sum_{i=1}^{N}\sum_{j=2}^{N} (\nu_{ij} - w_{ij}) V_{ij} \right\}$$

subject to (7)~(9), (12) and

$$\nu_{it} \geq 0, \quad \forall i, t$$

이 세 개의 서브문제들은 모두 Integrality property 조건을 만족하므로, 이 문제를 선형계획 완화문제로 풀 수 있다. 이 서브문제들 해의 합을 이용하여 라그랑지안 완화법에 대해 해를 구할 수 있다. 이 라그랑지안 완화법의 해를 본래의 문제에 대한 하한(Lower bound) 값이라고 하고, 다음과 같이 나타낼 수 있다.

$$Z_{LB} = SP1(\mu_{ijt}) + SP2(\lambda_t, \mu_{ijt}, \nu_{it}) + SP3(\nu_{it}) - \sum_{i=1}^{N}\sum_{j=2}^{N}\sum_{t=2}^{N} \lambda_t h^t$$

4.1 라그랑지안 상한(Z_{UB})

라그랑지안 완화법으로 해를 구하기 위해서는 먼저 본래의 문제에 가능해를 갖는 상한이 필요하다. 본 연구에서는, 이 상한을 구하기 위해 domain reduction 방법으로 해를 구하는 ILOG SOLVER5.3을 사용한다.

4.2 서브 그라디언트 최적화 알고리듬

라그랑지안 완화법을 이용한 선형계획문제에서는 최대의 하한 값이 선형계획문제의 최적해 또는 근사최적해라고 할 수 있다. 그러므로 이 최대의 하한을 구하기 위하여 서브 그라디언트 최적화 알고리듬(subgradient optimization algorithm)을 이용하였다. 이는, 앞에서 구한 상한 값 Z_{UB}를 이용하여 라그랑지안 승수 λ_t, μ_{ijt}, ν_{it}의 값을 변경하면서 찾아가는 과정을 말한다. 서브 그라디언트 최적화 알고리듬의 적용은 다음과 같다.

[절차]

1) 사용자정의 파라미터 π의 값을 $0 < \pi \leq 2$ 중에서 임의로 선택한다. 이 논문에서는 $\pi = 2$로 설정하였다. 임의의 라그랑지안 승수 λ_t, μ_{ijt}, ν_{it}를 정한다.
2) 현재의 라그랑지안 승수 λ_t, μ_{ijt}, ν_{it}를 이용하여 $SP1(\mu_{ijt})$, $SP2(\lambda_t, \mu_{ijt}, \nu_{it})$, $SP3(\nu_{ij})$로부터 X_{ij}^*와 $Y_{ij}^{t^*}$를 구한다. 또한 이들을 이용해 Z_{LB}를 구한다.
3) 완화된 제약식을 이용하여 서브 그라디언트 값 $G1_t$, $G2_{ijt}$, $G3_{it}$를 구한다.

$$G1_t = \sum_{i=1}^{N}\sum_{j=2}^{N} Y_{ij}^t - h^t, \quad G2_{ijt} = Y_{ij}^t - X_{ij},$$

$$G3_{it} = V_{it} - \sum_{j=2}^{N} Y_{ij}^t$$

4) 스텝사이즈 $T1$, $T2$, $T3$를 구한다.

$$T1 = \frac{\pi(Z_{UB} - Z_{LB})}{\sum_{t=2}^{N} (G1_t)^2}, \quad T2 = \frac{\pi(Z_{UB} - Z_{LB})}{\sum_{t=2}^{N}\sum_{i=1}^{N}\sum_{j=2}^{N} (G2_{ijt})^2},$$

$$T3 = \frac{\pi(Z_{UB} - Z_{LB})}{\sum_{t=2}^{N}\sum_{i=1}^{N} (G3_{it})^2}$$

5) 스텝사이즈와 서브 그라디언트 값을 이용하여, 새로운 라그랑지안 승수들을 구한다.

$$\lambda_t = \max(\lambda_t + T1 \times G1_t,\ 0),$$
$$\mu_{ijt} = \max(\mu_{ijt} + T2 \times G2_{ijt},\ 0),$$
$$\nu_{it} = \max(\nu_{it} + T3 \times G3_{it},\ 0)$$

새로운 라그랑지안 승수를 이용하여 (2)의 과정에서부터 반복하여 계산한다. 이 알고리즘의 종료조건은 반복연산의 수가 600회 이상이 되면 종료한다.

