BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

LUẬN VĂN THẠC SĨ KHOA HỌC

CẢI TẠO MỞ RỘNG LƯỚI ĐIỆN TRUYỀN TẢI CÓ XÉT ĐẾN KHẢ NĂNG TẢI

NGÀNH: KỸ THUẬT ĐIỆN

SỐ HIỆU: 119

KHOÁ: 2005 – 2007

ĐINH TRỌNG HIẾU

Người hướng dẫn khoa học : PGS.TS NGUYỄN LÂN TRÁNG

HÀ NỘI, 2008

LỜI CẢM ƠN

Tôi xin chân trành cảm ơn TS.Nguyễn Lân Tráng đã có những gợi mở và dẫn dắt tận tình, cung cấp các tài liệu quý giá liên quan để tôi có thể hoàn thành được luận văn này. Tôi cũng xin chân thành cảm ơn các thầy cô giáo trong Bộ môn Hệ thống điện - trường Đại Học Bách Khoa Hà Nội và những người đã giúp đỡ tôi hoàn thành bản luận văn này.

Đây là luận văn đầu tiên, về là những vấn đề rất đáng quan tâm hiện nay, trong Ngành điện, tôi sẽ cố gắng tìm hiểu và phát triển thêm sau này để có thể vận dụng tốt điều kiện thực tế.

Cuối cùng, tác giả vô cùng biết ơn sự quan tâm, động viên của gia đình và bạn bè trong thời gian qua. Nhờ đó, tôi có thêm thời gian và nghị lực để hoàn thành luận văn của mình.

Tác giả luận văn

Đinh Trọng Hiếu

MỤC LỤC

MỞ ĐẦU

1. GIỚI THIỆU CHUNG

Quy hoạch lưới điện là một phần quan trọng của quy hoạch hệ thống năng lượng. Nhiệm vụ của nó là xác định một cấu hình tối ưu theo sự tăng trưởng của phụ tải và một sơ đồ quy hoạch nguồn đối với thời gian quy hoạch ứng với yêu cầu phân phối điện năng một cách an toàn và kinh tế. Hay nói cách khác việc quy hoạch lưới điện sẽ trả lời các câu hỏi sau:

1. Đặt đường dây truyền tải mới ở đâu ?
2. Khi nào xây dựng chúng ?
3. Kiểu của đường dây truyền tải dự định xây là loại gì ?

Quy hoạch lưới điện có quan hệ chặt chẽ với quy hoạch nguồn điện. Nó dựa trên cơ sở một sơ đồ quy hoạch nguồn nhưng lại có ảnh hưởng trở lại tới quy hoạch nguồn điện. Như đã biết, quy hoạch nguồn điện không duy trì hoặc chỉ duy trì một chút ảnh hưởng của phân bố địa lý và giá thành chuyên tải. Quy hoạch lưới điện có thể dùng để chỉnh lại sơ đồ quy hoạch nguồn điện ban đầu. Vì vậy quy hoạch nguồn điện và quy hoạch lưới điện được xây dựng trên cơ sở phân tích và phối hợp để tối ưu hoá toàn bộ quy hoạch hệ thống năng lượng.

Sau khi có sơ đồ địa lí vị trí các nhà máy điện và các trung tâm phụ tải, ta phải tiến hành việc quy hoạch phát triển lưới điện với nhiều cấp điện áp khác nhau để truyền tải và phân phối điện năng từ các nhà máy điện đến các hộ tiêu thụ. Khoảng cách truyền tải càng xa, công suất truyền tải càng lớn thì cấp điện áp phải càng cao để tránh tổn thất trên đường dây.

Nguyên lý cơ bản của quy hoạch lưới điện là cực tiểu cấu trúc lưới và chi phí vận hành nhằm thoả mãn yêu cầu của sự phân phối điện năng an toàn và tin cậy tới các trung tâm phụ tải.

Các yêu cầu về độ tin cậy bao gồm:

1) Các yêu cầu vận hành bình thường. Khi các thiết bị của hệ thống năng lượng được vận hành trong những điều kiện tốt, đảm bảo các tiêu chuẩn vận hành khác nhau. Ví dụ như công suất chuyên tải của đường dây, công suất phát, cấp điện áp, dự trữ nóng và trong phạm vi giá cả đã cho.

2) Yêu cầu vận hành ngẫu nhiên. Khi một thiết bị hư hỏng hay khi tải xuất hiện các dao động, độ tin cậy cung cấp điện phải được thoả mãn. Chi phí lưới điện bao gồm sự đầu tư mua sắm thiết bị máy biến thế, thiết bị truyền tải và chi phí cho việc vận hành chúng.

So với quy hoạch nguồn, quy hoạch lưới điện phức tạp hơn. Thứ nhất, quy hoạch lưới điện phải lưu ý đến sơ đồ mạng thực tế và sự đúng đắn của phương án phải được coi là độc lập với các phương án đã biết. Hơn nữa kích thước của các phương án đã chỉ ra của quy hoạch lưới điện phải lớn hơn quy hoạch nguồn. Thứ hai, các ràng buộc của quy hoạch lưới điện phải thoả mãn là rất phức tạp, bao gồm các phương trình phi tuyến (ví dụ ràng buộc về cấp điện áp v.v...) và thậm chí là các phương trình vi phân (ví dụ về vấn đề ổn định). Như vậy khó mà có được một mô hình toán của quy hoạch lưới điện trọn vẹn và việc giải nó thậm chí còn khó hơn.

Để tránh được khó khăn đó, quy hoạch lưới điện được chia làm hai bước: lập sơ đồ và tính giá trị của nó. Nhiệm vụ của việc lập sơ đồ là xác định một hay nhiều phương án có chi phí thấp thoả mãn khả năng tải của các thiết bị truyền tải. Hiện nay các nhà quy hoạch dùng kinh nghiệm của họ để xác định các nhánh mở rộng và cấu hình của lưới điện. Với sự tăng trưởng về kích thước của hệ thống năng lượng, máy tính sẽ bắt đầu được sử dụng để tự động hoá việc quy hoạch lưới điện. Phương pháp này có thể phối hợp với các lĩnh vực công nghệ, kinh tế và tối ưu hoá một cách gần đúng để xác định cấu hình lưới điện tốt hơn, điều đó cho phép cải thiện chất lượng và tốc độ quy hoạch lưới điện. Dù sao vì hiện nay việc quy hoạch lưới điện đang trong bước phát triển, nó không thể thay thế hoàn toàn các nhà quy hoạch trong việc ra quyết định. Nhiệm vụ của việc đánh giá sơ đồ là đánh giá toàn bộ đặc tính kinh tế kỹ thuật của sơ đồ đã cho bao gồm dòng tải, phân tích ổn định, khả năng dòng ngắn mạch, độ tin cậy và tính toán kinh tế, để đi đến quyết định cuối cùng. Qua việc đánh giá sơ đồ, cấu hình của lưới điện có thể được cải thiện, thông qua biện pháp sử dụng các thông tin lấy từ máy tính.

2. PHƯƠNG PHÁP NGHIÊN CỨU

Phương pháp nghiên cứu của bản luận văn này là ứng dụng lý thuyết về ***bài toán mạng*** kết hợp với ***phương pháp nhánh và cận*** để giải bài toán quy hoạch mở rộng lưới điện truyền tải với các đặc điểm: "***tĩnh***", ***một giai đoạn*** – ngắn hạn, tối ưu hoá ***vốn đầu tư xây dựng***.

Với đặc điểm của lưới điện truyền tải bao gồm các nút nối với các phần tử phân bố trên phạm vi rộng nên sẽ rất phù hợp khi ta mô hình hoá thành một bài toán mạng rồi dùng các thuật toán của nó để giải quyết các vấn đề đặt ra. Luận văn sẽ đi sâu nghiên cứu ứng dụng của định lý Dòng chảy cực đại – Tập cắt nhỏ nhất (Max flow – Min cut).

Sau khi đã mô hình hóa hệ thống sang bài toán mạng thì việc xác định vị trí cần mở rộng và loại thiết bị tương đương với việc tìm ra các ***nút cổ chai*** trong mạng. Dựa trên định lý Max flow – Min cut, luận văn đã xây dựng nên thuật toán để có thể tìm được các ***nút cổ chai*** này.

Các nút cổ chai là các nhánh trong mạng (có thể là đường dây hoặc trạm biến áp) mà tại đó ta cần mở rộng. Sử dụng phương pháp nhánh và cận sẽ tìm ra phương án mở rộng có vốn đầu tư nhỏ nhất.

Quy hoạch hệ thống điện là bài toán tối ưu phức tạp, có phạm vi rộng với rất nhiều các tham số, bao gồm quy hoạch nguồn và quy hoạch lưới điện. Quy hoạch mở rộng lưới điện truyền tải dài hạn có mục tiêu là tìm ra được phương án xây dựng thêm các đường dây và thiết bị mới đáp ứng được với dự báo phụ tải với vốn đầu tư và chi phí vận hành nhỏ nhất. Do tính dài hạn nên bài toán được chia thành các giai đoạn và có sự chuyển tiếp, kết hợp giữa mỗi giai đoạn nên nó mang tính "***động***". Bài toán quy hoạch "***tĩnh***" là một bài toán con của bài toán quy hoạch "***động***". Nội dung chính của bài toán quy hoạch "***tĩnh***" là xác định xem ta nên xây thêm "ở đâu – where" và "loại thiết bị gì - what". Hiện nay, có nhiều hướng tiếp cận để giải bài toán trên. Một trong số đó là Phân tích Bender. Do tính phức tạp của bài toán, để việc giải được dễ dàng hơn, phân tích Bender đã chia bài toán thành hai bài toán nhỏ (subproblem): bài toán chính (master subproblem) chỉ xét đến vốn đầu tư xây dựng, bài toán phụ (slave subproblem) tối ưu hoá chi phí vận hành. Giữa hai bài toán con này quan hệ qua lát cắt Bender và chúng sẽ được giải lần lượt và lặp lại cho đến khi hội tụ. Phạm vi của luận văn là nghiên cứu giải bài toán chính – tối ưu vốn đầu tư.

3. ĐÓNG GÓP CỦA LUẬN VĂN

3.1. Về mặt lý thuyết

- Mô hình hoá được một hệ thống điện bất kỳ thành bài toán mạng. Xây dựng được chương trình tìm Max flow – Min cut trong ngôn ngữ lập trình Java.
- Lập được hàm mục tiêu và các ràng buộc của bài toán tối ưu vốn đầu tư, xây dựng được thuật toán để tìm ra phương án tối ưu. Mô hình đã được thử nghiệm thành công trên mạng 21-bus chuẩn của IEEE.

