TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO GIỮA KỲ**

**PHÂN TÍCH THIẾT KẾ VÀ GIẢI THUẬT ĐỀ 1**

*Người hướng dẫn*: **TS. Trần Lương Quốc Đại**

*Người thực hiện*: **Ngô Phan Minh Trí – 52200171**

**Châu Nguyễn Khánh Trình – 52200005**

**Lê Như Đạt - 52200160**

Lớp**: 22050301**

Khoá**: 26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

A red and blue logo

Description automatically generated

**BÁO CÁO GIỮA KỲ**

**PHÂN TÍCH THIẾT KẾ VÀ GIẢI THUẬT ĐỀ 1**

*Người hướng dẫn*: **TS. Trần Lương Quốc Đại**

*Người thực hiện*: **Ngô Phan Minh Trí – 52200171**

**Châu Nguyễn Khánh Trình – 52200005**

**Lê Như Đạt - 52200160**

Lớp**: 22050301**

Khoá**: 26**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2024**

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến thầy Đại vì đã tận tình hướng dẫn, truyền đạt kiến thức và dành thời gian quý báu hỗ trợ chúng em trong suốt quá trình thực hiện bài báo cáo này. Những góp ý và chỉ dẫn của thầy đã giúp chúng em hiểu sâu hơn về đề tài, hoàn thiện kỹ năng nghiên cứu và trình bày. Chúng em xin trân trọng cảm ơn thầy và kính chúc thầy luôn mạnh khỏe, thành công trong sự nghiệp trồng người.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH

TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS. Trần Lương Quốc Đại. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong luận văn còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung luận văn của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày tháng năm*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Châu Nguyễn Khánh Trình*

*Ngô Phan Minh Trí*

*Lê Như Đạt*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày 16 tháng 11 năm 2024

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày 16 tháng 11 năm 2024

TÓM TẮT

Bài báo cáo được hoàn tất 100% với các những vấn đề cần được giải quyết là Coinrow Problem và The Maximum-Flow Problem. Giải quyết các vấn đề trong bài báo cáo bằng những tư duy logic và những gì đã học trong môn phân tích thiết kế và giải thuật.

Nội dung

[LỜI CẢM ƠN i](#_Toc182648879)

[ĐỒ ÁN ĐƯỢC HOÀN THÀNH i](#_Toc182648880)

[TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG i](#_Toc182648881)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN ii](#_Toc182648882)

[TÓM TẮT iii](#_Toc182648883)

[DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT 3](#_Toc182648884)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 4](#_Toc182648885)

[CHƯƠNG 1: COIN-ROW PROBLEM 5](#_Toc182648886)

[1.1 Cơ sở lý thuyết 5](#_Toc182648887)

[1.1.1 Mô tả bài toán 5](#_Toc182648888)

[1.2 Phương pháp giải 5](#_Toc182648889)

[1.3 Example 6](#_Toc182648890)

[1.3.1 Example 1 6](#_Toc182648891)

[1.3.2 Example 2 7](#_Toc182648892)

[1.3.3 Example 3 7](#_Toc182648893)

[1.4 Trình bày vả so sánh kết quả 8](#_Toc182648894)

[1.4.1 Trình bày 8](#_Toc182648895)

[1.4.2 So sánh 9](#_Toc182648896)

[CHƯƠNG 2: THE MAXIMUM-FLOW PROBLEM 11](#_Toc182648897)

[2.1 Cơ sở lý thuyết 11](#_Toc182648898)

[2.1.1 Định nghĩa 11](#_Toc182648899)

[2.1.2 Tính chất 11](#_Toc182648900)

[2.1.3 Phát biểu 12](#_Toc182648901)

[2.2 Phương pháp giải 12](#_Toc182648902)

[2.2.1 Thuật toán Edmonds-Karp 12](#_Toc182648903)

[2.2.2 Network cuts 14](#_Toc182648904)

[2.3 Example 15](#_Toc182648905)

[2.3.1 Example 1 15](#_Toc182648906)

[2.3.1 Example 2 16](#_Toc182648907)

[2.3.1 Example 3 18](#_Toc182648908)

[2.4 Trình bày, so sánh và phân tích kết quả 19](#_Toc182648909)

[2.4.1 Trình bày 19](#_Toc182648910)

[2.4.2 So sánh 20](#_Toc182648911)

[2.4.3 Phân tích kết quả 21](#_Toc182648912)

[TÀI LIỆU THAM KHẢO 23](#_Toc182648913)

