МЕТОД ПОБУДОВИ ОПТИМАЛЬНИХ ШЛЯХІВ ПЕРЕГЛЯДУ РЕЗУЛЬТАТІВ ВЕБ-ПОШУКУ НА ОСНОВІ ЕВРИСТИЧНИХ АЛГОРИТМІВ

Володимир Дубовой, Олександар Краковецький, Ольга Глонь

Винницкий национальный технический университет,
21021, г. Винница, Хмельницкое шоссе, 95

Вступ

Задача пошуку оптимальних маршрутів на практиці розглядається дуже часто, наприклад, для визначення оптимальних транспортних маршрутів (транспортна задача, задача комівояжера), в телекомунікаційних мережах (оптимальний шлях передачі сигналів) тощо, тобто в тих областях, де є розгалужена мережа, а об'єкт має складну структуру, яку можна представити у вигляді графу [1].

Інтернет («всесвітня павутина») є величезною мережею з великою кількістю вузлів та зв'язків, що робить процес пошуку та перегляду інформації в Інтернеті досить складним. Для подолання цих проблем існують пошукові системи, проте їх основне завдання є пошук інформації, а не навігація і перегляд. Кожен користувач, шукаючи щось в Інтернеті за допомогою пошукових систем, зіштовхується із проблемою - які саме веб-ресурси із отриманого списку і в якій послідовності необхідно переглянути для найповнішого огляду бажаної теми. До того ж, на сучасному етапі розвитку Інтернету питання пошуку інформації не стоїть як «знайти інформацію на задану тему», а скоріше визначається як «знайти серед всього об'єму різноманітної інформації ту, що найповніше описує задану тему». Описана проблема може бути визначена як проблема пошуку оптимальних шляхів перегляду сайтів, яка в свою чергу може бути транспонована в проблему пошуку оптимального шляху в направленому графі при заданих обмеженнях.

Огляд літератури

Проблема аналізу взаємозв'язків між веб-сайтами та знаходження оптимальних маршрутів перегляду результатів пошуку є темою досліджень всіх провідних пошукових систем, таких як Google [2], Yahoo, MSN тощо. Крім того, ряд інших компаній пропонують свої продукти (як комерційні, так і з відкритим програмним кодом) для інтеграції з пошуковими системами для більш ефективного пошуку в мережі. Одним із таких продуктів є система візуалізації взаємозв'язків між сайтами, що має назву TouchGraph™ GoogleBrowser [3]. Ця система (рисунок 1), містить багато корисної інформації (візуальної і статистичної), проте не дає рекомендацій щодо оптимальних шляхів перегляду. Аналіз інших існуючих рішень дає змогу зробити висновок, що на сьогоднішній день не існує ефективних рішень даної проблеми.

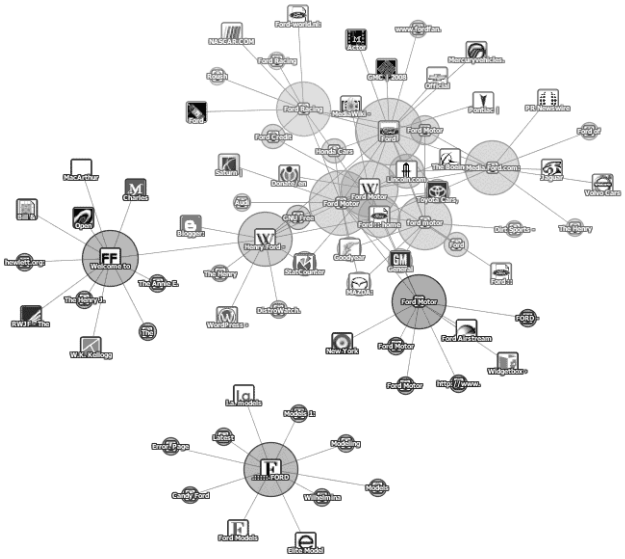

Рис.1. Система аналізу взаємозвязків між сайтами TouchGraph™ GoogleBrowser

Актуальність роботи

Велика кількість задач, пов'язаних із складними системами та системами прийняття рішень, зводяться до вирішення задачі пошуку оптимальних шляхів. Проте специфіка кожної конкретної системи як складного об'єкту вносить свої особливості в процес прийняття рішень. Динамічний розвиток існуючих складних систем, до яких відноситься і мережа Інтернет, створює нові задачі, для яких відсутні ефективні рішення. А тому адаптація існуючих алгоритмів на графах та розробка нових методів пошуку оптимальних шляхів в мережі Інтернет є новою і актуальною задачею.