3.2. Về mặt ứng dụng

- Đưa ra một hướng tiếp cận mới trong quy hoạch lưới điện truyền tải, nếu kết hợp thêm với các lý thuyết toán trí tuệ nhân tạo thì có khả năng ứng dụng trong quy hoạch mở rộng lưới điện cho hệ thống điện Việt Nam.
- Mô hình còn có thể ứng dụng cho các bài toán tối ưu của các ngành khác như: Giao thông, Viễn thông, …

4. CẤU TRÚC CỦA LUẬN VĂN

Gồm 4 chương:

- Chương 1: Tổng quan về quy hoạch hệ thống điện
- Chương 2: Một số phương pháp quy hoạch mở rộng lưới điện truyền tải
- Chương 3: Cơ sở lý thuyết của phương pháp
- Chương 4: Tối ưu hoá vốn đầu tư trong quy hoạch mở rộng lưới truyền tải bằng phương pháp nhánh và cận

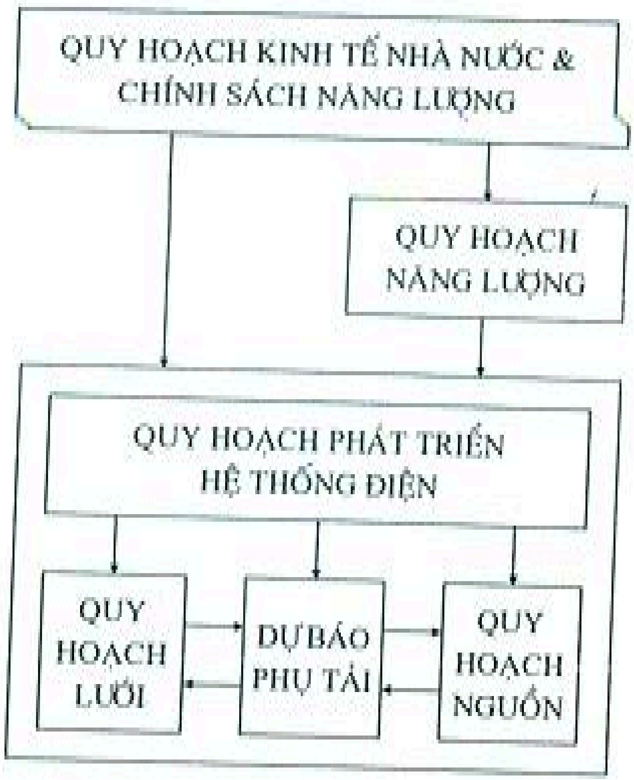
Luận văn kết thúc với phần kết luận.

Chương 1

TỔNG QUAN VỀ QUY HOẠCH HỆ THỐNG ĐIỆN

1.1 GIỚI THIỆU CHUNG

Quy hoạch năng lượng là bài toán được quan tâm đối với mọi quốc gia để có thể cung cấp năng lượng một cách có hiệu quả cho phát triển kinh tế cũng như phục vụ cho xã hội. Quy hoạch hệ thống điện được tiến hành dưới sự chỉ đạo của quy hoạch kinh tế quốc gia và chính sách năng lượng quốc gia. Mối quan hệ đó được thể hiện qua hình sau:



Hình 1.1 Cấu trúc của quy hoạch hệ thống điện

Quy hoạch kinh tế quốc gia và chính sách năng lượng sẽ xác định kế hoạch quy hoạch và phát triển nguồn năng lượng nhằm sử dụng hiệu quả, phối hợp và thay thế các nguồn năng lượng sơ cấp như than đá, dầu khí, thủy điện và điện nguyên tử...

Hệ thống điện là một hệ thống con rất quan trọng trong hệ thống kinh tế quốc gia. Sự phát triển của hệ thống điện chịu ảnh hưởng của những yếu tố như vốn đầu tư, nguồn năng lượng sơ cấp, nhu cầu về điện năng trong tương lai...

Quy hoạch hệ thống điện gồm dự báo phụ tải, quy hoạch nguồn, quy hoạch lưới. Dự báo phụ tải tạo nên cơ sở cho việc quy hoạch hệ thống điện, nó cung cấp thông tin về nhu cầu tiêu thụ điện năng, hình dáng của đường cong phụ tải và việc phân bố tải. Ngược lại kết quả của việc quy hoạch nguồn và lưới có thể dụng làm ảnh hưởng đến đường cong phụ tải và phân bố tải qua tác động của giá. Sơ đồ nguồn và sơ đồ lưới điện là những đặc trưng phụ thuộc trong hệ thống điện. Tuy nhiên hiện tại quy hoạch nguồn và quy hoạch lưới vẫn được giải quyết riêng rẽ như là hai bài toán độc lập. Mặt khác cũng rất khó để giải hai bài toán đồng thời trong cùng một mô hình tổng hợp.

Tùy thuộc vào tình hình kinh tế của mỗi nước mà mức độ quan tâm đối với các bài toán quy hoạch năng lượng có thể khác nhau. Ở nước ta, bài toán quy hoạch năng lượng nói chung và bài toán quy hoạch phát triển nguồn điện nói riêng đã được quan tâm và phát triển vào những năm 70. Cho đến nay đã có nhiều công trình nghiên cứu về vấn đề này ở các cơ quan nhà nước và các trường đại học.

Để có thể truyền tải điện năng từ nhà máy điện đến các hộ tiêu thụ điện và phân phối điện năng cho chúng cần thiết phải có lưới truyền tải và lưới phân phối. Người ta gọi lưới truyền tải là lưới điện có cấp điện áp danh định từ 110kV trở lên còn lưới phân phối là lưới từ 110kV trở xuống.

Các nhà máy điện thường đặt xa các trung tâm phụ tải. Đó là vì nhiều lý do: Đối với nhà máy nhiệt điện do nhà máy phải đặt gần nguồn nhiên liệu vì vận tải điện năng rẻ và thuận tiện hơn nhiều so với vận tải nhiên liệu; nhà máy còn phải đặt gần nguồn nước làm mát và xa khu dân cư để tránh các tác hại của việc gây ô nhiễm môi trường v.v... Ví dụ ở nước ta các nhà máy nhiệt điện Uông Bí và Phả Lại nằm gần vùng than Đông Bắc nhưng lại rất xa các trung tâm phụ tải. Còn đối với nhà máy thuỷ điện là do nhà máy buộc phải đặt ở những nơi mà điều kiện địa lý cho phép như nguồn nước dồi dào, có độ dốc lớn và có thể ngăn đập để tạo hồ chứa ở phía thượng lưu v.v... mà những nơi đó thường rất xa các trung tâm phụ tải. Nhà máy điện tuabin khí phải đặt ở những nơi thuận tiện cho việc cấp khí còn nhà máy điện nguyên tử thì phải đặt xa các khu dân cư cũng như các khu công nghiệp vì lý do an toàn. Muốn xác định vị trí tối ưu của các nhà máy điện cần phải giải bài toán kinh tế kỹ thuật rất phức tạp.

1.2 KHÁI NIỆM VỀ BÀI TOÁN QUY HOẠCH

1.2.1 Bài toán qui hoạch tổng quát

Bài toán qui hoạch tổng quát được phát biểu như sau:

Xác định tập giá trị các biến : $X = \{ x_1, x_2, ..., x_n \}$

Sao cho hàm $f(x_j) \to \min (\max)$ $\quad (j = 1,2,...,n)$ $\quad$ (1-1)

đồng thời thoả mãn các điều kiện $g_i(X)\ (\leq; =; \geq)\ b_i\ \ (i = 1,2,...,m)$ $\quad$ (1-2)

$$x_j \in X \subset R^n.$$

được gọi là 1 bài toán quy hoạch.

Hàm $f(X)$ gọi là hàm mục tiêu.

Các hàm $g_i(X)$; $(i = 1,2,...,m)$ được gọi là các ràng buộc.

Tập hợp $D = \{x \in X;\ g_i(X)\ (\leq;=;\geq)\ b_i\} i = 1...m$ gọi là miền ràng buộc.

Mỗi điểm $X = \{ x_1, x_2 ... x_n \} \in D$ gọi là 1 phương án (PA).

Một PA có : $X^* \in D$ đạt cực đại hay cực tiểu của hàm mục tiêu.

Cụ thể: $\quad f(X^*) \leq f(X),\ \forall X \in D$ (đối với bài toán min)

$\quad f(X^*) \geq f(X),\ \forall X \in D$ (đối với bài toán max) được gọi là lời giải tối ưu.

Khi đó giá trị f(X*) được gọi là giá trị tối ưu hoá của bài toán quy hoạch.

1.2.2 Phân loại bài toán qui hoạch

Một trong những phương pháp giải bài toán được đặt ra là phương pháp duyệt toàn bộ, tìm giá trị hàm mục tiêu của tất cả các phương án có thể trong miền ràng buộc. Sau đó so sánh các giá trị tính được của hàm mục tiêu f(X) để tìm ra giá trị tối ưu và phương án tối ưu của bài toán quy hoạch. Tuy nhiên cách giải quyết này khó có thể thực hiện được, ngay cả khi kích thước bài toán không lớn lắm (số biến n và số ràng buộc m là không lớn) bởi vì tập D thông thường gồm một số rất lớn các phần tử, trong nhiều trường hợp còn không đếm được.

Vì vậy cần có những nghiên cứu lý thuyết để có thể tách bài toán tổng quát thành những bài toán có thể giải được. Các nghiên cứu lý thuyết đó thường là nghiên cứu các tính chất của các thành phần bài toán (hàm mục tiêu, hàm ràng buộc, các biến số, các hệ số). Các điều kiện tồn tại lời giải chấp nhận được, các điều kiện cần và đủ của cực trị, tính chất của các đối tượng nghiên cứu.

Các tính chất của các thành phần bài toán và đối tượng nghiên cứu giúp ta phân loại bài toán.

Một bài toán quy hoạch được gọi là bài toán:

+ Quy hoạch tuyến tính nếu hàm mục tiêu f(X) và tất cả các hàm ràng buộc $g_i(X)$; $i = 1,2,\ldots,m$ là tuyến tính:

$$f(X) = \sum_{j=1}^{n} c_j x_j \rightarrow \min\ (\max) \qquad (1\text{-}3)$$

$$g_i(X) = \sum_{j=1}^{n} a_{ij} x_j (\leq; =; \geq) b_i \qquad (1\text{-}4)$$

$i = 1,m$

Trong đó C_j, a_{ij}, b_i là các hằng số.

+ Quy hoạch tham số nếu các hệ số trong biểu thức hàm mục tiêu và các ràng buộc phụ thuộc tham số.

+ Quy hoạch động nếu đối tượng xét là các quá trình có nhiều giai đoạn nói chung hay các quá trình phát triển theo thời gian nói riêng. Mô hình quy hoạch động thường được coi là công cụ tương đối vạn năng. Để giải bài toán (1-1), (1-2) người ta rời rạc hoá các giá trị của biến. Thực chất của phương pháp này là liệt kê, lựa chọn có quy tắc tổ hợp giá trị (rời rạc) của các biến thoả mãn (1-2) sao cho giá trị của hàm mục tiêu (1-1) đạt cực trị. Mỗi tổ hợp của các biến thoả mãn (1-2) còn được gọi là phương án chấp nhận được. Đối với bài toán qui hoạch phát triển nguồn điện số phương án chấp nhận được thường rất lớn. Do đó bước đầu tiên trước khi thực hiện liệt kê lựa chọn cần loại trừ bớt các phương án có thể là không khả thực, khi đó lại hạn chế nhiều đến tính tối ưu của lời giải. Ngoài ra mô hình quy hoạch động đòi hỏi những thuật toán phức tạp, công cụ tính toán hiện đại, và đặc biệt cần phải đưa vào một số lượng lớn các số liệu ban đầu.