4.3 라그랑지안 휴리스틱

본 연구에서는, 라그랑지안 완화모델을 이용해서 해를 구하기 위해 최적해의 하한 값과 상한 값을 구한 후 서브 그라디언트 최적화 알고리듬을 사용하였다. 그러나 앞에서 서브 그라디언트 최적화 알고리듬을 이용하여 구한 하한 값은 본래 문제에 대해 불능해를 갖기 때문에, 이 불능해를 본래 문제에 대해 가능해를 찾아야 한다. 이를 위하여 본 연구에서는 휴리스틱기법을 이용하였다.

이 연구에서 적용한 휴리스틱을 위해 다음과 같은 가정을 하였다.

1. 앞서 구한 라그랑지안 완화법의 가능해, 즉 본래 문제에 대해 불능해는 3.3절의 제약조건 (3)을 만족시키지 못한다.
2. 웹 사이트의 트리구조에서 경로제약에 위반되는 가지를 찾아 경로제약을 넘어서는 가지를 끊은 후, 다른 노드에 연결함으로써 본래 문제에 대해 좋은 가능해를 구할 수 있다.

이를 위해 우리는 위의 가정을 만족시키는 알고리듬을 개발하여 사용하였다. 이 알고리듬의 절차는 다음과 같다.

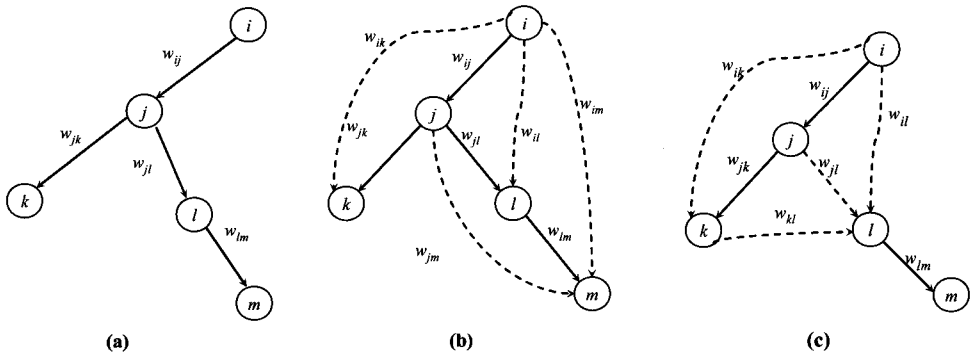
[알고리듬]

(1) 모든 페이지에서부터 경로를 역으로 추적하여 회로를 탐색한 후 회로가 있으면, 회로의 경로 중 하나를 삭제한다. 없다면, 시작페이지에 도달한다.

(2) 회로로부터 삭제된 경로가 향하던 페이지로의 경로를 생성한다. 모든 페이지에 대해 수행이 끝나면 트리구조가 완성된다.

(3) 웹사이트 트리구조의 종점페이지를 찾은 후, 모든 종점페이지에서부터 시작페이지까지의 경로의 수를 계산하여, 경로제약조건을 위반하는지 판단한다. 없으면 종료한다.

(4) 경로제약조건을 위반하는 종점페이지에 대하여, 시작페이지까지의 총 경로의 수가 (Hop수 * 2)이하면 종점페이지에서 (총 경로의 수 - Hop수)번째 되는 경로를 삭제하고, 총 경로의 수가 (Hop수 * 2)이상이면 종점페이지에서 (Hop수)번째의 경로를 삭제한다.



[그림 8] Hop을 고려한 Top-k 노드탐색

(5) 삭제되어 떨어진 종점페이지를 포함한 가지에 대해, 경로제약조건을 만족시키면서 목적식을 최대화하는 경로에 연결한다.
(6) 절차 (3)의 과정으로 돌아가 반복한다.
(7) 마지막으로 최적화된 트리구조에서 Greedy Search를 이용하여 가장 좋은 k개의 페이지를 갖는 웹사이트 구조를 찾는다.

