**ỨNG DỤNG CÁC THUẬT TOÁN UNINFORMED SEARCH ĐỂ TÌM ĐƯỜNG ĐI TỪ MỘT ĐƯỜNG DẪN URL ĐẾN MỘT URL ĐẾN MỘT ĐƯỜNG DẪN URL KHÁC**

**Nguyễn Thanh Lập 1, Luyện Ngọc Thanh 1, Ôn Đức Khang 1**

1 *Đại học Sư Phạm Kỹ Thuật thành phố Hồ Chí Minh*

Tóm tắt:

Mạng Internet phát triển mang đến rất nhiều tiện ích cho cuộc sống hiện đại ngày nay và đã thúc đẩy việc kinh doanh chuyển dần từ truyền thống sang thương mại điện tử. Thương mại điện tử là quá trình mua, bán, vận chuyển hay trao đổi các sản phẩm, dịch vụ, thông tin qua mạng máy tính, đặc biệt là mạng Internet. Và nói đến thương mại điện tử thì chúng ta nghĩ ngay đến tiếp thị điện tử - một yếu tố quan trọng và không thể thiếu giúp hoạt động kinh doanh này phát triển rực rỡ. Và để quản lý việc tiếp thị, cụ thể là quảng cáo trên môi trường internet thì khá là rắc rối phức tạp do nó bao gồm một lượng thông tin khổng lồ và luôn không ngừng thay đổi. Vì vậy, việc xây dựng một phương pháp tìm kiếm dựa trên hệ thống trí tuệ nhân tạo nhằm tìm đường dẫn từ một trang web đến một đường dẫn khác là cần thiết. Nhằm giúp quản lý mô hình quảng cáo trên Internet giữa các trang web có liên kết, hợp tác với nhau một cách dễ dàng và hiệu quả hơn. Ngoài ra việc này còn góp phần làm tăng hiệu quả của các quảng cáo do có thể tính được các hướng để đi đến một đường dẫn nào đó để có thể gắn quảng cáo một cách phù hợp và hiệu quả nhất. Nghiên cứu này ứng dụng các thuật toán tìm kiếm để đưa ra đường đi cụ thể từ một đường dẫn đến một đường dẫn khác nếu có. Nhóm tác giả xây dựng và kiểm chứng dựa trên tập các trang web HTML đơn giản để mô phỏng các trang web trong thực tế.

Từ khóa: trí tuệ nhân tạo trong thương mại điện tử, thuật toán tìm kiếm, tiếp thị điện tử, quảng cáo, SEO website.