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

**CÁC KÝ HIỆU**

*f Tần số của dòng điện và điện áp (Hz)*

*p Mật độ điện tích khối (C/m3)*

**CÁC CHỮ VIẾT TẮT**

CSTD Công suất tác dụng

MF Máy phát điện

BER Tỷ lệ bít lỗi

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

[Hình 1. Kết quả chạy thực nghiệm 5 trường hợp 9](#_Toc182663751)

[Hình 2: Hình ảnh minh họa kết quả thuật toán 14](#_Toc182663752)

[Hình 3. Đồ thị của ví dụ 1 15](#_Toc182663753)

[Hình 4. Lần chạy đầu tiên trong ví dụ 16](#_Toc182663754)

[Hình 5. Kết quả của ví dụ 1 16](#_Toc182663755)

[Hình 6. Đồ thị của ví dụ 2 17](#_Toc182663756)

[Hình 7. Lần chạy đầu tiên của ví dụ 2 17](#_Toc182663757)

[Hình 8. Kết quả của ví dụ 2 18](#_Toc182663758)

[Hình 9. Đồ thị của ví dụ 3 18](#_Toc182663759)

[Hình 10. Kết quả của ví dụ 3 19](#_Toc182663760)

[Hình 11. Kết quả khi chạy trên máy của ví dụ 1 21](#_Toc182663761)

[Hình 12. Kết quả khi chạy trên máy của ví dụ 2 21](#_Toc182663762)

[Hình 13. Kết quả khi chạy trên máy của ví dụ 3 21](#_Toc182663763)

CHƯƠNG 1: COIN-ROW PROBLEM

1.1 Cơ sở lý thuyết

1.1.1 Mô tả bài toán

Bài toán yêu cầu như sau:

- Đầu vào:

* Một tập gồm **n** đồng xu, với giá trị lần lượt là c1, c2,…, cn.
* Các giá trị này là số nguyên dương và không nhất thiết phải giống nhau.

- Đầu ra:

* Tổng giá trị lớn nhất có thể thu được khi chọn các đồng xu, **đảm bảo rằng không chọn hai đồng xu liền kề** trong hàng (Nếu chọn đồng xu thứ i, bạn không được chọn đồng xu thứ i – 1 hoặc i + 1).
* Ví dụ:

Giả sử dãy đồng xu có giá trị như sau: {5, 1, 2, 10, 6, 2}

Bạn không được chọn 5 và 1 cùng lúc vì chúng nằm cạnh nhau. Tương tự, bạn không được chọn 10 và 6 cùng lúc.

**Mục tiêu:** Chọn các đồng xu sao cho tổng giá trị thu được là lớn nhất.

1.2 Phương pháp giải

- Bài toán có thể giải quyết bằng **kỹ thuật** **quy hoạch động (Dynamic Programming).** Đây là một kỹ thuật giải bài toán tối ưu bằng cách chia nhỏ bài toán lớn thành các bài toán con nhỏ hơn, giải quyết chúng một lần, và lưu trữ kết quả để tránh tính toán lại. Ví dụ: Bài toán Fibonacci, Knapsack Problem,…).

- Phương pháp giải:

Gọi **F(n)** là tổng giá trị lớn nhất có thể thu được khi chọn các đồng xu từ dãy gồm n đồng xu đầu tiên, với điều kiện không chọn 2 đồng xu liền kề.

Khi xét đồng xu thứ **n** (đồng xu cuối cùng), ta có hai lựa chọn:

* **Không chọn đồng xu thứ n**: Trong trường hợp này, tổng giá trị lớn nhất khi xét n đồng xu bằng tổng giá trị lớn nhất khi xét n − 1 đồng xu.

**F(n) = F(n - 1)**

* **Chọn đồng xu thứ n**: Khi chọn đồng xu thứ n, không chọn đồng xu thứ n - 1. Khi đó, giá trị tối ưu là giá trị của đồng xu n (cn) cộng với giá trị tối ưu của dãy n − 2 đồng xu trước đó.