[그림 8]에서 (a)는 상기 절차 (1), (2)과정을 거친 후의 트리구조이다. Hop이 2이라고 할 때, (a)의 구조는 (b)에서와 같이 노드m으로의 연결이 끊어지며 점선으로 된 아크인 후보아크들 중에서 가장 좋은 아크로 연결한다. 이와 같은 과정을 통해서 하한 값으로부터 본래 문제에 가능해를 구할 수 있다.

5. 실험

5.1 최적 Top-k 구조화 실험

본 연구에서 최적 Top-k 적응적 웹 구조화를 수행하도록 다음과 같은 두 종류의 실험을 수행하였다. 우선 가중치의 변경에 따른 차이에 대한 실험과 실제 웹 사이트에 대한 분석을 하였으며, 두 번째로 라그랑지안 완화법의 유용성을 알아보는 실험을 시뮬레이션과 실제 웹 사이트를 대상으로 수행하였다.

우선 최적 Top-k 적응적 웹 구조화의 효용성을 알아보기 위해 기존의 가중치들 즉, 코사인 및 페이지랭크라는 가장 유명한 방식들과 본 방식을 비교하는 실험에 대해 살펴보기로 하자. 본 연구에서는 재현율(Recall)을 차치하고 정확도(Precision)를 살펴보았다. 왜냐면 재현율은 대상노드수가 커지면 차이가 없는 탓도 있지만, 실제 웹 검색에서도 사용자들이 검색결과로부터 가장 좋은 k개(예컨대 *Top* 10개)만 보는 것을 반영한 일반적 연구방향의 일환이다. 따라서 이들 가중치들에 대해 시뮬레이션과 실제 웹사이트에 대해 각각 적용하였다. 여기서 사용된 모수(parameters)들은 다음과 같다. 데이터는 노드와 아크의 수가 각각 다음과 <표 3>과 같으며, 이때 해당 최적 구조에서 최소 깊이(Min depth)와 최대 깊이(Max depth) 그리고 최소 인링크수(Min degree)와 최대값(Max degree) 또한 결과적으로 얻은 *Top-k* 트리의 결과에 대해 정확도의 합계가 각각 정리되어 있다.

〈표 3〉 실험 데이터의 모수 및 Top-10 결과

Dataset w/ #nodes	50	100	300	500	1000
& #arcs	257	464	1468	2022	4413
Min depth	4	3	4	2	3
Max depth	9	9	9	7	10
Min degree	3	3	7	6	16
Max degree	11	22	29	15	81
Sum (Top-10 results)	7.413	9.054	9.776	9.911	9.872

위의 표로부터 *Top*-10 합의 결과는 큰 웹 노드 3종류에 대해서 매우 높게(9.7이상) 나타났는데, 이는 *Top*-10 결과의 각각의 정확도가 1에 가까이 나왔다는 의미이다.

5.2 라그랑지안 완화법 실험

라그랑지안 완화법에 대해서도 2가지 종류의 실험을 수행하였다. 첫째, 시뮬레이션을 수행한 결과와 둘째, 실제 웹 사이트에 대해 수행한 결과를 다음과 <표 5>와 같이 제시하였다.

우선, 시뮬레이션을 위해 라그랑지안 완화법에 대해서는 각각의 페이지의 수를 10, 20, 30, 40, 50에 대해서 실험을 하였고, 제시한 모델에 대하여 각 페이지마다 무작위로 3또는 4로 경로제약을 주면서 시행하였다. 각 모델의 풀이는 본래의 정수계획 문제에 대한 CPLEX를 이용한 풀이와 라그랑지안 완화법을 이용한 해법의 시간 및 최적해를 비교하였다. 정수계획 문제에 대한 해는 정확한 최적해이며, 라그랑지안 완화법의 해는 가능해 값을 나타내었다. 이 실험은 Pentium4 3.0GHz의 퍼스널컴퓨터에서 ILOG CPLEX8.1과 SOLVER5.3을 이용하여 실