Мета за задачі досліджень

Метою даної роботи є розробка методу пошуку оптимальних шляхів перегляду результатів веб-пошуку на основі евристичних алгоритмів.

Постановка задачі

Під інформативністю деякого шляху p будемо розуміти певну безрозмірну величину, яка є сумою інформативностей усіх веб-ресурсів, що входять в цей шлях[1]:

$$I_p^n = \sum_{j=1}^{n-1} i_j, \tag{1}$$

де n - кількість ресурсів (вершин графу), i_j - інформативність конкретного ресурсу.

Нехай в результаті деякого пошукового запиту $query$ ми отримали структурну модель у вигляді графу G, що відображає взаємозв'язки між сайтами, отриманими в результаті пошуку. Необхідно знайти такий шлях, який охоплював би максимальну кількість релевантних веб-ресурсів, інформативність якого була б максимальною, а інформативність даних, що повторюються – мінімальною.

Формально задача може бути описана наступним чином:

$$find\ \ p\ with\ n \to max, I_k^n \to max, I_{k\ repeated}^n \to min \tag{2}$$

Обґрунтування вибору методу оптимізації на графах

Задачею пошуку оптимальних шляхів в направлених графах займалися і продовжують займатися багато вчених та дослідників. На сьогоднішній день існує ряд ефективних методів та алгоритмів, що вирішують дану задачу, а також велика кількість їх модифікацій. До класичних методів пошуку оптимальних шляхів в направлених графах належать метод пошуку в ширину, метод пошуку в глибину та метод Дейкстри [1]. До евристичних алгоритмів належать метод A*, генетичні алгоритми [4] та мурашині алгоритми [5]. Специфіка останніх алгоритмів – використання природної поведінки мурах для пошуку шляхів – якнайкраще відповідає специфіці проблеми пошуку оптимальних шляхів в Інтернеті. Крім того, дані алгоритми є абсолютно збіжними і дають досить точний результат за допустимий час, що є обов'язковою вимогою до всіх методів та алгоритмів, що використовуються в мережі Інтернет. В даній роботі буде використовуватися саме цей алгоритм, модифікацію якого розглянуто в наступному розділі.

Опис методу

Моделювання поведінки мурах пов'язане з розподілом феромону на шляху – ребрі графу. Ймовірність того, що мураха піде по конкретному ребру пропорційна кількості феромону на цьому ребрі, а кількість феромону, що відкладається на шляху, пропорційна «відстані» між двома веб-ресурсами. В якості міри відстані може бути обрана будь-яка із загальновідомих мір відстаней [6]. Таким чином, чим «ближчими» будуть два довільні ресурси, тим більше феромону буде відкладено на ребрі, що їх з'єднує, і відповідно, тим більше мурах буде рухатися по даному маршруту. Для попередження знаходження локально оптимальних рішень необхідно додати зворотній зв'язок у вигляді випарування феромону. Це дасть змогу отримати стійке оптимальне рішення.

Опишемо правила поведінки мурах при виборі маршруту:

1. Мурахи можуть рухатися по тим самим ребрам декілька разів, якщо це необхідно, проте це є небажаним, так як призводить до дублювання фрагментів шляху. Крім того, в спірних моментах мурахи повинні віддавати перевагу тим ребрам, які вони ще не відвідали (якщо такі є). Позначимо через $J_{i,k}$ список ресурсів, які ще не відвідувала комаха k, що знаходиться на i-у ресурсі.
2. Видимість визначається як евристичне бажання комахи відвідати ресурс j, якщо вона знаходиться на ресурсі i. Будемо обчислювати як обернено пропорційну величину до значення «відстані» між ресурсами i та j. Для цього використаємо критерій χ^2 Пірсона [7]:

$$\chi^2 = n_1 n_2 \sum_{i=1}^{k} \frac{1}{m_{1,i} + m_{2,i}} \left(\frac{m_{1,i}}{n_1} - \frac{m_{2,i}}{n_2} \right)^2, \tag{3}$$

де n_1 - об'єм першого тексту, n_2 - об'єм другого тексту, $m_{1,i}$ - частота i-го признаку в першому тексті, $m_{2,i}$ - частота i-го признаку в другому тексті.