+ Quy hoạch phi tuyến nếu như hoặc f(X) hoặc có ít nhất 1 trong các hàm g(X) là phi tuyến .Về nguyên tắc qui hoạch phi tuyến cho phép mô phỏng bài toán quy hoạch phát triển hệ thống điện chính xác hơn. Tuy nhiên khó khăn chủ yếu của mô hình lại nằm trong các phương pháp giải. Cho đến nay chưa có một phương pháp chung hiệu quả nào cho phép giải trọn vẹn bài toán (1-1), (1-2) trong trường hợp phi tuyến. Trong trường hợp này để tìm cực trị hàm (1-1) thoả mãn ràng buộc (1-2) thường phải dùng các phương pháp lặp, phổ biến nhất là dùng phương pháp tuyến tính hoá và phương pháp Gradient. Ngoài ra còn có thể sử dụng phương pháp Lagrange và phương pháp hàm phạt.

+ Quy hoạch rời rạc nếu miền ràng buộc D là tập rời rạc. Trong trường hợp riêng khi các biến chỉ nhận giá trị nguyên ta có quy hoạch nguyên. Một trường hợp riêng của quy hoạch nguyên là quy hoạch biến Boole, khi các biến số chỉ nhận giá trị 0 hay 1.

+ Quy hoạch đa mục tiêu nếu trên cùng 1 miền ràng buộc ta xét đồng thời các hàm mục tiêu khác nhau.

Các phương pháp kể trên có nhược điểm chung là không đảm bảo được tính hội tụ chắc chắn. Thông thường tính hội tụ đảm bảo được khi các giá trị đầu của lời giải lựa chọn được gần với lời giải tối ưu. Nhược điểm quan trọng khác của phương pháp giải bài toán quy hoạch phi tuyến là không đảm bảo lời giải tối ưu toàn cục. Nhược điểm càng tăng khi số biến cần tìm của bài toán càng nhiều. Như vậy, do tính phức tạp nhiều yếu tố của bài toán qui hoạch phi tuyến nên mô hình quy hoạch phi tuyến thường được đưa về bài toán qui hoạch tuyến tính.

1.3 CÁC PHƯƠNG PHÁP QUY HOẠCH MỞ RỘNG LƯỚI ĐIỆN TRUYỀN TẢI.

Các phương pháp qui hoạch phát triển lưới điện có thể được phân ra thành 2 loại: Các phương pháp tối ưu toán học chặt chẽ và các phương pháp không chính quy.

1.3.1 Các phương pháp quy hoạch không chính quy

Phương pháp không chính quy được đặt trên cơ sở các phân tích trực quan. Nó có quan hệ chặt chẽ với suy nghĩ của các chuyên gia. Nó có thể đưa ra một sơ đồ thiết kế tốt trên cơ sở của kinh nghiệm và sự phân tích. Dù sao nó cũng không phải là một phương pháp tối ưu hoá toán học chặt chẽ.

Phương pháp quy hoạch không chính quy được áp dụng rộng rãi trong quy hoạch lưới điện vì tính chất dễ hiểu, mềm dẻo, tốc độ tính toán nhanh, dễ thu hút cá nhân trong công việc thiết kế và có thể thu được một lời giải tối ưu tương đối mà điều đó phù hợp với những yêu cầu thực tế của kỹ thuật.

Phương pháp không chính quy bao gồm việc kiểm tra quá tải, phân tích độ nhạy và thành lập sơ đồ. Chúng được mô tả như sau:

1. Kiểm tra quá tải: trong giai đoạn lập sơ đồ, vấn đề mấu chốt là liệu có đủ khả năng tải không, tức là liệu có đường dây nào bị quá tải không. Vì vậy, kiểm tra quá tải là điều bắt buộc. Theo sự vận hành bình thường và ngẫu nhiên của thiết bị, ta phải khẳng định rằng không có đường dây nào bị quá tải trong điều kiện làm việc bình thường và đôi khi thậm chí cả trong điều kiện sự cố một đường dây. Điều đó được gọi là "nguyên lý kiểm tra N-1". Vì vậy để kiểm tra một đường dây có bị quá tải hay không là việc tính toán phân phối dòng tải và khả năng tải của một đường dây là rất quan trọng.

Sự cân bằng dòng tải xoay chiều có thể được dùng để thực hiện việc phân tích dòng tải một cách chính xác và đưa ra một sự phân bố toàn diện của công suất tác dụng và công suất phản kháng, điện áp và góc pha trong hệ thống. Phương pháp này, dù sao cũng dẫn tới một khối lượng tính toán rất lớn khi nó cần phải tiến hành phân tích và tính toán nhiều lần trong điều kiện đã biết sơ đồ. Vì vậy, nhiều nhà quy hoạch hiện nay đã áp dụng việc cân bằng dòng tải một chiều để kiểm tra quá tải. Việc cân bằng dòng một chiều là sự đơn giản hoá của việc cân bằng dòng xoay chiều và có đặc điểm là tính toán nhanh và phân tích dễ dàng khả năng tải của đường dây với độ chính xác cao.

2. Phân tích độ nhạy: Khi một đường dây bị quá tải, việc phân tích độ nhạy thường được mở rộng ra lưới điện đó cho đường dây có ảnh hưởng nhất đối với việc giới hạn quá tải. Đường dây có ảnh hưởng ở đây liên quan tới đường dây được đầu tư có hiệu quả nhất. Việc giải thích từ "có ảnh hưởng" ở đâu có khác nhau giữa các nhà quy hoạch với những thể hiện khác nhau.

3. Vẽ sơ đồ: Những phần bổ sung hợp lý có thể được thêm vào để tính hiệu quả nhất của chúng được thể hiện ra bởi vì việc phân tích độ nhạy dẫn đến việc mở rộng lưới điện có thể được thực hiện bằng các phương pháp chắc chắn. Một phương pháp so sánh đơn giản là mở rộng lưới điện từng bước bằng

cách bổ sung một hoặc một nhóm các đường dây hiệu quả hơn. Các phương pháp đặc biệt cũng có thể được sử dụng bằng các bổ sung một tổ hợp của các đường dây hiệu quả có thể có vào hệ thống để cho đường dây tối ưu này nối liền sơ đồ được xác định trên cơ sở cải tiến hệ thống vận hành. Thông qua việc vẽ sơ đồ, nhà quy hoạch có thể can thiệp vào các quá trình ra quyết định thông qua giao diện *người – máy*.

Phương pháp quy hoạch lưới điện này đặc trưng bởi việc mở rộng một lưới điện theo từng bước và do việc thiếu cân nhắc đến mối quan hệ giữa các quyết định của các phần bổ sung nên không thể đảm bảo một lời giải toán học tối ưu và đó là nhược điểm chính của nó.

1.3.2 Các phương pháp quy hoạch toán học

Quy hoạch toán học bằng phương pháp toán học là phương pháp mô hình hoá bài toán quy hoạch lưới điện về dạng toán học rồi dùng các thuật toán tối ưu để tìm ra lời giải tối ưu thoả mãn tất cả các ràng buộc. Mô hình tối ưu toán học của bài toán quy hoạch lưới điện sẽ bao gồm: biến, ràng buộc và một hàm mục tiêu.

- Biến: có hai nhóm sau: biến quyết định và biến trạng thái. Biến quyết định biểu diễn đường dây truyền tải nào được chọn để xây dựng mới vào lưới do đó đây sẽ là biến nguyên. Các biến này sẽ xác định cấu trúc hình học của lưới điện. Biến trạng thái biểu diễn trạng thái vận hành của hệ thống như là dòng công suất, điện áp nút, ... Chúng thường là các biến thực.
- Ràng buộc: bao gồm các điều kiện xây dựng của biến quyết định, cận trên cận dưới của biến trạng thái, ... Hiện nay hầu hết các mô hình toán quy hoạch lưới điện chỉ xét đến các ràng buộc về quá tải đường dây và cân bằng công suất và không xét đến các yêu cầu về điện áp, ổn định...

- Hàm mục tiêu: là một hàm của các biến quyết định và biến trạng thái. Nó chủ yếu bao gồm chi phí đầu tư xây dựng và chi phí vận hành. Mục đích của bài toán quy hoạch lưới điện là tối thiểu hoá hàm mục tiêu nói trên.

Để giải bài toán quy hoạch lưới điện có các công cụ như quy hoạch tuyến tính, quy hoạch động, quy hoạch nguyên hỗn hợp, thuật toán nhánh và cận và phương pháp hình học. Nhìn chung các công cụ trên đang ở trong quá trình phát triển và hoàn thiện nên có một số hạn chế ứng dụng vào thực tế.

So với phương pháp quy hoạch bằng kinh nghiệm, phương pháp quy hoạch bằng toán học có xét đến sự tác động lẫn nhau giữa các biến. Tuy nhiên, do số lượng biến rất lớn và các ràng buộc là rất phức tạp nên các công cụ tối ưu hoá nêu trên sẽ rất khó có thể giải quyết những bài toán cho lưới điện có quy mô lớn. Do đó khi lập công thức toán cho một bài toán quy hoạch lưới, mỗi phương pháp đều có những đơn giản hoá các vấn đề thực tế. Hơn nữa, có một số nhân tố có tính quyết định rất khó có thể mô hình hoá dưới dạng toán học được dẫn đến một lời giải tối ưu toán chưa chắc chắn là một phương án tối ưu trong thực tế. Hiện nay, xu hướng của quy hoạch lưới điện là kết hợp phương pháp kinh nghiệm và phương pháp toán học để đạt được kết quả tối ưu nhất.

Chương 2

MỘT SỐ PHƯƠNG PHÁP QUY HOẠCH TOÁN HỌC MỞ RỘNG LƯỚI ĐIỆN TRUYỀN TẢI

Hiện nay, trên thế giới đang ứng dụng các tiến bộ toán học vào giải quyết các bài toán tối ưu và trong đó có bài toán quy hoạch mở rộng lưới điện truyền tải. Để có một cái nhìn rộng hơn, nội dung của chương 2 sẽ trình bày *sơ lược* một số phương pháp quy hoạch toán học đã được phát triển. Các phương pháp sẽ được đề cập đến là: Phương pháp tìm kiếm Tabu (TS), Thuật toán Kernel-Oriented, Phương pháp liệt kê ẩn 0-1, Phương pháp mô phỏng tôi (SA).

2.1. PHƯƠNG PHÁP TÌM KIẾM TABU.

Phương pháp tìm kiếm Tabu là phương pháp giải quyết vấn đề bằng cách đánh giá kinh nghiệm và tìm đến giải pháp bằng làm phép thử và rút ra sai lầm.

Trong phương pháp này, bài toán quy hoạch mở rộng lưới điện truyền tải được phát biểu dưới dạng bài toán tối ưu phi tuyến nguyên hỗn hợp như sau:

$$\min \quad v = \sum c_{ij} n_{ij} + \sum \alpha_i r_i \qquad (1.1)$$

với các ràng buộc

$$B(x + \gamma^0)\theta + g + r = d \qquad (1.2)$$

$$f_{ij} - (\gamma_{ij}^0 + x_{ij})(\theta_i - \theta_j) = 0 \qquad (1.3)$$

$$|f_{ij}| - x_{ij}\phi_{ij}^{max} \le \gamma_{ij}^0 \phi_{ij}^{max}; \qquad \phi_{ij}^{max} = \frac{f_{ij}^{max}}{\gamma_{ij}^0} \qquad (1.4)$$

$$0 \le g \le g^{max}; \qquad 0 \le r \le d \qquad (1.5)$$

$$x_{ij} = n_{ij}\gamma_{ij}; \qquad 0 \le n_{ij} \le n_{ij}^{\max} \qquad \forall (i,j) \in \Omega \tag{1.6}$$

trong đó:

C_{ij}: chi phí xây dựng mạch mới tại nhánh i-j

n_{ij}: số lượng mạch mới được xây dựng tại nhánh i-j

α: thông số phạt liên quan đến tổn thất của phụ tải do thiếu công suất truyền tải.

r: mảng phụ tải bị sa thải.