〈표 4〉 구글 웹 검색 결과 및 웹사이트 상세사항

#	Returned URLs	Anchor text	Node	Arc
1	http://www.djeffrey.id.au/	ADHD related functions and causes dysregulation of these functions. …	12	62
2	http://www.brainhealer.com/	NeuroTherapy ADD, ADHD, Anxiety, Autism, Carpal Tunnel Syndrome	54	898
3	http://www.myadhd.com/causesofadhd.html	ADHD exposure to toxic substances in fetus and brain injury due to trauma …	298	1179
4	http://www.adoptionarticlesdirectory.com/Article/ADHD-or-Hyperarousal--Hyperactivity-in-Traumatized-and-Adopted-Children/	ADHD that the child is born with, but is hyperarousal caused by early trauma, neglect or attachment disorder. …	890	10610
5	http://www.brainmattersinc.com/brain_injury.html	Brain SPECT Imaging is recognized as MRI and CT.	56	826
6	http://www.familyHopecenter.org/	These short, tight cycles are the result of pressure on the brain.	383	1677
7	http://www.carolinewalrad.com/ADHD-research.shtml	ADHD Research. Scottsdale, Arizona-ABC Wellness Center; The improvement of injury related brain waves shows the QXCI extremely effective. …	26	194
8	http://www.academyanalyticarts.org/galvesealker.htm	Debunking the Science Behind ADHD as a "Brain Disorder".	113	1453
9	http://www.erinelster.com/	Medical research focused upon dopamine neurotransmitter involved in ADHD. …	28	298
10	http://www.corepsychblog.com/	Brain Trauma, Alcohol, Drugs and… Denial : …	183	8755

험을 하였다. 그에 대한 결과는 다음 <표 5>와 같다. 실험에서는 앞서 설명한 <표 2>에서와 같이 시간제약이 심한 2가지 조건 즉, Top-k와 Hop 조건을 완화할 경우에 대한 것으로 매우 적은 오차범위 내에서 바람직한 결과를 주는 것으로 나타났다.

〈표 5〉 라그랑지안 시뮬레이션 계산결과

노드수	방 법	시간(초)	해	오차*(%)
10	라그랑지안	0.75	103.6	0.06
	정수계획법	0.55	97.8	
20	라그랑지안	1.53	211.8	0.04
	정수계획법	254.84	204.1	
30	라그랑지안	4.92	307.6	0.02
	정수계획법	1760.22	301.9	
40	라그랑지안	11.48	408.8	0.01
	정수계획법	6453.28	403.4	

주) *오차** = *(라그랑지안 해-정수계획법 해) / 라그랑지안 해.*

<표 6>에서는 전자회사들을 중심으로 11개의 구현사례를 보였다. <표 6>에서 왼쪽에는 해당 웹사이트 그중에서도 특정 웹페이지를 루트로드로 지정한 경우, 그 하위 노드의 숫자와 그들 노드사이의 아크(hypertext link) 숫자 및 그것을 분석하는데 걸린 시간이 각각 제시되었다. 예컨대, 처음 있는 apple.com의 경우 index.html에서부터 접근 가능한(reachable) 노드수 1126개, 아크 수는 21294개이고 계산에 걸린 시간은 323초이다. 그 아래열의 경우 iphone이라는 하위 디렉터리를 루트로 삼으면 노드수, 아크수, 걸린 시간이 각각 71, 1879, 42초로 분석된 결과를 나타내준다.

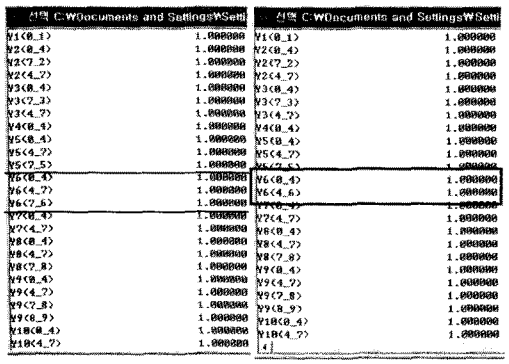
다음으로 특정 웹 사이트를 대상으로 앞에서 모델링을 적용한 실험을 수행하였다. 이 문제에 대한 CPLEX를 수행한 결과가 다음 [그림 9]에 제시되어 있다. 예컨대, 노드 6번 관점에서 Hop 제약조건을 적