3. «Нюх» комах визначається їх здатністю відчувати запах феромону, що підтверджує бажання відвідати ресурс j із ресурсу i на основі досвіду інших мурах. Кількість феромону на ребрі (i, j) в певний момент t позначимо як $\tau_{i,j}(t)$.

[1] Метод оцінки інформативності конкретного ресурсу в даній роботі не розгдядається.

Таким чином на основі вищерозглянутих правил запишемо правило ймовірності переходу k-ої мурахи з ресурсу i в ресурс j:

$$\begin{cases} P_{ij,k}(t) = \dfrac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum\limits_{l=1}^{J_{i,k}} \left[\tau_{il}(t)\right]^{\alpha} \cdot \left[\eta_{il}\right]^{\beta}}, j \in J_{i,k}, \\ P_{ij,k}(t) = 0, j \notin J_{i,k}. \end{cases} \quad (4)$$

Параметри α і β визначаються дослідним шляхом.

Кількість феромону, що відкладе мураха k на ребрі (i, j), перейшовши з ресурсу i на ресурс j, визначається наступною формулою:

$$\Delta\tau_{ij,k}(t) = \begin{cases} f_{p_k(t)}, (i, j) \in p_k(t), \\ 0, (i, j) \notin p_k(t), \end{cases} \quad (5)$$

де $p_k(t)$ - шлях, що пройшла мураха k на момент часу t,

$f_{p_k(t)}$ - числова міра оптимальності шляху p.

Числова мірою оптимальності шляху розраховується на основі оптимізаційної функції

$$f_p = n_p I_p v_p, \quad (6)$$

де n_p - кількість ресурсів, що входять в маршрут p,

I_p - інформативність (1) маршруту p,

v_p - середня швидкість зміни інформативності з врахуванням порядкового номеру ресурсу в маршруті p, що обчислюється за наступною формулою:

$$v_p = \left(\sum_{l=1}^{n_k - 1} \frac{(l+1) i_l}{\sum\limits_{j=1}^{l+1} i_j} \right) \Big/ \left(n_p - 1\right). \quad (7)$$

Випаровування феромону обчислюється за наступною формулою:

$$\begin{aligned} \tau_{ij}(t+1) &= (1-p) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t), \\ \Delta\tau_{ij}(t) &= \sum_{k=1}^{m} \Delta\tau_{ij,k}(t), \end{aligned} \quad (8)$$

де $p \in [0;1]$ - коефіцієнт випаровування, m - кількість мурах (ресурсів).

Висновки

В даній роботі було розглянуто сучасний стан розвитку проблеми пошуку оптимальних шляхів перегляду Інтернет-ресурсів та було зроблено висновок, що на сьогоднішній день ця проблема є актуальною і до кінця не вирішеною.

Дана проблема була інтерпретована як проблема пошуку оптимального шляху в направленому графі при заданих обмеженнях. Серед усіх методів вирішення даної проблеми було обрано метод мурашиних колоній, що найкращим чином підходить до нашого об'єкту.

Було модифіковано метод мурашиних колоній та адаптовано його до об'єкту, що розглядається.

Список літератури:

[1] Дубовой В.М. Моделювання систем контролю та керування. Навчальний посібник. - Вінниця, ВНТУ, 2005. - 176 с.

[2] Інформаційно-пошукова система Google // Режим доступу http://google.com.ua.

[3] Система візуалізації взаємозв'язків між сайтами TouchGraph™ GoogleBrowser // Режим доступу http://www.touchgraph.com/TGGoogleBrowser.html.

[4] Д. Рутковская, М. Пилиньский, Л. Рутковский. Нейронные сети, генетические алгоритмы и нечеткие системы // Горячая Линия – Телеком. - 2007 г., - 452 с.

[5] Чураков Михаил, Якушев Андрей. Муравьиные алгоритмы // Режим доступу http://rain.ifmo.ru/cat/data/theory/unsorted/ant-algo-2006/article.pdf.

[6] Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004. – 336 с.

[7] Бормашов Д. А. Дипломная работа "Кластерный анализ текстов" // Режим доступу http://inf.tsu.ru/library/DiplomaWorks/CompScience/2006/bormashov/diplom.pdf.