**F(n) = cn + F(n - 2)**

* Kết hợp hai trường hợp:

**F(n) = max{F(n − 1), cn ​+F(n − 2)}, n > 1**

**F(0) = 0: Nếu không có đồng xu nào, giá trị lớn nhất là 0.**

**F(1) = c1: Nếu chỉ có một đồng xu, giá trị lớn nhất là giá trị của đồng xu đó.**

1.3 Example

1.3.1 Example 1

- Dãy đồng xu có giá trị {5, 1, 2, 10, 6, 2}

* F(0) = 0
* F(1) = c1 = 5
* F(2) = max{F(1), c2 ​+F(0)} = max{5, 1 + 0} = 5
* F(3) = max{F(2), c3 ​+F(1)} = max{5, 2 + 5} = 7
* F(4) = max{F(3), c4 ​+F(2)} = max{7, 10 + 5} = 15
* F(5) = max{F(4), c5 ​+F(3)} = max{15, 6 + 7} = 15
* F(6) = max{F(5), c6 ​+F(4)} = max{15, 2 + 15} = 17

**Vậy tổng giá trị tối đa có thể thu được là 17**

|  |  |  |
| --- | --- | --- |
| **n** | **F(n)** | **Diễn giải** |
| 0 | 0 | Không có đồng xu |
| 1 | 5 | Chọn đồng xu c1 = 5 |
| 2 | 5 | Không chọn c2 = 1, giữ F(1) = 5 |
| 3 | 7 | Chọn c3 = 2, cộng với F(1) = 5 |
| 4 | 15 | Chọn c4 = 10, cộng với F(2) = 5 |
| 5 | 15 | Không chọn c5 = 6, giữ F(4) = 15 |
| 6 | 17 | Chọn c6 = 2, cộng với F(4) = 15 |

1.3.2 Example 2

- Dãy đồng xu có giá trị {4, 12, 7}

* F(0) = 0
* F(1) = c1 = 4
* F(2) = max{F(1), c2 ​+F(0)} = max{4, 12 + 0} = 12
* F(3) = max{F(2), c3 ​+F(1)} = max{12, 7 + 4} = 12

**Vậy tổng giá trị tối đa có thể thu được là 12**

|  |  |  |
| --- | --- | --- |
| **n** | **F(n)** | **Diễn giải** |
| 0 | 0 | Không có đồng xu |
| 1 | 4 | Chọn đồng xu c1 = 4 |
| 2 | 12 | Chọn c2 = 12, bỏ qua F(1) = 4 |
| 3 | 12 | Không chọn c3 = 7, giữ F(2) = 12 |

1.3.3 Example 3

**-** Dãy đồng xu có giá trị {3, 3, 3, 3, 3}

* F(0) = 0
* F(1) = c1 = 3
* F(2) = max{F(1), c2 ​+F(0)} = max{3, 3 + 0} = 3
* F(3) = max{F(2), c3 ​+F(1)} = max{3, 3 + 3} = 6
* F(4) = max{F(3), c4 ​+F(2)} = max{6, 3 + 3} = 6
* F(5) = max{F(4), c5 ​+F(3)} = max{6, 3 + 6} = 9

**Vậy tổng giá trị tối đa có thể thu được là 9**

|  |  |  |
| --- | --- | --- |
| **n** | **F(n)** | **Diễn giải** |
| 0 | 0 | Không có đồng xu |
| 1 | 3 | Chọn đồng xu c1 = 3 |
| 2 | 3 | Không chọn c2 = 3, giữ F(1) = 3 |
| 3 | 6 | Chọn c3 = 3, cộng với F(1) = 3 |
| 4 | 6 | Không chọn c4 = 10, giữ F(3) = 6 |
| 5 | 9 | Chọn c5 = 3, cộng với F(3) = 6 |

1.4 Trình bày, so sánh và phân tích kết quả

1.4.1 Trình bày

- Function coin\_row():

* Input: Tập hợp các đồng xu
* Output: Giá trị tối ưu F(n)

Thuật toán chạy như trình bày ở phần cơ sở lý thuyết

- Function TruyXuat():

* Input: Tập hợp các đồng xu
* Out put: Tập hợp các đồng xu được chọn
  + Đầu tiên, hàm sẽ chạy lại thuật toán CoinRow để có được tập F
  + Hàm sẽ duyệt qua tập hợp bằng cách chạy từ cuối tập hợp chạy về đầu
  + Nếu giá trị đang xét F[i] = giá trị F[i-1] thì hàm sẽ duyệt phần tử tiếp theo
  + Khi nào hàm thấy giá trị F bị thay đổi thì hàm sẽ lấy đồng xu ở vị trí đó và index – 2(Vì chọn đồng xu không kề nhau).
  + Khi index = 0 dừng vòng lặp.

