**Report**

**Project: Combinatorial Optimization**

Member:

1. Võ Văn Toàn 51704114
2. Bùi Long Giang 51704039
3. Nguyễn Hữu Nghĩa 51704047

Group: 04-02

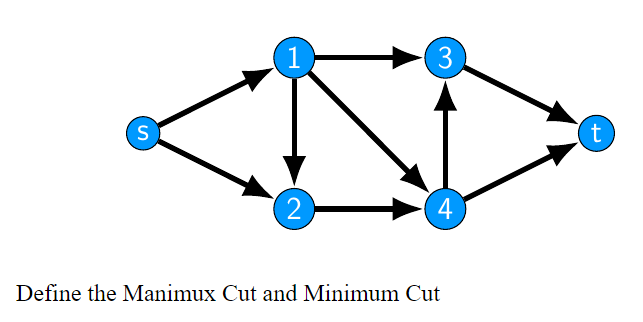
**Maximum Network Flow**

1. **Giới Thiệu**

* Vấn đề dòng chảy tối đa được hình thành lần đầu tiên vào năm 1954 bởi T. E. Harris và F. S. Ross như là một mô hình đơn giản hóa của lưu lượng giao thông đường sắt Liên Xô. Năm 1955, Lester R. Ford, Jr. và Delbert R. Fulkerson đã tạo ra thuật toán được biết đến đầu tiên, thuật toán Ford- Fulkerson.
* Trong những năm qua, nhiều giải pháp cải tiến khác nhau cho vấn đề dòng chảy tối đa đã được phát hiện, đáng chú ý là thuật toán đường dẫn tăng ngắn nhất của Edmonds và Karp và Dinitz độc lập; thuật toán chặn dòng của Dinitz; thuật toán đẩy nhãn của Goldberg và Tarjan; và thuật toán dòng chặn nhị phân của Goldberg và Rao. Các thuật toán của Sherman và Kelner, Lee, Orecchia và Sidford, tương ứng, tìm thấy một luồng tối đa xấp xỉ tối ưu nhưng chỉ hoạt động trong các đồ thị vô hướng.

1. **Bài Tập**

* Input: a connected digraph G=(V,E) and edge capacity.
* Task: find a feasible s-t flow of maximum value



1. **Thí Nghiệm Và Kết Qủa**

* The code and results in Github: <https://github.com/rongnuoc/Giang/blob/master/MaximumNetworkFlow.java>

1. **Kết Luận**

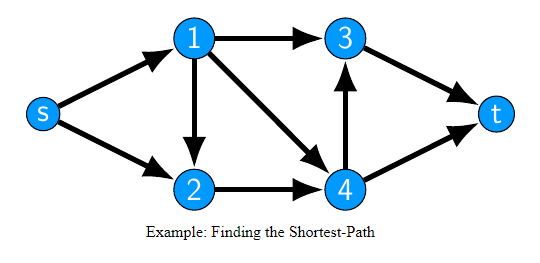
* **Ứng dụng thực tế :**
  + **Loại bỏ bóng chày:** Trong bài toán loại bỏ bóng chày có n đội thi đấu trong một giải đấu. Ở một giai đoạn cụ thể của mùa giải, wi là số trận thắng và ri là số trận còn lại để chơi cho đội i và rij là số trận còn lại trước đội j. Một đội bị loại nếu không có cơ hội kết thúc mùa giải ngay từ đầu. Nhiệm vụ của vấn đề loại bỏ bóng chày là xác định đội nào bị loại ở mỗi thời điểm trong mùa giải. Schwartz đã đề xuất một phương pháp làm giảm vấn đề này đến lưu lượng mạng tối đa. Trong phương pháp này, một mạng được tạo để xác định xem đội k có bị loại hay không.
  + **Lập lịch trình hàng không:** Trong ngành hàng không, một vấn đề lớn là lịch trình của các phi hành đoàn. Vấn đề lập lịch trình của hãng hàng không có thể được coi là một ứng dụng của lưu lượng mạng tối đa mở rộng. Đầu vào của vấn đề này là một tập hợp các chuyến bay F chứa thông tin về nơi và thời gian mỗi chuyến bay khởi hành và đến. Trong một phiên bản của lịch trình hãng hàng không, mục tiêu là tạo ra một lịch trình khả thi với hầu hết các phi hành đoàn.