B(.): ma trận điện dẫn.

g: mảng các bus công suất tác dụng.

d: mảng các bus dự báo phụ tải.

f_{ij}: dòng công suất tác dụng chạy trong nhánh i-j.

γ_{ij}^{0}: điện dẫn ban đầu của nhánh i-j.

x_{ij}: tổng số mạch điện nạp mới thêm vào nhánh i-j.

θ_i, θ_j: góc pha của điện áp tại bus i và j.

γ_{ij}: điện dẫn của mạch.

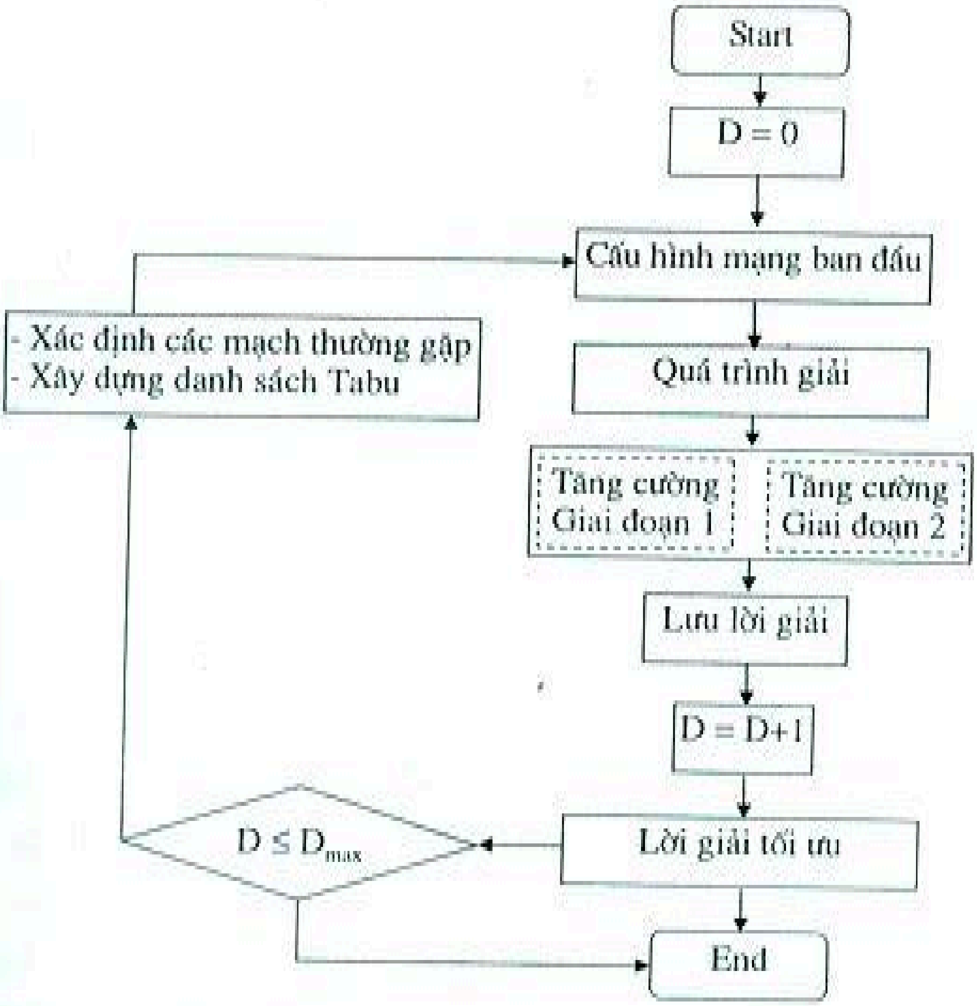
$f_{ij}^{\max}$: giới hạn dòng công suất trong nhánh i-j.

$g^{\max}$: mảng công suất lớn nhất của các bus nguồn.

$n_{ij}^{\max}$: số mạch mới lớn nhất tại nhánh i-j.

Ω: tập hợp tất cả các mạch ứng cử viên.

Trong hàm mục tiêu (1.1) thể hiện chi phí xây dựng các đường dây mới, máy biến áp mới, ... cùng với thông số phạt cao khi phụ tải bị sa thải. Thông số phạt phải đủ lớn để sao cho tại giải pháp tối ưu thì thông số phụ tải bị sa thải phải bằng hoặc gần bằng 0. Thông thường, α được xác định từ việc nghiên cứu tĩnh nhằm chỉ ra những tác động lên khách hàng do mất điện. Thông số này thể hiện mức giá cao nhất mà khách hàng muốn trả để được cung cấp điện liên tục. Lưu ý rằng việc không phải sa thải phụ tải nào phải tương ứng với không có mạch nào trong mạng bị quá tải.



Hình 2.1 : Sơ đồ khối thuật toán tìm kiếm Tabu

2.2. PHƯƠNG PHÁP LIỆT KÊ ẨN 0-1

Trong phương pháp này, bài toán quy hoạch mở rộng lưới điện được mô tả dưới dạng công thức toán học sau đây:

$$\min v = \left\{ \sum_{(i,j)\in\Omega} c_{ij} x_{ij} + \alpha e' r \right\} \tag{1.7}$$

với các ràng buộc

$$Sf + g + r = d \tag{1.8}$$

$$f_{ij} - (\gamma_{ij}^{0} + x_{ij})(\theta_i - \theta_j) = 0 \tag{1.9}$$

$$|f_{ij}| - x_{ij}\overline{\Phi}_{ij} \le \gamma_{ij}^{0}\overline{\Phi}_{ij} \tag{1.10}$$

$$0 \le g \le \overline{g}; \qquad 0 \le r \le d \tag{1.11}$$

$$x_{ij} = n_{ij}\overline{\gamma}_{ij}; \qquad 0 \le n_{ij} \le \overline{n}_{ij}; \qquad \forall (i,j) \in \Omega \tag{1.12}$$

Trong đó:

$x_{ij} = n_{ij}\overline{\gamma}_{ij}$ là các giá trị dung dẫn rời rạc của các mạch có thêm được thêm vào nhánh ij (n_{ij} là số mạch, $\overline{\gamma}_{ij}$ là dung dẫn).

c_{ij} : là chi phí tăng thêm để xây dựng mạch mới trong nhánh ij ($/MW).

S: ma trận nút-nhánh.

f: vectơ dòng công suất tác dụng trong nhánh.

g: vectơ dòng công suất tác dụng của nguồn.

d: vectơ dòng công suất tác dụng của tải.

f_{ij}: vectơ tổng dòng công suất trong nhánh ij.

f_{ij}^{0} : giới hạn công suất truyền tải của các mạch có sẵn trong nhánh ij.

$(\gamma_{ij} + x_{ij})$: tổng dung dẫn của nhánh ij.

θ_i, θ_j : góc pha điện áp tại các nút i và j.

Ω : tập hợp tất cả các mạch mới có thể được thêm vào.

Theo phân tích Bender, bài toán trên được phân tích thành hai bài toán nhỏ như sau:

- Bài toán chi phí vận hành (Phụ)

$$\min w = \alpha e' r \tag{1.13}$$

với các ràng buộc

$$B\theta + g + r = d \tag{1.14}$$

$$|(\theta_i - \theta_j)| \le \overline{\phi}_{ij}; \qquad \forall (i,j) \in \Omega \tag{1.15}$$

$$0 \le g \le \bar{g}; \qquad 0 \le r \le d \tag{1.16}$$

- Bài toán vốn đầu tư (Chính)

$$Min \qquad v = \left\{ \sum_{(i,j) \in \Omega} c_{ij} x_{ij} + \beta \right\} \tag{1.17}$$

với các ràng buộc

$$w^k + \sum_{(i,j) \in \Omega} \sigma_{ij}^k (x_{ij} - x_{ij}^k) \le \beta; \qquad \beta \ge 0 \tag{1.18}$$

$$x_{ij} = n_{ij} \bar{\gamma}_{ij}; \qquad 0 \le n_{ij} \le \bar{n}_{ij}; \qquad \forall (i,j) \in \Omega \tag{1.19}$$

Trong đó

w là giá trị tối ưu của bài toán chi phí vận hành

B là ma trận điện dẫn của mạng

β là giới hạn trên

σ_{ij} là tham số độ nhạy

$$\sigma_{ij} = (\pi_i^{\prime\prime} - \pi_j^{\prime\prime})(\theta_i - \theta_j); \qquad \forall (i,j) \in \Omega \tag{1.20}$$

Thuật toán thứ tự (hierachical algorithm) bao gồm 3 giai đoạn:

- Giai đoạn I:

Mạng được biểu diễn dưới dạng mô hình bài toán vận tải, các bài toán con được giải lặp lại cho đến khi hội tụ. Bài toán vốn đầu tư được giải bằng quy *hoạch tuyến tính*. Bài toán chi phí vận hành được giải ở giai đoạn II sau đây.

- Giai đoạn II:

Mạng được biểu diễn dưới dạng mô hình lai (các nhánh có sẵn dùng mô hình một chiều, các nhánh mới dùng mô hình bài toán vận tải). Mô hình dòng công suất được giải bằng thuật toán tuyến tính đã được chuyên hoá.

- Giai đoạn III:

Mô hình dòng công suất đã được tuyến tính hoá (gọi là mô hình một chiều) được dùng để biểu diễn mạng do tính rời rạc của các phần tử trong lưới truyền tải. Bài toán vốn đầu tư được giải bằng thuật toán liệt kê ẩn còn bài toán chi phí vận hành đã được giải ở giai đoạn 2.

Áp dụng phân tích Bender vào giai đoạn III ta có thuật toán sau:

1. Chọn một giá trị ban đầu cho biến vốn đầu tư x_{ij}.
2. Giải bài toán chi phí vận hành ứng với giá trị x_{ij}.
   - Nếu tổn thất nhỏ hơn ε_w thì dừng.
   - Ngược lại thì ta sẽ giảm vốn (thêm một lượng giảm Bender vào bài toán vốn đầu tư - Bender cuts).
3. Dùng thuật toán liệt kê ẩn 0-1 để giải bài toán vốn đầu tư và xác định mức vốn mới đáp ứng được các ràng buộc được thêm vào ở bước 2. Đến bước 2.