〈표 6〉 다양한 웹 사이트(URL)에 대한 분석결과

URL of the root node	Number of web nodes	Number of arcs	Elapsed time
www.apple.com	1126	21294	323
www.apple.com/iphone	71	1879	42
www.hitachi.com	661	6122	114
www.intel.com	1282	4073	295
www.intel.com/design/celect/mp.htm	84	250	19
www.lge.com	554	50345	205
www.microsoft.com	2201	6642	343
www.nec.co.jp	1468	5194	330
www.nokia.com	559	9187	412
www.samsung.com	511	3453	117
www.samsung.co.kr	31	88	3

용한 결과를 살펴보고자 한다. [그림 9]의 (a) 즉, 최적화 모형의 결과에서 알 수 있는 바와 같이 노드 6번까지 이르는 경로는 0 -> 4 -> 7 -> 6의 3단계가 소요된다. 이를 제약조건 "$h_6 < 3$"을 가할 경우 0 → 4 → 6의 2단계로 단축된 것을 확인할 수 있다.

또한 이를 전체적인 모식도에서 살펴보면 [그림 10]과 같이, 경로가 변경된 것을 알 수 있다. 예컨대 [그림 9]에서 살펴본 노드 6번까지의 경로(Hop) 3에서 2로의 유익을 얻는 경우, 그에 대한 목적함수 값이 43.360에서 40.761로 손실을 입은 것을 알 수 있다. 그러므로 Hop 제약을 적용하여 원하는 페이지(Target)를 원하는 만큼 상위로 올리는 효과를 얻기 위해서는, 한편으로 최적해의 손실을 어느 정도 감수해야 하는 지를 확인할 수 있었다.

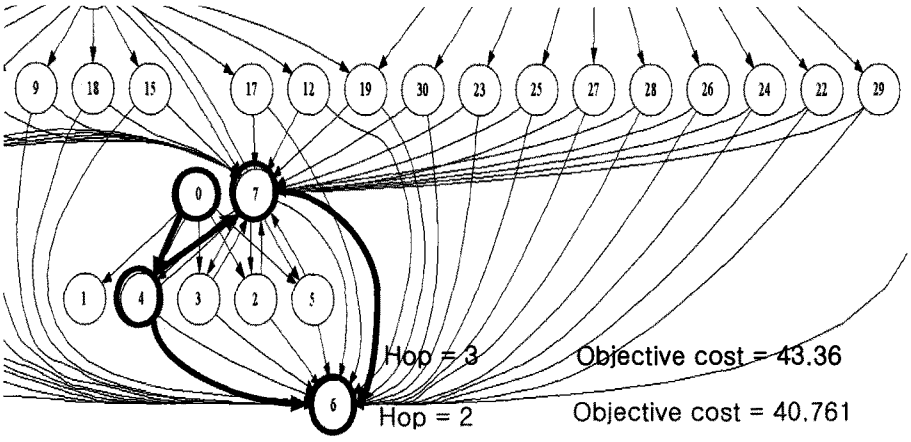


(a) (b)

[그림 9] 노드 6번에 대한 제약조건 적용 전(a) 및 적용 후(b)에 대한 구현화면

7. 결 론

본 연구에서는 웹 사이트로부터 정보를 얻으려 할 때 효율적으로 웹 검색에 적용할 수 있도록 웹의 정보를 최적 구조화하고, 특히 가장 좋은 k개의 정보를 보여주는 웹 페이지로의 접근경로를 제시하는 문제를 정수계획법을 이용하여 모델링하였다. 이러한 접근법의 특징은 다음과 같다. 우선 최적해 접근법을 정보검색의 방법론으로 적용하였으며, 그 중에서도 네트워크 모델과 정수계획법을 이용하여 웹 정보의 최적 구조화를 시도하였다. 또한 상기 얻어진 최적구조화 해로부터 웹사용자 즉, 일반사용자 및 웹로봇이 특정 깊이 이상으로 접근 경로가 구성되지 않도록 제약을 하면서 가장 좋은 k개의 페이지(*Top-k* page)에 대한 구조를 추가적으로 제시하였다. 따라서 웹 정보의 최적 구조화뿐만 아니라 실용적으로도 가장 좋은 k개의 페이지에 대하여 접근성을 높일 수 있도록 하는 것이다. 이러한 모델은 웹서버 관점에서는 사이트맵이라는 논리적인 구조를 표현할 때 활용될 수 있으며, 클라이언트나 검