1. **ĐẶT VẤN ĐỀ**

Quảng cáo trực tuyến, còn được gọi là tiếp thị trực tuyến, quảng cáo Internet, quảng cáo kỹ thuật số hoặc quảng cáo web, là một hình thức tiếp thị và quảng cáo sử dụng Internet để truyền tải thông điệp tiếp thị quảng cáo đến người tiêu dùng. Quảng cáo trực tuyến đóng vai trò quan trọng trong thương mại điện tử vì đây một phương pháp nhanh chóng, linh hoạt và có thể đo lường được để điều chỉnh các thông điệp tiếp thị của công ty đến mọi người trên khắp thế giới. Năm 2016, doanh thu quảng cáo trên Internet ở Hoa Kỳ đã vượt qua các kênh truyền hình cáp và truyền hình phát sóng. Năm 2017, doanh thu quảng cáo trên Internet ở Hoa Kỳ đạt 83,0 tỷ đô la, tăng 14% so với 72,50 tỷ đô la doanh thu trong năm 2016. Năm 2019, chi tiêu quảng trực tuyến trên toàn thế giới tăng 17,6% lên 333,25 tỷ đô la. Điều đó có nghĩa là, lần đầu tiên quảng cáo trực tuyến chiếm khoảng một nửa thị trường quảng cáo toàn cầu. Ở một số quốc gia, bao gồm Vương quốc Anh, Trung Quốc, Na Uy và Canada, quảng cáo trực tuyến đã trở thành phương tiện quảng cáo thống trị. Năm nay, Mỹ và Hà Lan sẽ tham gia nhóm đó, với tỷ lệ tương ứng là 54,2% và 52,6% tổng chi tiêu quảng cáo. Ở Nga, một nửa tổng đầu tư quảng cáo sẽ chuyển sang kỹ thuật số. Thế nhưng nhiều người tiêu dùng thấy quảng cáo trực tuyến gây khó chịu và ngày càng chuyển sang chặn quảng cáo vì nhiều lý do. Họ đã bắt đầu tìm cách loại bỏ quảng cáo với các tiện ích như AdBlock Plus được tích hợp ngay trên các trình duyệt Firefox, Chrome… để lọc bỏ quảng cáo không mong muốn, hay đơn giản chỉ là bỏ qua không quan tâm đến quảng cáo. Trước thực tế đó, nhiều công ty lớn đang tìm cách thay đổi phương thức tiếp thị hướng đến tính chia sẻ, thân thiện và tương tác để quảng bá sản phẩm và chăm sóc khách hàng. Quảng cáo hiện nay không chỉ còn dừng ở một hình ảnh banner thông thường nữa, mà đã tiến đến tích hợp với những thông tin và tiện ích thu hút người dùng nhiều hơn. Vì vậy việc tính được các hướng đi từ một đường dẫn đến một đường dẫn nào đó là cần thiết để có thể gắn quảng cáo một cách phù hợp và hiệu quả nhất. Trong những năm gần đây, trí tuệ nhân tạo (AI) được ứng dụng vào các ứng dụng web ngày càng nhiều, điển hình như là sematic web. Từ đó giúp năng cao trải nghiệm của người dùng một cách hiệu quả. Do đó, việc xây dựng một phương pháp tìm kiếm dựa trên hệ thống trí tuệ nhân tạo nhằm tìm đường dẫn từ một trang web đến một đường dẫn khác sẽ nhằm giúp quản lý mô hình quảng cáo trên Internet giữa các trang web có liên kết, hợp tác với nhau một cách dễ dàng và hiệu quả hơn. Hơn hết việc này sẽ giúp các nhà quảng cáo có thể tìm ra những trang web phù hợp để đặt quảng cáo từ đó giúp hạn chế làm phiền đến người dùng. Hiện nay đã có một số nhóm tác giả đã sử dụng các thuật toán uninformed search và đã xuất bản sách như: [Mining the worldwide web](https://www.springer.com/gp/book/9780792373490), [Web Intelligence](https://www.springer.com/gp/book/9783540443841?utm_campaign=3_pier05_buy_print&utm_content=en_08082017&utm_medium=referral&utm_source=google_books#otherversion=9783662053201), [Intelligent Information Integration for the Semantic Web](https://link.springer.com/book/10.1007/b100348), ... để nghiên cứu và giải quyết vấn đề. Với các nghiên cứu trên họ đã tìm được đường đi từ một trang web đến một đường dẫn khác một cách chính xác, nhanh chóng nhất để có thể tối ưu hoá quảng cáo nhưng lại không làm phiền đến người dùng.

1. **PHƯƠNG PHÁP**

2.1. Thuật toán tìm kiếm theo chiều sâu (Depth-First Search - DFS)

Giải thuật tìm kiếm theo chiều sâu (Depth-First Search – viết tắt là DFS), còn được gọi là giải thuật tìm kiếm ưu tiên chiều sâu, là một [thuật toán](https://vi.wikipedia.org/wiki/Thu%E1%BA%ADt_to%C3%A1n) duyệt hoặc tìm kiếm trên một [cây](https://vi.wikipedia.org/wiki/C%C3%A2y_(l%C3%BD_thuy%E1%BA%BFt_%C4%91%E1%BB%93_th%E1%BB%8B)) hoặc một [đồ thị](https://vi.wikipedia.org/wiki/%C4%90%E1%BB%93_th%E1%BB%8B). Thuật toán khởi đầu tại gốc (hoặc chọn một đỉnh nào đó coi như gốc) và phát triển xa nhất có thể theo mỗi nhánh.

Thông thường, Depth-First Search là một dạng [tìm kiếm thông tin không đầy đủ](https://vi.wikipedia.org/w/index.php?title=T%C3%ACm_ki%E1%BA%BFm_th%C3%B4ng_tin_kh%C3%B4ng_%C4%91%E1%BA%A7y_%C4%91%E1%BB%A7&action=edit&redlink=1) mà quá trình tìm kiếm được phát triển tới đỉnh con đầu tiên của nút đang tìm kiếm cho tới khi gặp được đỉnh cần tìm hoặc tới một nút không có con. Khi đó giải thuật [quay lui](https://vi.wikipedia.org/wiki/Quay_lui_(khoa_h%E1%BB%8Dc_m%C3%A1y_t%C3%ADnh)) về đỉnh vừa mới tìm kiếm ở bước trước. Trong dạng không đệ quy, tất cả các đỉnh chờ được phát triển được bổ sung vào một [ngăn xếp](https://vi.wikipedia.org/wiki/Ng%C4%83n_x%E1%BA%BFp) [LIFO](https://vi.wikipedia.org/w/index.php?title=LIFO&action=edit&redlink=1).

Ý tưởng giải quyết thuật toán: từ một đỉnh S ban đầu ta sẽ có các đỉnh kề là x, từ đỉnh x ta sẽ có các đỉnh kề là y, và nó cũng thuộc nhánh s-x-y… Chúng ta thăm các nhánh đó theo chiều sâu (thăm đến khi không còn đỉnh kề chưa duyệt).