1.4.2 So sánh và phân tích kết quả

* Kết quả:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Hình 1. Kết quả chạy thực nghiệm 5 trường hợp

* Trường hợp 1: Dãy không có đồng xu nào, giá trị tối ưu của dãy là 0
* Trường hợp 2: Dãy có 1 đồng xu, giá trị của đồng xu là giá trị tối ưu của dãy.
* Trường hợp 3: Có giá trị lớn xen kẽ, thuật toán chọn được 100 là lớn nhất.
* Trường hợp 4: Thuật toán chạy từng bước để có thể chạy ra đáp án cuối cùng.
* Trường hợp 5: Giá trị tăng dần, thuật toán chọn các giá trị cách đều 10, 30, 50.

CHƯƠNG 2: THE MAXIMUM-FLOW PROBLEM

2.1 Cơ sở lý thuyết

2.1.1 Định nghĩa

* Giả sử ta có một mạng lưới để vận chuyển một loại vật chất (như nước trong hệ thống ống dẫn, hoặc dòng điện trong một hệ thống điện). Mục tiêu của ta là đẩy càng nhiều vật chất qua mạng lưới này từ một điểm xuất phát (gọi là *source*) đến điểm đích cuối cùng (gọi là *sink*).
  + **Source**: Đây là nơi vật chất bắt đầu di chuyển vào mạng lưới, không có gì "chảy vào" nguồn.
  + **Sink**: Đây là nơi vật chất thoát ra khỏi mạng lưới, không có gì "chảy ra" từ đích.
  + **Edge capacity**: Mỗi cạnh có hướng với trọng số là một số nguyên dương, được gọi là dung lượng cạnh.

2.1.2 Tính chất

* **Dung lượng cạnh**: Mỗi đoạn đường (cạnh) nối giữa hai điểm trong mạng lưới đều có một giới hạn dung lượng – tức là giới hạn về lượng vật chất tối đa có thể di chuyển qua đó.
* **Yêu cầu bảo toàn dòng chảy**: Tại mỗi điểm trung gian, tổng lượng vật chất đi vào phải bằng tổng lượng đi ra.

A close up of a word

Description automatically generated

* **Value(v)**: Tổng lượng vật liệu đi ra khỏi nguồn phải đi đến bồn chứa.

A math equation with a triangle and a triangle

Description automatically generated with medium confidence

* Luồng (khả thi): là phép gán các số thực cho các cạnh (i, j) của một mạng nhất định thỏa mãn các ràng buộc **bảo toàn luồng** và các ràng buộc về **dung lượng.**



2.1.3 Phát biểu

* Bài toán maximum-flow problem có thể được phát biểu như sau:

A math equations and formulas

Description automatically generated with medium confidence

2.2 Phương pháp giải

2.2.1 Thuật toán Edmonds-Karp

* Edmonds-Karp là một biến thể của Ford-Fulkerson, trong đó **tìm kiếm theo chiều rộng (BFS)** được sử dụng để tìm đường đi tăng cường ngắn nhất (theo số cạnh). Thuật toán này khắc phục được vòng lặp vô hạn trong các đồ thị có cạnh với dung lượng không nguyên. Thuật toán này được gọi là **shortest-augmenting-path** hoặc **first-labeled-firstscanned algorithm.**
* Thuật toán này sử dụng việc đánh nhãn giúp tìm đường đi từ đỉnh nguồn (*source*) đến đỉnh đích (*sink*) trong đồ thị khả dụng (*residual graph*) bằng cách theo dõi khả năng luồng còn có thể truyền qua từng đỉnh. Quy trình cụ thể như sau:
  + **Ý nghĩa của nhãn (label)**: Khi một đỉnh mới (*unlabeled vertex*) được đánh nhãn, nó sẽ có hai thông tin:
    - **Nhãn đầu tiên** cho biết lượng luồng tối đa có thể truyền từ nguồn đến đỉnh này.
    - **Nhãn thứ hai** là tên của đỉnh liền trước đã dẫn đến đỉnh này (có thể bỏ qua với đỉnh nguồn). Để dễ nhận biết hướng của cạnh, ký hiệu “+” sẽ được thêm vào nếu đỉnh được đến qua cạnh thuận (forward edge) và “−” nếu đi qua cạnh ngược (backward edge).
* **Cách đánh nhãn đỉnh nguồn**: Đỉnh nguồn luôn được đánh nhãn là ∞,- (vô cực, không có đỉnh trước).
* **Cách đánh nhãn các đỉnh khác**:
  + Nếu một đỉnh chưa có nhãn (*vertex j*) được kết nối với đỉnh đầu của hàng đợi (*vertex i*) qua một cạnh thuận từ *i* đến *j* có dung lượng khả dụng (*residual capacity*) dương , thì nhãn của *j* sẽ là:
    - Nhãn đầu tiên: , với là luồng tối đa có thể đi qua đỉnh *i*.
    - Nhãn thứ hai: , cho biết rằng j được đến từ *i* qua cạnh thuận.
  + Nếu một đỉnh chưa có nhãn (*vertex j*) được kết nối với đỉnh đầu của hàng đợi (*vertex i*) qua một cạnh ngược từ *j* đến *i* có luồng dương ​, thì nhãn của *j* sẽ là:
    - Nhãn đầu tiên:
    - Nhãn thứ hai: , cho biết rằng *j* được đến từ *i* qua cạnh ngược.
* Nếu quy trình đánh nhãn này dẫn đến việc đỉnh đích (*sink*) cũng được đánh nhãn, có nghĩa là có một đường đi tăng cường (*augmenting path*) từ nguồn đến đích. Luồng có thể được tăng thêm một lượng bằng với giá trị của nhãn đầu tiên của đỉnh đích.
* Việc tăng cường luồng này thực hiện bằng cách đi ngược lại từ đỉnh đích đến đỉnh nguồn theo các nhãn thứ hai, để xác định đường đi tăng cường. Luồng được tăng lên trên các cạnh thuận và giảm đi trên các cạnh ngược.
* Nếu hàng đợi hết mà đỉnh đích không có nhãn, nghĩa là không còn đường đi nào có thể truyền thêm luồng từ nguồn đến đích. Lúc này, thuật toán kết thúc và trả về luồng hiện tại là luồng cực đại.