Thuật toán liệt kê ẩn 0-1 giải bài toán vốn đầu tư

Bài toán vốn đầu tư là bài toán tuyến tính nguyên với các biến nguyên biểu diễn số lượng đường dây và máy biến áp được xây dựng thêm. Số lượng các phần tử thêm vào mạng là tương đối nhỏ nên ta có thể biểu diễn qua biến nhị phân. Ví dụ giới hạn trên β số lượng phần tử thêm vào một nhánh là 3 thì các biến nguyên có thể được biểu diễn bằng (0-1) là tập hợp {00, 01, 10, 11}, tương ứng tập các biến nguyên là {0, 1, 2, 3}. Áp dụng vào bài toán vốn đầu tư ta sẽ thu được bài toán tuyến tính ở dạng 0-1 như sau:

$$Min \qquad Z = \sum c_j x_j; \qquad c_j \geq 0; \qquad j \in N = \{1,2,...,n\}$$

với các ràng buộc:

$$\sum_{j \in N} a_{ij} x_j + S_i = b_i; \qquad i \in M = \{1,2,...,m\}$$

$$x_j = (0,1); \qquad j \in N; \qquad S_i \geq 0; \qquad i \in M$$

Bài toán này có thể được giải rất hiệu quả bằng phương pháp liệt kê ẩn. Việc liệt kê toàn bộ tất cả các phương án là không thể thực hiện được đối với các bài toán có số biến nhị phân lớn. Với phương pháp liệt kê ẩn các thông tin được tạo ra trong quá trình liệt kê được sử dụng để xem xét loại trừ một số phương án đề cử. Ta có thể hình dung việc liệt kê giống như đi ngang qua một cây quyết định nhị phân (là tất cả những sự kết hợp có thể nhưng không lặp lại giống nhau của các biến quyết định nhị phân). Theo đó, việc loại bỏ các phương án đề cử giống như ta cắt tỉa cây quyết định. Phương pháp liệt kê ẩn dựa trên sách lược "chia để trị". Sách lược này cũng được ứng dụng trong phương pháp nhánh và cận và các phương pháp khác trong hệ chuyên gia.

2.3. PHƯƠNG PHÁP MÔ PHỎNG TÔI.

Bài toán quy hoạch mở rộng lưới điện truyền tải được viết dưới dạng bài toán quy hoạch phi tuyến nguyên hỗn hợp, trong đó lưới điện được mô hình hoá dưới dạng dòng công suất một chiều như sau:

$$\min \quad v = \sum C_{ij} n_{ij} + \sum \alpha_i r_i \qquad (1.21)$$

với các ràng buộc

$$B(x + \gamma^0)\theta + g + r = d \qquad (1.22)$$

$$(x_{ij} + \gamma_{ij}^0)|\theta_i - \theta_j| \le (x_{ij} + \gamma_{ij}^0)\overline{\phi}_{ij} \qquad (1.23)$$

$$0 \le g \le \overline{g} \qquad (1.24)$$

$$0 \le r \le d \qquad (1.25)$$

$$0 \le n_{ij} \le \overline{n_{ij}} \qquad (1.26)$$

trong đó:

C_{ij}: chi phí xây dựng mạch mới tại nhánh i-j

x_{ij}: tổng điện dẫn thêm vào nhánh i-j.

B(.): ma trận điện dẫn.

θ_i, θ_j: góc pha của điện áp tại bus i và j.

γ_{ij}^0: điện dẫn ban đầu của nhánh i-j.

n_{ij}: số lượng mạch mới được xây dựng tại nhánh i-j; $n_{ij} = \frac{x_{ij}}{\gamma_{ij}}$

$\bar{\phi}_{ij}$: xác định từ tỷ lệ $\bar{\phi}_{ij} = \frac{\bar{f}_{ij}}{\gamma_{ij}}$; với $\bar{f}_{ij}$ là dòng công suất lớn nhất trong nhánh ij

α: thông số phạt liên quan đến tổn thất của phụ tải do thiếu công suất truyền tải.

r: vec tơ nguồn nhân tạo.

g: vec tơ nguồn phát.

d: vec tơ phụ tải.

γ_{ij}: điện dẫn của mạch.

$\bar{g}$: vec tơ công suất lớn nhất của các bus nguồn.

Phương pháp mô phỏng tôi là mô hình hoá bài toán tối ưu phức tạp bằng một cặp (G, v). Trong đó G là tập hợp các cấu hình có kích thước lớn (một không gian các cấu hình) và v là hàm mục tiêu là giá trị thực của từng cấu hình. Sau đó, phương pháp sẽ tìm kiếm cấu hình có v nhỏ nhất. Bắt đầu từ một cấu hình ban đầu, phương pháp mô phỏng tôi sẽ tạo ra một dãy các cấu hình sẽ có thể có chi phí nhỏ nhất. Sự chuyển tiếp giữa 2 cấu hình liên tiếp được quản lý bằng cơ cấu stochastic. Một cấu hình mới được chấp nhận hay không tuỳ thuộc vào giá trị hàm mục tiêu của nó: một cấu hình mà có hàm mục tiêu giảm sẽ được chấp nhận và ngược lại.

Quá trình trên được mô hình hoá giống như việc ta tôi một chất rắn. Nó được hiểu như quá trình làm mát dần dần một chất rắn. Tại một nhiệt độ đã biết, phương pháp mô phỏng tôi sẽ đi từ cấu hình này đến cấu hình kế tiếp cho đến khi đạt được cân bằng nhiệt độ.

Thuật toán của phương pháp mô phỏng tôi có thể được hiểu như sau:

- **Bước 1:**

Xác định nhiệt độ ban đầu T_0

- **Bước 2:**

Xác định các tham số điều khiển: $\rho, \beta, \xi, \omega_0, \varepsilon_w$. Xác định giá trị ban đầu cho số lần lặp của mỗi mức nhiệt độ: $N_0 = \xi . N_t$ với N_t là tổng số các nhánh mà tại đó có thể xây thêm mạch mới.

- **Bước 3:**

Tăng chỉ số đếm bước lặp IT; nếu IT > N_k thì chuyển sang bước 7, nếu ngược lại chuyển sang bước 4

- **Bước 4:**

Nếu chỉ số mất tải của cấu hình hiện tại, w^k_{ls}, nhỏ hơn dung sai, w_{ls}, chuyển sang bước 5, nếu không chuyển sang bước 6

- **Bước 5:**

a) Mô phỏng việc cắt bỏ mạch thứ i được chọn ngẫu nhiên từ các mạch đã được thêm vào trước đó (phụ thuộc vào cấu hình hiện tại). Kiểm tra tính thoả đáng của kết quả theo mô hình chi phí vận hành. Nếu hàm mục tiêu giảm ($\Delta v < 0$) hoặc là xác suất được đưa ra từ phân bố Boltzman P_{ac} lớn hơn $P_r(0,1)$ thì ta chấp nhận cấu hình mới, update con trỏ và w^k_{ls} và chuyển đến bước 3, còn nếu ngược lại chuyển đến bước 5(b).

b) Mô phỏng việc đánh đổi giữa các mạch, ví dụ như cắt bỏ mạch i đã được chọn ở 5(a) và thêm một mạch được chọn ngẫu nhiên j. Kiểm tra tính thoả đáng của kết quả theo mô hình chi phí vận hành. Nếu hàm mục tiêu giảm ($\Delta v < 0$) hoặc là xác suất được đưa ra từ phân bố Boltzman P_{ac} lớn hơn $P_r(0,1)$ thì ta chấp nhận cấu hình mới, update con trỏ và w^k_{ls} và chuyển đến bước 3, còn nếu ngược lại chuyển đến bước 5(c).

c) Mô phỏng việc thêm vào mạch thứ j được chọn từ 5(b). Kiểm tra tính thoả đáng của kết quả theo mô hình chi phí vận hành. Nếu hàm mục tiêu giảm

($\Delta v < 0$) hoặc là xác suất được đưa ra từ phân bố Boltzman P_{an} lớn hơn $P_r(0,1)$ thì ta chấp nhận cấu hình mới, update con trỏ và w^k_{u}, và chuyển đến bước 3.

- **Bước 6:**

a) Mô phỏng việc thêm mạch thứ i được chọn ngẫu nhiên từ các mạch đã được thêm vào trước đó (phụ thuộc vào cấu hình hiện tại). Kiểm tra tính thoả đáng của kết quả theo mô hình chi phí vận hành. Nếu hàm mục tiêu giảm ($\Delta v < 0$) hoặc là xác suất được đưa ra từ phân bố Boltzman P_{an} lớn hơn $P_r(0,1)$ thì ta chấp nhận cấu hình mới, update con trỏ và w^k_{u}, và chuyển đến bước 3, còn nếu ngược lại chuyển đến bước 6(b).

b) Mô phỏng việc đánh đổi giữa các mạch, ví dụ như cắt bỏ mạch i đã được chọn ở 6(a) và thêm một mạch được chọn ngẫu nhiên j. Kiểm tra tính thoả đáng của kết quả theo mô hình chi phí vận hành. Nếu hàm mục tiêu giảm ($\Delta v < 0$) hoặc là xác suất được đưa ra từ phân bố Boltzman P_{an} lớn hơn $P_r(0,1)$ thì ta chấp nhận cấu hình mới, update con trỏ và w^k_{u}, và chuyển đến bước 3, còn nếu ngược lại chuyển đến bước 6(c).

c) Mô phỏng việc cắt bỏ mạch thứ j được chọn từ 6(b). Kiểm tra tính thoả đáng của kết quả theo mô hình chi phí vận hành. Nếu hàm mục tiêu giảm ($\Delta v < 0$) hoặc là xác suất được đưa ra từ phân bố Boltzman P_{an} lớn hơn $P_r(0,1)$ thì ta chấp nhận cấu hình mới, update con trỏ và w^k_{u}, và chuyển đến bước 3.

- **Bước 7:**

Kiểm tra tiêu chuẩn kết thúc, nếu tiêu chuẩn được thoả mãn thì chuyển sang bước 9, nếu không chuyển sang bước 8.

- **Bước 8:**

Update N_k, T_k: $N_{k+1} = \rho.N_k$; $T_{k+1} = \beta.T_k$; chuyển đến bước 3

- **Bước 9:**

Thực hiện tìm kiếm.

Ký hiệu:

T_0: nhiệt độ ban đầu

T_f: nhiệt độ cuối cùng, tiêu chuẩn dừng lại của thuật toán, tại nhiệt độ này việc cải thiện hàm mục tiêu là không đáng kể

N_k: số lần chuyển giao

T_k: nhiệt độ ở lần chuyển giao N_k

β: tỷ số thay đổi nhiệt độ nằm trong khoảng 0,5 đến 0,99

ρ: tỷ số chuyển giao nhiệt độ

2.4. PHƯƠNG PHÁP KERNEL-ORIENTED.

Hiện nay, hệ thống điện của các quốc gia trên thế giới đang có xu hướng chuyển từ một hệ thống điện tập trung (các khâu sản xuất, truyền tải và phân phối điện năng đều do nhà nước nắm giữ) sang một hệ thống điện không tập trung- deregulation (trong đó đối tượng tham gia vào các khâu nói trên sẽ đa dạng hơn). Khi đó, các phương pháp quy hoạch hệ thống điện phải có những thay đổi để phù hợp với xu hướng này.

Phương pháp Kernel – Oriented là phương pháp được xây dựng dựa trên cở sở lý thuyết trò chơi.

Hàm mục tiêu của phương pháp này được viết như sau:

$$\min \frac{1}{2}\sum_{j=1}^{M} c_j P_j^2 \qquad (1.27)$$

với các ràng buộc sau

$$B\Theta + K^T P_D = P \qquad (1.28)$$

$$|B_l A\Theta| \le \overline{P}_l \qquad (1.29)$$

trong đó:

c_j: chi phí xây mới đường dây j.

P_j: công suất tác dụng (hệ đơn vị tương đối) của đường dây j

P_D: vec tơ dòng công suất của đường dây mới khả thi.

M: số lượng các đường dây mới có thể được chọn.