**Shortest Path**

1. **Giới Thiệu**

* Vấn đề đường đi ngắn nhất là vấn đề tìm đường đi ngắn nhất hoặc tuyến đường từ điểm bắt đầu đến đích cuối cùng.
* Nói chung, để biểu diễn bài toán đường đi ngắn nhất, chúng tôi sử dụng biểu đồ. Biểu đồ là một đối tượng trừu tượng toán học, chứa các tập hợp các đỉnh và cạnh. Các cạnh nối các cặp đỉnh. Dọc theo các cạnh của đồ thị có thể đi bộ bằng cách di chuyển từ đỉnh này sang đỉnh khác. Tùy thuộc vào việc người ta có thể đi dọc theo các cạnh của cả hai bên hay chỉ bởi một bên xác định xem biểu đồ là đồ thị có hướng hay đồ thị không có hướng.
* Ngoài ra, độ dài của các cạnh thường được gọi là trọng số và các trọng số thường được sử dụng để tính toán đường đi ngắn nhất từ ​​điểm này đến điểm khác. Ví dụ: để thể hiện bản đồ, chúng ta có thể sử dụng biểu đồ, trong đó các đỉnh biểu thị các thành phố và các cạnh thể hiện các tuyến đường kết nối các thành phố. Nếu các tuyến là một chiều thì biểu đồ sẽ được chỉ dẫn; nếu không, nó sẽ không bị ảnh hưởng. Tồn tại các loại thuật toán khác nhau giải quyết vấn đề đường đi ngắn nhất: Thuật toán Dijkstra, Thuật toán Floyd-Warshall, Thuật toán Bellman-Ford, Thuật toán di truyền

1. **Bài Tập**
2. Input: a connected (di-)graph G=(V,E) an edge valuation c:E->R, distinct nodes s,t  V s, t ϵ V
3. Task: find a shortest path connecting s and t in G with respect to c



1. **Thí Nghiệm Và Kết Qủa**
2. The code and results in Github: <https://github.com/rongnuoc/Giang/blob/master/ShortestPath.java>
3. **Kết Luận**
4. Độ phức tạp thời gian được tính toán cho từng thuật toán Dijkstra, cho thấy thuật toán có thể chấp nhận được về mặt hiệu suất tổng thể trong việc giải quyết vấn đề đường đi ngắn nhất.
5. Ngày nay, cũng có nhiều thuật toán đường dẫn ngắn nhất thông minh đã được giới thiệu trong một số tài liệu nghiên cứu trước đây

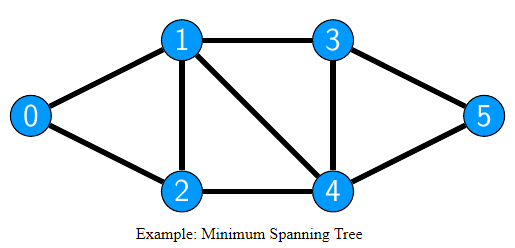
**Minimum Spanning Tree**

1. **Giới Thiệu**

* A **minimum spanning tree** (**MST**) hoặc cây bao trùm tối thiểu là tập hợp con của các cạnh của đồ thị vô hướng có trọng số cạnh được kết nối, kết nối tất cả các đỉnh với nhau, không có chu kỳ và với tổng trọng lượng tối thiểu có thể có. Đó là, nó là một cây bao trùm có tổng trọng lượng cạnh càng nhỏ càng tốt. Tổng quát hơn, bất kỳ đồ thị vô hướng có trọng số cạnh (không nhất thiết phải được kết nối) đều có một rừng bao trùm tối thiểu, đó là sự kết hợp của các cây bao trùm tối thiểu cho các thành phần được kết nối của nó.
* Nếu có n đỉnh trong biểu đồ, thì mỗi cây bao trùm có n - 1 cạnh. Có thể có một số cây bao trùm tối thiểu có cùng trọng lượng; đặc biệt, nếu tất cả các trọng số cạnh của đồ thị đã cho là như nhau, thì mọi cây bao trùm của đồ thị đó là tối thiểu.

1. **Bài Tập**

* Input: a connected graph G=(V, E) and an edge valuation c:E->R.
* Task: find a spanning tree T⸦ E of minimal total weight c(T)



1. **Thí Nghiệm Và Kết Qủa**

* The code and results in Github: <https://github.com/rongnuoc/Giang/blob/master/Minimumspanningtree.java>

1. **Kết Luận**

* Cây kéo dài tối thiểu có các ứng dụng trực tiếp trong thiết kế mạng, bao gồm mạng máy tính, mạng viễn thông, mạng giao thông, mạng cấp nước và lưới điện (lần đầu tiên chúng được phát minh ra, như đã đề cập ở trên). Chúng được gọi là chương trình con trong các thuật toán cho các vấn đề khác, bao gồm thuật toán Christofides để tính gần đúng bài toán nhân viên bán hàng du lịch, xấp xỉ bài toán cắt tối thiểu đa đầu cuối (tương đương trong trường hợp một đầu cuối cho bài toán luồng cực đại) và xấp xỉ kết hợp hoàn hảo với chi phí tối thiểu

**Tham khảo:**

<https://en.wikipedia.org/wiki/Maximum_flow_problem>

<https://en.wikipedia.org/wiki/Shortest_path_problem>

<https://en.wikipedia.org/wiki/Minimum_spanning_tree>