A diagram of lines and numbers

Description automatically generated with medium confidence

Hình 2: Hình ảnh minh họa kết quả thuật toán

2.2.2 Network cuts

* **Cut**: Một phép cắt được tạo ra bằng cách phân vùng các đỉnh của mạng thành một tập con X chứa nguồn và X, X¯ là phần bù của X. Chứa phần chìm là tập hợp tất cả các cạnh có đuôi trong X và đầu trong X¯, tập hợp này được gọi là C(X, X¯), sau khi bỏ đi các cạnh của đồ thị trong tập C thì sẽ không còn đường đi nào từ nguồn đến đích trong đồ thị.
* **Minimum cut**: do tồn tại rất nhiều **cut** nên luôn tồn tại một vết cắt tối thiểu, tức là một vết cắt có khả năng nhỏ nhất.
* **Max-Flow Min-Cut Theorem**: The value of a maximum flow in a network is equal to the capacity of its minimum cut (giá trị của luồng cực đại trong mạng bằng với khả năng cắt cực tiểu của nó).

2.3 Example

2.3.1 Example 1

A graph with arrows and points

Description automatically generated

Hình 3. Đồ thị của ví dụ 1

* Gán mọi cạnh ij = 0, tạo label cho source = inf, ‘-’, dùng bfs để lan ra và tạo label cho những đỉnh thỏa điều kiện **giới hạn dung lượng**
* Label: 1:[inf, '-'], 2: [2, '1+'], 3: [2, '2+'], 4: [3, '1+'], 5: [2, '2+'], 6: [2, '3+']
* Khi label của sink đã được gán nhãn thì ta sẽ gán giá trị ngược về từ sink, phải đảm bảo được điều kiện **bảo toàn luồng**

A graph with arrows and points

Description automatically generated

Hình 4. Lần chạy đầu tiên trong ví dụ 1

* Xóa tất cả nhãn trừ nhãn source và tiếp tục quá trình gán nhãn
* Label: 1: [inf, '-'], 2: [1, '3-'], 3: [1, '4+'], 4: [3, '1+'], 5: [1, '2+'], 6: [1, '5+']
* Gán giá trị ngược về.

A graph with arrows and points

Description automatically generated

Hình 5. Kết quả của ví dụ 1

* Kết thúc quá trình do đã đảm bảo được tất cả **giới hạn dung lượng**

2.3.1 Example 2

A graph with numbers and arrows

Description automatically generated

Hình 6. Đồ thị của ví dụ 2

* Gán mọi cạnh ij = 0, tạo label cho source = inf, ‘-’, dùng bfs để lan ra và tạo label cho những đỉnh thỏa điều kiện **giới hạn dung lượng**
* Label: 1: [inf, '-'], 2: [7, '1+'], 3: [6, '1+'], 4: [4, '2+'], 5: [2, '2+'], 6: [4, '4+']
* Khi label của sink đã được gán nhãn thì ta sẽ gán giá trị ngược về từ
* sink, phải đảm bảo được điều kiện **bảo toàn luồng**