B: ma trận các phần ảo trong mạng

Θ: góc pha

K^T: ma trận chuyển vị của ma trận liên kết nút-nhánh

P: công suất tại các nút

B_L: ma trận đường chéo của các phần tử trong nhánh

$\overline{P}_L$: vectơ công suất tác dụng trong nhánh

A: ma trận mạng.

Ta hình dung các đối tượng tham gia vào các khâu của hệ thống điện được gọi là người tham gia vào trò chơi mở rộng lưới điện. Quyết định của họ sẽ xây dựng thêm đường dây hoặc nhà máy tương tự như quyết định tham gia vào trò chơi với mục đích thu được lợi nhuận cao nhất cho cá nhân mình. Để giải quyết được bài toán này, phương pháp Kernel – Oriented dựa vào lý thuyết trò chơi hợp tác là cơ sở cho các thoả thuận liên kết giữa các người chơi. Các người chơi sẽ đàm phán với nhau theo kiểu xoay vòng. Kết thúc mỗi vòng đàm phán, một liên kết mới sẽ được tạo ra từ hai liên kết cũ hoặc là không có sự chấp nhận liên kết nào. Mỗi vòng đàm phán gồm có 3 giai đoạn:

- **Tính toán và gửi đi các chào hàng liên kết:**

Ban đầu, tất cả các người chơi đều là các liên kết đơn. Chọn ra một liên kết tiêu biểu nhất để tính toán giá chào hàng liên kết. Sau đó, mỗi cấu trúc liên kết mới mà có thể được tạo ra từ các cấu trúc cũ đều được tính toán cơ cấu chi trả gọi là K-stable. Cuối cùng, đề xuất được gửi đến tất cả người chơi còn lại.

- **Hình thành liên kết:**

Nếu số tiền thu được của tất cả thành viên tham gia vào liên kết là lớn hơn so với hiện tại và nếu giá chào hàng nhận được là tốt nhất thì đề xuất được chấp nhận và liên kết được hình thành. Quyết định này được chuyển đến cho tất cả các người chơi.

- **Phân chia chi phí và các quy tắc kết thúc:**

Chi phí được phân chia cho mọi bước lặp và khi quá trình hình thành liên kết chấm dứt thì việc tính toán phân chia chi phí được tiến hành sau cùng. Việc đàm phán sẽ vẫn tiếp tục cho đến khi tất cả các đề xuất của các người chơi bị từ chối hoặc là một liên kết rộng nhất đã được hình thành.

Từ các liên kết đơn, sau cuộc chơi quy hoạch mở rộng, mọi người chơi đều tham gia vào một liên kết rộng nhất với lợi nhuận thu được là lớn nhất.

2.5. TỔNG KẾT CHƯƠNG

Bốn phương pháp được đề cập trên đây đều dựa trên các công cụ toán học có lịch sử ra đời và phát triển chưa lâu. Phương pháp mô phỏng tôi và Phương pháp tìm kiếm Tabu dựa trên hai thuật toán trí tuệ nhân tạo cùng tên. Nghiên cứu ứng dụng các phương pháp tính toán mới thuộc hệ trí tuệ nhân tạo cho các bài toán chưa được giải quyết triệt để bởi các phương pháp toán học cũ đang là một xu hướng trong những năm gần đây trên thế giới. Các kết quả nghiên cứu ứng dụng cho thấy thuật toán trí tuệ nhân tạo có triển vọng ứng dụng to lớn. Phương pháp liệt kê ẩn 0-1 có cùng sách lược với phương pháp nhánh và cận là "chia để trị" có khối lượng tính toán lớn. Phương pháp Kernel-Oriented dựa trên cơ sở lý thuyết trò chơi thích hợp cho quy hoạch hệ thống điện theo xu hướng Deregulation.

Các phương pháp trên rất phức tạp, nếu muốn hiểu sâu về chúng cần phải có sự đầu tư nghiên cứu sâu hơn. Trên đây chỉ là những nội dung mang tính khái quát nhất của từng phương pháp mà tác giả nêu ra nhằm mục đích có cái nhìn đa dạng trong cách tiếp cận giải bài toán quy hoạch mở rộng lưới điện truyền tải.

Chương 3

CƠ SỞ LÝ THUYẾT CỦA PHƯƠNG PHÁP

Phương pháp nhánh và cận để giải bài toán quy hoạch mở rộng lưới điện truyền tải dựa trên cơ sở của các lý thuyết sau:

- Định lý Dòng chảy cực đại – Tập cắt nhỏ nhất (Max flow - Min cut set) trong lý thuyết về mạng và bài toán mạng.
- Thuật toán nhánh và cận.

3.1. ĐỊNH LÝ DÒNG CHẢY CỰC ĐẠI VÀ TẬP CẮT NHỎ NHẤT (MAX FLOW – MIN CUT SET THEOREM)

Định lý Dòng chảy cực đại và Tập cắt nhỏ nhất được hai nhà khoa học Ford và Fulkerson chứng minh vào những năm 70 của thế kỷ trước và cho đến nay nó đã được ứng dụng vào nhiều lĩnh vực khoa học. Trước khi nghiên cứu về định lý này, ta tìm hiểu một số khái niệm cơ bản về mạng và bài toán mạng.

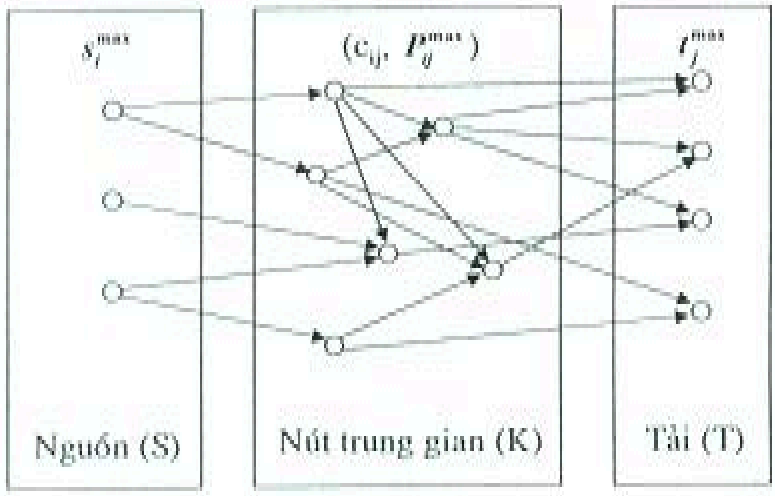
3.1.1 BÀI TOÁN MẠNG (NETWORK PROBLEMS):

Một Graph là một tập hợp các nút và nhánh nối giữa các nút.

Một Graph trọng số được gọi là mạng. Trọng số ở đây thường là các dữ liệu như thời gian, khoảng cách, chi phí, hàng hoá, công suất điện,... Mỗi nhánh hoặc nút trong mạng đều có một trọng số và biểu diễn các đánh giá về sự kết nối giữa các nút.

Bài toán tối ưu hoá được biểu diễn và mô hình hoá dưới dạng một Graph trọng số được gọi là bài toán mạng. Bài toán mạng có rất nhiều dạng và được mô tả tổng quát như sau:

Xét mô hình mạng tổng quát như ở hình vẽ sau:



Hình 3.1: Mô hình mạng của bài toán mạng tổng quát

Trong mạng trên gọi:

S là tập hợp các nút có nguồn (gọi tắt là nút nguồn) với dung lượng lớn nhất mà nguồn cung cấp vào nút i là s_i^{max} với i ∈ S

T là tập hợp các nút có tải (gọi tắt là nút tải) với dung lượng lớn nhất mà tải yêu cầu tại nút j là t_j^{max} với j ∈ T

K là tập hợp các nút trung gian.

c_{ij} : "chi phí" của nhánh nối nút i và nút j với i,j ⊂ K

P_{ij}^{max} : lượng dòng chảy lớn nhất có thể chuyển qua nhánh ij.

P_{ij}: dòng chảy chuyển qua nhánh ij

Bài toán mạng được phát biểu như sau: Chuyển một lượng hàng hoá cho trước (có thể là than, dầu, điện năng, thông tin,...) từ các điểm cung cấp (nguồn-S) đến các điểm thu nhận (tải-T) sao cho chi phí nhỏ nhất với giả thiết tổng dung lượng của các nút nguồn lớn hơn tổng dung lượng của các nút tải ($\Sigma s_i^{max} \geq \Sigma t_j^{max}$)

Mô tả toán học tổng quát của bài toán mạng như sau:

Minimize $\Sigma\Sigma\, c_{ij} P_{ij}$ (3.1)

với các ràng buộc $\Sigma\, P_{ij} \le s_i^{max}$ với mọi $j \in S$ (3.2)

$\Sigma\, P_{ij} \ge t_j^{max}$ với mọi $i \in T$ (3.3)

$\Sigma\, P_{ij} - \Sigma\, P_{ji} = 0$ với mọi $i \in K$ (3.4)

$P_{ij} \le P_{ij}^{max}$ với mọi $i,j \in K$ (3.5)

$P_{ij} \ge 0$ (3.6)

Bài toán mạng tổng quát có các trường hợp riêng sau:

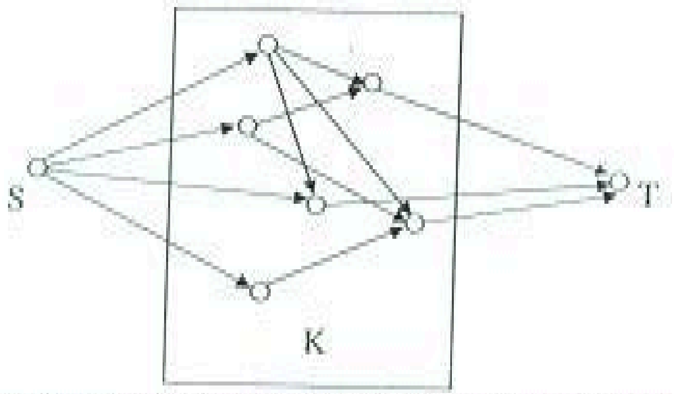
- Trường hợp 1: Khi K là tập rỗng và $P_{ij}^{max} = \infty$ thì bài toán trên sẽ trở thành bài toán vận tải.
- Trường hợp 2: Khi K là tập rỗng, $P_{ij}^{max} = \infty$, $S = T$, $s_i^{max} = 1$ và $t_j^{max} = 1$ bài toán trên sẽ trở thành bài toán phân công.
- Trường hợp 3: Khi $S = T = 1$, $P_{ij}^{max} = 1$, $s_i^{max} = t_j^{max} = 1$, bài toán trên sẽ trở thành bài toán tìm đường đi ngắn nhất.
- Trường hợp 4: Khi $S = T = 1$, $s_j^{max} = t_j^{max} = \infty$, $c_{ij} = -1$, bài toán trên sẽ trở thành bài toán tìm dòng chảy cực đại (Max flow Problem).

Như vậy bài toán tìm dòng chảy cực đại là trường hợp riêng của bài toán mạng tổng quát. Nội dung và kết quả của bài toán này liên quan trực tiếp đến định lý Dòng chảy cực đại và Tập cắt nhỏ nhất, do đó ta sẽ đi sâu nghiên cứu về bài toán này.