[그림 10] 노드 6번에 대한 변경과정 및 최적해의 변화내역

색엔진의 입장에서는 사용자가 접근해야할 대상 웹 페이지들을 구조화하여 표현할 수 있는 장점이 있다. 본 연구에서는 이 모델을 복잡하게 하는 제약조건의 수를 줄이고, 계산에 있어 부담을 주는 제약조건들을 라그랑지안 완화법을 통해 완화함으로써 사용자에게 편익을 제공함과 동시에 근접해를 통한 최적해의 손실정도를 구할 수 있음을 제시하였다. 이러한 모델을 기반으로 가상노드 및 실제 웹에 적용하였으며, 제약조건을 완화한 해를 구하여 최적해의 유지와 제약식의 완화효과에 대한 비용분석을 하여 그 실제성을 입증하였다.

향후 다양한 사용자 프로파일을 감안한 개인화된 검색가중치를 적용하는 연구와 본 연구에서 제시된 새로운 시스템을 구축하여 기존의 검색엔진들과의 비교하는 실증적 접근, 특히 서버 및 클라이언트에 대한 사이트맵의 효과성 입증, 그리고 새로운 브라우징 패턴에 관한 연구 등이 필요하다.

참 고 문 헌

[1] Abiteboul, S., Predal, M., and Cobena, G., "Adaptive On-Line Page Importance Computation," In Proc. *WWW*, (2003), pp.280-290.

[2] Álvarez, M., J. Raposo, A. Cacheda, F. Bellas, and V. Carneiro, "DeepBot : a Focused Crawler for Accessing Hidden Web Content," In Proc. *DEECS*, (2007), pp.18-25.

[3] Bergman, M., "The Deep Web : Surfacing Hidden Value," *Journal of Electronic Publishing*, Vol.7, No.1(2001).

[4] Botafogo, R., E. Rivlin, and B. Shneiderman, "Structural Analysis of Hypertexts : Identifying Hierarchies and Useful Metrics," *ACM TOIS*, Vol.10, No.2(1992), pp.142-180.

[5] Brynjolfsson, E., A. Dick, and M. Smith, "Search and Product Differentiation at an Internet Shopbot," *MIT Sloan Working Paper*, No. 4441-03(2003).

[6] Caldo, P., B. Ribeiro-Neto, and N. Ziviani, "Local Versus Global Link Information in the Web," *ACM TOIS*, Vol.21, No.1(2003), pp.42-62.

[7] Chakrabarti, S., "Dynamic Personalized PageRank in Entity-Relation Graphs," In Proc. *WWW*, (2007), pp.571-580.

[8] Chang, C., M. Kayed, M. Girgis, and K. Shaalan, "A Survey of Web Information Extraction Systems," *IEEE TKDE*, Vol.18, No.

10(2006), pp.1411-1428.

[9] Cilibrasi, R., and P. Vitányi, "The Google Similarity Distance," *IEEE TKDE*, Vol.19, No.3(2007), pp.370-383.

[10] Eichmann, D., "The RBSE spider : Balancing Effective Search Against Web Load," In Proc. *WWW*, (1994), pp.113-120.

[11] Eiron, N., K. McCurley, and J. Tomlin, "Ranking the Web Frontier," In Proc. *WWW*, (2004), pp.309-318.

[12] Fu, Y., M. Creado, and C. Ju, "Reorganizing Web Sites Based on User Access Patterns," In Proc. *CIKM*, (2002), pp.583-585.

[13] Garofalakis, J., P. Kappos, and D. Mourloukos, "Web Site Optimization Using Page Popularity," *IEEE Internet Computing*, Vol.3, No.4 (1999), pp.22-29.

[14] Gouveia, L., "Multicommodity Flow Models For Spanning Trees with Hop Constraints," *European Journal of Operational Research*, Vol.95, No.1(1996), pp.178-190.

[15] Gyongyi, Z., P. Berkhin, H. Garcia-Molina, and J. Pedersen, "Link Spam Detection Based on Mass Estimation," In Proc. *VLDB*, (2006), pp. 439-450.