Cách thực hiện:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Quy trình duyệt Depth First Search | | | | | | | | | | | | | | | | | | | | | | | | | |
| Thông tin đầu vào  Source link: 0  Destination link: 9 | Stack node | | | | | | | | | | | | | | | | | | | | | | | | |
| Link 0 đến được link 1 2 3 | 0,1 | 1,2 | 2,6 | | 1,4 | | 4,8 | | 1,5 | | 5,10 | | 5,11 | | 0,2 | | 2,6 | | 0,3 | | 3,7 | | 7,4 | | 7,9 |
| Link 1 đến được link 2 4 5 | Path (đã duyệt) | | | | | | | | | | | | | | | | | | | | | | | | |
| Link 2 đến được link 0 6 | 0,1 | 1,2 | | 2,6 | | 1,4 | | 4,8 | | 1,5 | | 5,10 | | 5,11 | | 0,2 | | 2,6 | | 0,3 | | 3,7 | | 3,9 | |
| Link 3 đến được link 7 | Kết quả | | | | | | | | | | | | | | | | | | | | | | | | |
| Link 4 đến được link 0 8 | 0->3  3->7  7->9 | | | | | | | | | | | | | | | | | | | | | | | | |
| Link 5 đến được link 10 11 |
| Link 6 |
| Link 7 đến được link 4 9 |

Bảng 1: Cách thực hiện thuật toán depth first search

2.2. Thuật toán tìm kiếm theo chiều rộng (Breadth-First Search – BFS)

Thuật toán Breadth-First Search bắt đầu từ đỉnh gốc và lần lượt nhìn các đỉnh kề với đỉnh gốc. Sau đó, với mỗi đỉnh trong số đó, thuật toán lại lần lượt nhìn trước các đỉnh kề với nó mà chưa được quan sát trước đó và lặp lại.

Sử dụng thuật toán Breadth-First Search để tìm kiếm đường đi từ một đỉnh gốc cho trước tới một đỉnh đích, và tìm kiếm đường đi từ đỉnh gốc tới tất cả các đỉnh khác.

Ý tưởng giải quyết thuật toán: chúng ta sẽ xây dựng một danh sách chứa các đỉnh đang chờ duyệt, tại mỗi bước chúng ta thăm đỉnh ở đầu danh sách và thêm những đỉnh kề với nó chưa có trong danh sách chờ vào cuối danh sách.

Cách thực hiện:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Quy trình duyệt Breadth First Search | | | | | | | | | | | | | | | | | | | | | |
| Thông tin đầu vào  Source link: 0  Destination link: 9 | Struct node {prelink, curlink} | | | | | | | | | | | | | | | | | | | | |
| Link 0 đến được link 1 2 3 | 0,1 | 0,2 | | 0,3 | | 1,2 | 1,4 | | 1,5 | | 2,6 | 3,7 | | 4,8 | | 5,10 | 5,11 | | 7,4 | | 7,9 |
| Link 1 đến được link 2 4 5 | Path (đã duyệt) | | | | | | | | | | | | | | | | | | | | |
| Link 2 đến được link 0 6 | 0,1 | | 0,2 | | 0,3 | | | 1,2 | | 1,4 | | | 1,5 | | 2,6 | | | 3,7 | | 7,9 | |
| Link 3 đến được link 7 | Kết quả | | | | | | | | | | | | | | | | | | | | |
| Link 4 đến được link 0 8 | 0->3  3->7  7->9 | | | | | | | | | | | | | | | | | | | | |
| Link 5 đến được link 10 11 |
| Link 6 |
| Link 7 đến được link 4 9 |

Bảng 2: Cách thực hiện thuật toán breadth first search

2.3. Tìm kiếm hai chiều (Bidirectional Search – BDS)

Tìm kiếm hai chiều là một thuật toán tìm kiếm đồ thị tìm nguồn nguồn dạng đường dẫn nhỏ nhất tới đỉnh mục tiêu. Nó chạy hai tìm kiếm đồng thời:

* Chuyển tiếp nguồn mẫu tìm kiếm / đỉnh ban đầu về phía đỉnh mục tiêu
* Tìm kiếm ngược mục tiêu / đỉnh mục tiêu đối với đỉnh nguồn

Ý tưởng giải quyết thuật toán: chúng ta sẽ bắt đầu từ hai chiều (đầu và cuối), sử dụng thuật toán BFS với mỗi đầu khi giao nhau tại một nút trung gian thì kết thúc thuật toán hoặc không tìm thấy nút trung gian thì cũng kết thúc.