3.1.2 BÀI TOÁN TÌM DÒNG CHẢY CỰC ĐẠI (MAXIMUM FLOW PROBLEM)

1. Nội dung bài toán:

Ứng với các giá trị $S = T = 1$, $s_j^{max} = t_j^{max} = \infty$, $c_{ij} = -1$ ta thu được mô hình mạng và mô hình toán của bài toán tìm dòng chảy cực đại như sau:



Hình 3.2: Mô hình mạng của bài toán tìm dòng chảy cực đại

Mô hình toán của bài toán tìm dòng chảy cực đại:

$$\text{Minimize } \Sigma -P_{ij} \quad \Leftrightarrow \quad \text{Maximize } \quad \Sigma P_{ij} \qquad (3.7)$$

với các ràng buộc

$$\Sigma P_{ij} - \Sigma P_{ji} = 0 \text{ với mọi } i \in K \qquad (3.8)$$

$$P_{ij} \leq P_{ij}^{max} \text{ với mọi } i,j \in K, S, T \qquad (3.9)$$

$$P_{ij} \geq 0 \qquad (3.10)$$

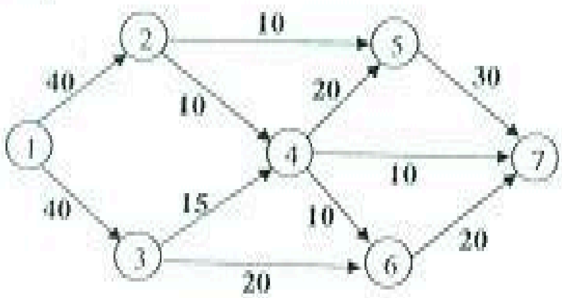
Bài toán tìm dòng chảy cực đại được phát biểu như sau:

Xét một mạng gồm có một nút nguồn và nút tải, hãy tìm dòng chảy lớn nhất mà nút nguồn có thể chuyển tới được nút tải. Biết rằng:

Lượng dòng chảy mà nút nguồn cung cấp vào mạng (n^{max}) là vô cùng lớn

Lượng dòng chảy mà nút tải thu nhận được từ mạng (t^{max}) là vô cùng lớn

2. Ví dụ minh hoạ:



Hình 3.3

Xét bài toán tìm dòng chảy cực đại có mô hình mạng như ở hình 3.3.

Mạng trên gồm có 7 nút trong đó nút 1 là nút nguồn, nút 7 là nút tải.

Các nhánh trong mạng chuyển được lượng dòng chảy lớn nhất như sau (trên hình vẽ các giá trị này được in đậm).

$P_{12}^{max} = 40$ $P_{13}^{max} = 40$ $P_{25}^{max} = 10$ $P_{24}^{max} = 10$ $P_{34}^{max} = 15$ $P_{36}^{max} = 20$

$P_{45}^{max} = 20$ $P_{47}^{max} = 10$ $P_{46}^{max} = 10$ $P_{57}^{max} = 30$ $P_{67}^{max} = 20$

Dung lượng mà nút 1 có thể cung cấp vào mạng là vô cùng lớn nhưng do hai nhánh nối với nút 1 có $P_{12}^{max} = 40$ và $P_{13}^{max} = 40$ nên dung lượng lớn nhất mà nút 1 cung cấp vào mạng là 40 + 40 = 80.

Tương tự, dung lượng mà nút 7 có thể nhận được từ mạng là vô cùng lớn nhưng do các nhánh nối với nút 7 có $P_{57}^{max} = 30$ $P_{67}^{max} = 20$ nên nút 7 chỉ có thể nhận được dung lượng lớn nhất là 10 + 30 + 20 = 60.

Bài toán đặt ra là ta phải tìm dòng chảy lớn nhất mà nút 1 có thể cung cấp đến cho nút 7.

Bài toán được giải như sau:

Trước tiên ta phải tìm một đường dẫn từ 1 đến 7. Đường dẫn từ 1 đến 7 bao gồm các nhánh nối liên thông với nhau và có nhánh đầu đi từ 1, nhánh cuối kết thúc ở 1. Các đường dẫn sẽ là:

1-2-4-5-7	1-2-5-7	1-2-4-7	1-2-4-6-7	1-3-4-6-7	1-3-6-7	1-3-4-7	1-3-4-5-7
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

Do giới hạn P_{ij}^{max} trong các nhánh nên dòng chảy trong đường dẫn sẽ là dung lượng còn dư nhỏ nhất trong các nhánh. Dung lượng còn dư trong nhánh ij (Δ_{ij}) được tính như sau:

$$\Delta_{ij} = P_{ij}^{max} - P_{ij}$$

Ký hiệu dòng chảy trong đường dẫn thứ (i) là F_i

(1)Xét đường dẫn 1-2-4-5-7

$P_{12}^{max} = 40$ $P_{24}^{max} = 10$ $P_{45}^{max} = 20$ $P_{57}^{max} = 30$

Ta thấy 2-4 là nhánh có P_{24}^{max} nhỏ nhất và bằng 10. Như vậy dòng chảy lớn nhất trong đường dẫn này là 10.

Dòng chảy trong từng nhánh

$P_{12} = 10$ $P_{24} = 10$ $P_{45} = 10$ $P_{57} = 10$

$F_1 = 10$

Dung lượng còn dư trong các nhánh là

$\Delta_{12} = 30$ $\Delta_{24} = 0$ $\Delta_{45} = 10$ $\Delta_{57} = 20$

(2)Xét đường dẫn 1-2-5-7

$\Delta_{12} = 30$ $P_{25}^{max} = 10$ $\Delta_{57} = 20$

Dòng chảy trong đường dẫn sẽ là min{30, 10, 20} = 10

Dòng chảy trong từng nhánh

$P_{12} = 10 + 10$ $P_{25} = 10$ $P_{57} = 10 + 10$

$F_2 = 10$

Dung lượng còn dư trong các nhánh là

$\Delta_{12} = 20$ $\Delta_{25} = 0$ $\Delta_{57} = 10$

(3)Xét đường dẫn 1-2-4-7

$\Delta_{12} = 20$ $\Delta_{24} = 0$ $P_{47}^{max} = 10$

Dòng chảy trong đường dẫn sẽ là min {20, 0, 10} = 0

Như vậy $F_3 = 0$

(4)Xét đường dẫn 1-2-4-6-7.

$\Delta_{12} = 20$ $\Delta_{24} = 0$ $P_{46}^{max} = 10$ $P_{67}^{max} = 20$

Dòng chảy trong đường dẫn sẽ là min {20, 0, 10, 20} = 0

Như vậy $F_4 = 0$

(5)Xét đường dẫn 1-3-4-6-7

$P_{13}^{max} = 40$ $P_{34}^{max} = 15$ $P_{46}^{max} = 10$ $P_{67}^{max} = 20$

Dòng chảy trong đường dẫn sẽ là min{40, 15, 10, 20} = 10

Dòng chảy trong từng nhánh

$P_{13} = 10$ $P_{34} = 10$ $P_{46} = 10$ $P_{67} = 10$

$F_5 = 10$

Dung lượng còn dư trong các nhánh là

$\Delta_{13} = 30$ $\Delta_{34} = 5$ $\Delta_{46} = 0$ $\Delta_{67} = 10$

(6)Xét đường dẫn 1-3-6-7

$\Delta_{13} = 30$ $P_{36}^{max} = 20$ $\Delta_{67} = 10$

Dòng chảy trong đường dẫn sẽ là min{30, 20, 10} = 10

Dòng chảy trong từng nhánh

$P_{13} = 20$ $P_{36} = 10$ $P_{67} = 20$

$F_6 = 10$

Dung lượng còn dư trong các nhánh là

$\Delta_{13} = 20$ $\Delta_{36} = 10$ $\Delta_{67} = 0$

(7)Xét đường dẫn 1-3-4-7

$\Delta_{13} = 20$ $\Delta_{34} = 5$ $P_{47}^{max} = 10$

Dòng chảy trong đường dẫn sẽ là min{20, 5, 10}

Dòng chảy trong từng nhánh

$P_{13} = 25$ $P_{34} = 15$ $P_{47} = 5$

$F_7 = 5$

Dung lượng còn dư trong các nhánh là

$\Delta_{13} = 15$ $\Delta_{34} = 0$ $\Delta_{47} = 5$

(8)Xét đường dẫn 1-3-4-5-7

$\Delta_{13} = 15$ $\Delta_{34} = 0$ $\Delta_{45} = 10$ $\Delta_{57} = 10$

Dòng chảy trong đường dẫn sẽ là min{15, 0, 10, 10} = 0

Như vậy $F_8 = 0$

Ta đã xét hết các đường dẫn, dòng chảy mà nút 1 có thể nhánh cấp đến nút 7 sẽ là tổng các dòng chảy trong các đường dẫn:

$$F = \sum_{i=1}^{8} F_i = 10 + 10 + 0 + 0 + 10 + 10 + 5 + 0 = 45$$

Ta chưa thể chắc chắn được giá trị trên là dòng chảy lớn nhất. Ta phải thực hiện lại các thao tác trên nhưng với thứ tự các đường dẫn khác ví dụ như:

Ta tính lại theo thứ tự như sau: (6), (2), (3), (8), (1), (4), (5), (7) thì sẽ thu được tổng dòng chảy là

$$F = \sum_{i=1}^{8} F_i = 20 + 10 + 10 + 15 + 0 + 0 + 0 + 0 = 55$$

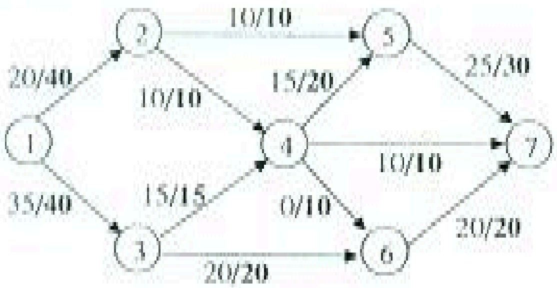
với dòng chảy trong từng cạnh là (được biểu diễn trên hình 3.4):

$P_{12} = 20$ $P_{13} = 35$

$P_{25} = 10$ $P_{24} = 10$ $P_{34} = 15$ $P_{36} = 20$ $P_{45} = 15$ $P_{46} = 0$

$P_{47} = 10$ $P_{57} = 25$ $P_{67} = 20$

Sau khi so sánh ta thấy giá trị F = 55 là lớn nhất và đây chính là lời giải của bài toán tìm dòng chảy cực đại.



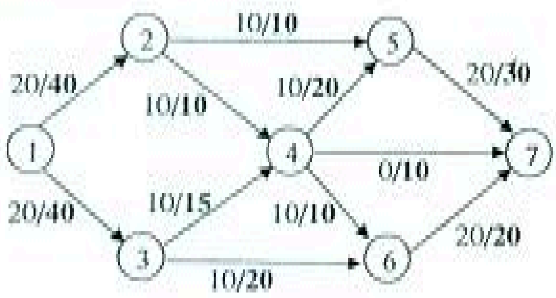
Hình 3.4

Sau quá trình tính toán trên ta rút ra nhận xét:

- Thứ tự tính toán các đường dẫn khác nhau thì tổng dòng chảy khác nhau.
- Trong quá trình tính xuất hiện những đường dẫn có dung lượng còn dư trong các nhánh (Δ_{ij}) đều khác không. Các đường dẫn này gọi là đường dẫn khả tăng.
- Phương pháp tìm dòng chảy lớn nhất phải không được phụ thuộc vào thứ tự tính toán các đường dẫn.