[16] Hammami, M., Y. Chahir, and L. Chen, "WebGuard : A Web Filtering Engine Combining Textual, Structural, and Visual Content-Based Analysis," *IEEE TkDE*, Vol.18, No.2(2006), pp.272-284.

[17] Haveliwala, T., "Topic-Sensitive PageRank : A Context-Sensitive Ranking Algorithm for Web Search," *IEEE TKDE*, Vol.15, No.4(2003), pp.784-796.

[18] Henzinger, M.R., A. Heydon, M. Mitzenmacher, and M. Najork, "On Near-Uniform URL Sampling," *Computer Networks*, Vol.33, No.1-6(2000), pp.295-308.

[19] Henzinger, M.R., "Combinatorial algorithms for web search engines : three success stories," In Proc. *SODA*, (2007), pp.1022-1026.

[20] Ikeda, R., K. Zhao, and H. Garcia-Molina, "Matching Hierarchies Using Shared Objects," In Proc. *ECDL*, (2008), pp.209-220.

[21] John, J.C., and U. Schonfeld, "RankMass Crawler : A Crawler with High PageRank Coverage Guarantee," In Proc. *VLDB*, (2007), pp.375-386.

[22] Kawatra, R., "A Hop Constrained Min-Sum Arborescence with Outage Costs," In Proc. *HICSS*, (2003), pp.2648-2656.

[23] Kleinberg, Jon. M., "Navigation in a Small World", *Nature*, Vol.406, No.6798(2000), p. 845.

[24] Lin, C.C., "Optimal Web Site Reorganization Considering Information Overload and Search Depth," *European Journal of Operational Research*, (2005), pp.839-848.

[25] Meng, W., C. Yu, and k. Liu, "Building Efficient and Effective Metasearch Engines," *ACM Computing Surveys*, Vol.34, No.1(2002), pp.48-89.

[26] Miller, R., and k. Bharat, "Sphinx : A Framework for Creating Personal, Site-Specific Web Crawlers," In Proc. *WWW*, (1998), pp.119-130.

[27] Najork, M., and J.L. Wiener, "Breadth-First Crawling Yields High-Quality Pages," In Proc. *WWW*, (2001), pp.114-118.

[28] Najork, M., H. Zaragoza, and M.J. Taylor, "Hits on the Web : How Does It Compare?" In Proc. *SIGIR*, (2007), pp.471-478.

[29] Novak, J., P. Raghavan, and A. Tomkins, "Anti-Aliasing on the Web," In Proc. *WWW*, (2004), pp.30-39.

[30] Ntoulas, A., Zerfos, P., and Cho, J., "Downloading Textual Hidden Web Content Through Keyword Queries," In Proc. *ICDL*, (2005), pp.

100-109.

[31] Pandurangan, G., P. Raghavan, and E. Upfal, "Using PageRank to Characterize Web Structure," In Proc. *COCOON*, (2002), pp.330-339.

[32] Perkowitz, M., and O. Eizioni, "Toward Adaptive Web Sites : Conceptual Framework and Case Study," *Artificial Intelligence*, Vol.118, No.1-2(2000), pp.245-275.

[33] Robertson, S.E. and S. Walker, "Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval, ACM SIGIR, (1994), pp.232-241.

[34] Siganos, G., M. Faloutsos, P. Faloutsos, and C. Faloutsos, "Power Laws and the AS-level Internet Topology," *IEEE Transactions on Networking*, Vol.11, No.4(2003), pp.514-524.

[35] Varadarajan, R., V. Hristidis, and T. Li, "Beyond Single-Page Web Search Results," *IEEE TKDE*, Vol.20, No.3(2008), pp.411-424.

[36] Wookey, Lee and S. Lim, "Maximum Rooted Spanning Trees for the Web," OTM Workshops, Vol.2(2006), pp.1873-1882.

[37] Wookey, Lee, S. Kim, and S. Kang, "Structuring Web Sites Using Linear Programming," LNCS, (2004), pp.328-337.

[38] Xu, G., and W. Ma, "Building Implicit Links from Content for Forum Search," In Proc. *SIGIR*, (2006), pp.300-307.

[39] http://www.websiteoptimization.com/.

[40] http://www.poweradmin.com/servermonitor/.