Cách thực hiện:

Định dạng: Node {prelink, current link}, TempNode {current link, {next link}}

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Quy trình duyệt Bidirectional Search | | | | | | | | | |
| Thông tin đầu vào  Source link: 1  Destination link: 8 | Forward Queue | | | | | | | | |
| Link 1 đến được link 2 3 4 | 1,2 | 1,3 | 1,4 | 2,4 | 2,5 | 3,4 | 3,6 | 4,7 | 5,8 |
| Link 2 đến được link 4 5 | Backward Queue | | | | | | | | |
| Link 3 đến được link 4 6 | 8,5 | | | 5,2 | | | 2,1 | | |
| Link 4 đến được link 7 | Kết quả | | | | | | | | |
| Link 5 đến được link 8 | 1->2  2->5  5->8 | | | | | | | | |
| Link 6 |
| Link 7 |

Bảng 3: Cách thực hiện thuật toán bidirectional search

**3. KẾT QUẢ**

Một số kết quả tìm kiếm đường đi của các thuật toán như sau:

* Thuật toán Depth-First Search

|  |  |  |
| --- | --- | --- |
| Nguồn | Đích đến | Kết quả tìm kiếm |
| 3.html | 94.html | 3.html->5.html  5.html->4.html  4.html->12.html  12.html->94.html |
| 12.html | 36.html | Không có đường đi |
| 80.html | 2.html | 80.html->8.html  8.html->5.html  5.html->4.html  4.html->12.html  12.html->7.html  7.html->2.html |

Bảng 4: Kết quả tìm kiếm DFS

* Thuật toán Breadth-First Search

|  |  |  |
| --- | --- | --- |
| Nguồn | Đích đến | Kết quả tìm kiếm |
| 3.html | 94.html | 3.html->5.html  5.html->14.html  14.html->94.html |
| 12.html | 36.html | Không có đường đi |
| 80.html | 2.html | 80.html->90.html  90.html->14.html  14.html->2.html |

Bảng 5: Kết quả tìm kiếm BFS

* Thuật toán Bidirectional Search

|  |  |  |
| --- | --- | --- |
| Nguồn | Đích đến | Kết quả tìm kiếm |
| 3.html | 94.html | 3.html->5.html  5.html->4.html  4.html->12.html  12.html->94.html  Gặp nhau tại: 4.html |
| 12.html | 36.html | Không có đường đi |
| 80.html | 2.html | 80.html->90.html  90.html->5.html  5.html->14.html  14.html->2.html  Gặp nhau tại: 5.html |

Bảng 6: Kết quả tìm kiếm BDS

**4. KẾT LUẬN**

Sau quá trình thực hiện “Viết chương trình nhập vào 2 đường dẫn URL của 2 trang web và tìm đường đi (danh sách các URL) từ một trang đến trang còn lại” nhóm đã thực hiện được thuật toán Breadth-First Search, Depth-First Search, Bidirectional Search.

Nếu hệ số phân nhánh của cây là b và khoảng cách của đỉnh mục tiêu từ nguồn là d, thì độ phức tạp tìm kiếm BFS / DFS bình thường sẽ là . Với BDS, chúng ta thực hiện hai thao tác tìm kiếm thì độ phức tạp sẽ là  cho mỗi tìm kiếm và tổng độ phức tạp sẽ  ít hơn nhiều .

Trong nghiên cứu này nhóm đã giải quyết được vấn đề tìm đường đi từ một trang đến một trang còn lại. Thuật toán đơn giản, dễ dàng thực hiện. Nhưng code còn chưa được tối ưu. Trong tương lai nhóm sẽ thực hiện thêm một số thuật toán có thời gian giải quyết toán hơn và tối ưu code.

**TÀI LIỆU THAM KHẢO**

1. Thuật toán breadth first search: <https://en.wikipedia.org/wiki/Breadth-first_search>
2. Thuật toán depth first search: <https://en.wikipedia.org/wiki/Depth-first_search>
3. Thuật toán bidirectional search: <https://www.geeksforgeeks.org/bidirectional-search/>
4. <https://vi.wikipedia.org/wiki/Qu%E1%BA%A3ng_c%C3%A1o_tr%E1%BB%B1c_tuy%E1%BA%BFn>

Thông tin nhóm chịu trách nhiệm bài viết  
1. Họ tên: Nguyễn Thanh Lập  
Mã số sinh viên: 17110169  
Email: [17110169@student.hcmute.edu.vn](mailto:17110169@student.hcmute.edu.vn)

2. Họ tên: Luyện Ngọc Thanh  
Mã số sinh viên: 17110221  
Email: [17110221@student.hcmute.edu.vn](mailto:17110221@student.hcmute.edu.vn)

3. Họ tên: Ôn Đức Khang  
Mã số sinh viên: 17110310  
Email: [17110310@student.hcmute.edu.vn](mailto:17110310@student.hcmute.edu.vn)