A graph with numbers and lines

Description automatically generated

Hình 7. Lần chạy đầu tiên của ví dụ 2

* Lặp lại quá trình này, các label tiếp theo
* 1: [inf, '-'], 2: [3, '1+'], 3: [6, '1+'], 4: [2, '3+'], 5: [2, '2+'], 6: [2, '5+']
* 1: [inf, '-'], 2: [1, '1+'], 3: [6, '1+'], 4: [2, '3+'], 5: [3, '3+'], 6: [2, '4+']
* 1: [inf, '-'], 2: [1, '1+'], 3: [4, '1+'], 4: 0, 5: [3, '3+'], 6: [3, '5+']
* Kết quả cuối

A graph with arrows and numbers

Description automatically generated

Hình 8. Kết quả của ví dụ 2

2.3.1 Example 3

A diagram of a graph

Description automatically generated with medium confidence

Hình 9. Đồ thị của ví dụ 3

* Gán mọi cạnh ij = 0, tạo label cho source = inf, ‘-’, dùng bfs để lan ra và tạo label cho những đỉnh thỏa điều kiện **giới hạn dung lượng**
* Label: 1: [inf, '-'], 2: [2, '1+'], 3: [8, '1+'], 4: [2, '2+'], 5: [2, '2+'], 6: [2, '4+']
* Khi label của sink đã được gán nhãn thì ta sẽ gán giá trị ngược về từ sink, phải đảm bảo được điều kiện **bảo toàn luồng**
* Lặp lại quá trình và đây là quá trình cuối cùng

A diagram of a triangle with arrows

Description automatically generated

Hình 10. Kết quả của ví dụ 3

2.4 Trình bày, so sánh và phân tích kết quả

2.4.1 Trình bày

* Function shortest\_augmenting\_path():
  + Input: thông tin đồ thị
  + Output: x
    - x: đồ thị dưới dạng danh sách kề chứa maximum-flow
  + Mục đích: tìm ra đồ thị maximum-flow
* Function find\_maximum\_flow():
  + Input: x
  + Output: maximum-flow
  + Mục đích: tìm maximum-flow của đồ thị
* Function find\_minimum\_cut():
  + Input: maximum, x
  + Output: cut\_edges
  + Mục đích: tìm minimum-cut
* Function remove\_cut\_edges():
  + Input: graph, cut\_edges
    - graph: đồ thị gốc
    - cut\_edges: các trường hợp trong cut\_edges
  + Output: new\_graph
    - new\_graph: đồ thị mới đã loại bỏ các cut\_edges
  + Mục đích: trả về đồ thị đã xóa đi cut\_edges
* Function bfs():
  + Input: graph, source, sink
    - graph: đồ thị từ kết quả trả về của remove\_cut\_edges()
    - source: nguồn
    - sink: đích
  + Output: True or False
    - True: có đường đi từ source đến sink
    - False: không có đường đi từ source đến sink
  + Mục đích: kiểm tra xem đồ thị mới có đáp ứng định nghĩa minimum-cut không

2.4.2 So sánh

* Example 1

A computer screen with white text

Description automatically generated

Hình 11. Kết quả khi chạy trên máy của ví dụ 1

* Example 2

A black screen with numbers and symbols

Description automatically generated

Hình 12. Kết quả khi chạy trên máy của ví dụ 2

* Example 3

A computer screen with white text

Description automatically generated

Hình 13. Kết quả khi chạy trên máy của ví dụ 3

* Code hoạt động tốt trong các trượng hợp khác nhau.

2.4.3 Phân tích kết quả

* Kết luận:
  + Dựa trên kết quả chạy tay, ta thấy:
    - Example 1: có label chứa dấu ‘-’, chứng tỏ trong example 1 có backward edges.
    - Example 2, 3: tất cả nhãn đều dấu ‘+’.
  + Dựa vào kết quả code, ta thấy: khi ta dùng shortest-augmenting-path để tìm minimum-cut thì:
    - Example 1,3: chỉ có 1 cut phù hợp là minimum-cut.
    - Example 2: có 2 cut phù hợp là minimum-cut.

TÀI LIỆU THAM KHẢO

**Tiếng Anh**

1. [1] A. Levitin. Introduction to the Design and Analysis of Algorithms, Addison Wesley, 3rd edition, 2011
2. Algorithm Design and Applications[A4]