3.1.3 ĐƯỜNG DẪN KHẢ TĂNG:

Đường dẫn khả tăng đóng vai trò quan trọng trong thuật toán tìm dòng chảy lớn nhất. Vẫn xét mạng trên, giả sử trong mạng lúc này có dòng chảy trong các nhánh như hình 3.5. Dòng công suất trong các nhánh P_{ij} được in thường, P_{ij}^{max} của các nhánh được in đậm



Hình 3.5

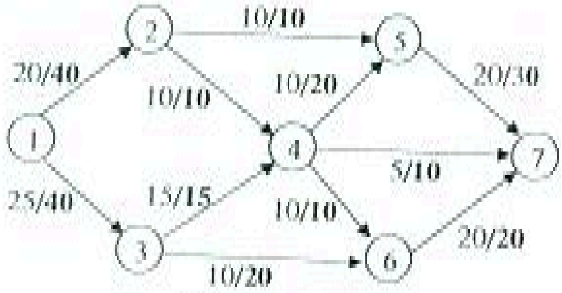
$P_{12} = 20$ $\quad P_{13} = 20$

$P_{24} = 10$ $\quad P_{25} = 10$ $\quad P_{34} = 10$ $\quad P_{36} = 10$ $\quad P_{45} = 10$ $\quad P_{46} = 10$

$P_{47} = 0$ $\quad P_{57} = 20$ $\quad P_{67} = 20.$

Ta có thể thấy rằng đường dẫn 1-3-4-7, tất cả các nhánh có $P_{ij} < P_{ij}^{max}$ (có khả năng tăng thêm dòng chảy). Cụ thể là 1-3 có thể tăng thêm 20, 3-4 có thể tăng thêm 5 và 4-7 có thể tăng lên 10 đơn vị. Dòng chảy có thể tăng thêm trong đường dẫn là giá trị nhỏ nhất trong các lượng dư trên (ở đây là bằng 5)

Từ đó ta có các dòng chảy trong hình 3.6



Hình 3.6

Ta gọi đường dẫn 1-3-4-7 là đường dẫn khả tăng do ta còn có thể tăng dòng chảy trong đường dẫn bằng lượng tăng nhỏ nhất trong các nhánh.

Ta có thể thấy rằng mạng trên không còn đường dẫn khả tăng nào bởi vì mọi đường dẫn đều có ít nhất một nhánh không thể tăng thêm được dòng chảy.

Tuy nhiên ta mở rộng khái niệm đường dẫn khả tăng như sau:

Xét đường dẫn 1-3-6-4-7;

Ta biết rằng dòng chảy từ 6-4 là không được phép, nhưng ta định nghĩa việc một dòng chảy đi từ 6-4 tương đương với việc giảm dòng chảy truyền từ 4-6. Với định nghĩa trên thì đường dẫn 1-3-6-4-7 cũng được gọi là đường dẫn khả tăng.

Dung lượng còn dư ở trong các nhánh là

$A_{13} = 15$ $\quad A_{36} = 10$ $\quad A_{13} = 15$ $\quad A_{47} = 5$

Dung lượng có thể giảm được của nhánh 6-4 là $P_{46} = 10$

Như vậy lượng dòng chảy tăng thêm của đường dẫn 1-3-6-4-7 là

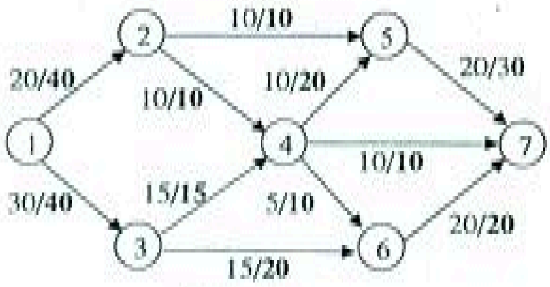
$$\min (15, 10, 15, 5,10) = 5$$

Dòng chảy trong từng nhánh là

$$P_{13} = 30, P_{36} = 15, P_{46} = 5, P_{47} = 10.$$

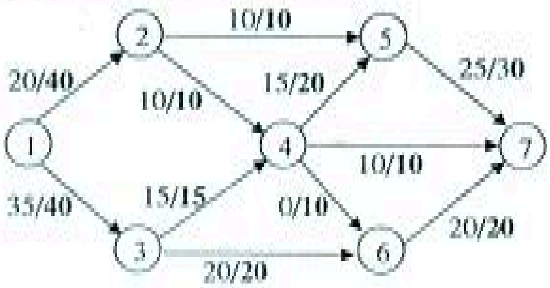
Dòng chảy trong nhánh 4-6 đã được giảm từ 10 xuống 5 và được gọi là dòng chảy nghịch do có chiều từ 6-4.

Các dòng chảy trong mạng được biểu diễn ở hình 3.8 dưới đây:



Hình 3.7

Trong mạng trên ta cũng có thể tìm thấy đường dẫn tương tự là 1-3-6-4-5-7 và sau khi đã phân bố các dòng chảy theo cách trên ta thu được sơ đồ các dòng chảy trong mạng như ở hình 3.8 dưới đây:



Hình 3.8

Qua phân tích trên ta định nghĩa đường dẫn khả tăng như sau:

<u>Đường dẫn khả tăng</u>

Một đường dẫn được gọi là khả tăng khi:

1. Tất cả các nhánh trong đường dẫn đều có khả năng tăng thêm dòng chảy theo hướng quy định (còn dư dung lượng). Lượng dòng chảy tăng thêm bằng lượng dư dung lượng nhỏ nhất trong các nhánh.
2. Có hai loại nhánh là thuận và nghịch. Nhánh nghịch là nhánh có dư dung lượng dòng chảy nhưng ngược với chiều quy định. Và ngược lại là nhánh thuận. Trong trường hợp này lượng dòng chảy tăng thêm là min của lượng dư dung lượng trong các nhánh thuận và dòng chảy trong các nhánh nghịch.

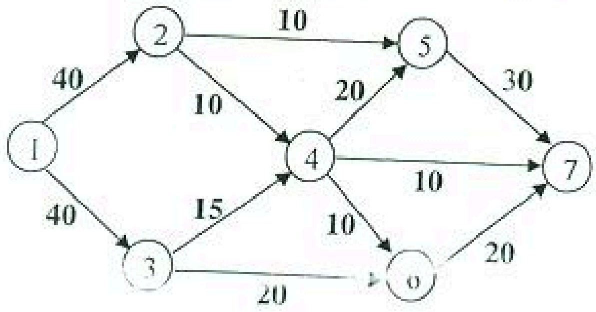
Ta thấy rằng, khi trong mạng không còn đường dẫn khả tăng nào thì dòng công suất trong mạng sẽ là lớn nhất. Từ đó ra rút ra thuật toán sau:

Thuật toán đường dẫn khả tăng

1. Xác định đường dẫn khả tăng và tăng dòng chảy trong mạng.
2. Dừng lại khi không còn tồn tại đường dẫn khả tăng nào.

3.1.4 DÒNG CHẢY CỰC ĐẠI - TẬP CẮT NHỎ NHẤT (MAX FLOW – MIN CUT)

Trong phần này ta sẽ nghiên cứu mối quan hệ giữa max flow - min cut.



Hình 3.9

Một lát cắt được hiểu là khi ta cắt đôi mạng thành hai tập hợp trong đó một tập hợp chứa nút nguồn và một tập hợp chứa nút tải. Có rất nhiều cách cắt một mạng.

Tập cắt là tập hợp các nhánh mà lát cắt cắt qua. Giá trị của tập cắt là tổng các P_{ij} của các nhánh ij nằm trong tập cắt

[illegible] mạng có 7 nút trong ví dụ trên. Nếu ta cắt mạng thành 2 phần tương ứng với 2 tập hợp các nút thì nút nguồn (1) phải luôn thuộc tập thứ nhất và nút tải (7) phải luôn thuộc tập thứ 2.

Ví dụ như ta có thể cắt mạng thành hai tập hợp $S_1 = \{1, 2, 3, 4\}$ và $S_2 = \{5, 6, 7\}$. Điều này có nghĩa là các cạnh 2-5, 3-6, 4-5, 4-6 và 4-7 được bỏ ra khỏi mạng và do vậy không có dòng chảy nào có thể chảy qua hai tập hợp các nút. Tổng dung lượng của các cạnh trên là 10 + 20 + 20 + 10 + 10 = 70 và được gọi là giá trị tập cắt.

Ví dụ ta cắt mạng thành 2 tập hợp các nút S_1 = {1} và S_2 = {2, 3, 4, 5, 6, 7} sẽ có giá trị bằng tổng dung lượng của các cạnh 1-2 và 1-3 và bằng = 80. Cắt thành S_1 = {1, 2, 3, 4, 5, 6} và S_2 = {7} sẽ có giá trị là 30 + 10 + 20 = 60.

Nếu cắt thành S_1 = {1,3,5} và S_2 = {2, 4, 6, 7} có nghĩa là các cạnh 1-2, 3-4, 3-6, và 5-7 bỏ ra khỏi mạng. Dung lượng lúc đó là 40 + 15 + 20 + 30 = 105.

Đối với một mạng thì có rất nhiều cách cắt và tương ứng ta sẽ tính được giá trị tập cắt của nó. Trong các giá trị tập cắt đó ta sẽ tìm ra được một giá trị nhỏ nhất. Ta định nghĩa tập cắt có giá trị tập cắt nhỏ nhất là **tập cắt nhỏ nhất.**

Hai nhà khoa học Ford và Fulkerson đã tìm ra được quan hệ giữa dòng chảy cực đại và giá trị của lát cắt nhỏ nhất và đã chứng minh thành một định lý như sau:

Định lý Dòng chảy cực đại – Tập cắt nhỏ nhất

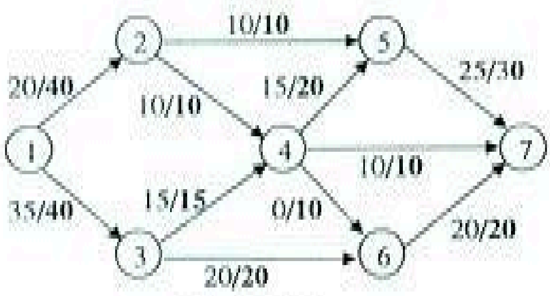
(Max flow – Min cut set Theorem)

Giá trị của dòng chảy lớn nhất bằng giá trị của tập cắt nhỏ nhất.

Quay lại với ví dụ trên ta đã tìm ra được dòng chảy lớn nhất là 55.

Tổng dung lượng các cạnh nằm trong lát cắt S_1 = {1, 2, 3} và S_2 = {4, 5, 6, 7} là tổng của $P_{24} + P_{25} + P_{34} + P_{36}$ = 10 + 10 + 15 + 20 = 55. Như vậy tập cắt này chính là tập cắt nhỏ nhất.

S_{min} = {2-4, 2-5, 3-4, 3-6}



Hình 3.10

Ta có thể mô hình hoá quan hệ giữa Dòng chảy cực đại – Tập cắt nhỏ nhất sang quan hệ **Primal-Dual**.

Xét vào ví dụ trên.

Bài toán tìm dòng chảy cực đại được viết dưới dạng **Primal** sau:

Maximize f

Với

$$
\begin{aligned}
P_{12} &+ P_{13} && && && - f &= 0\\
-P_{12} &+ P_{24} &&+ P_{25} && && &= 0\\
-P_{13} &+ P_{34} &&+ P_{36} && && &= 0\\
-P_{24} &- P_{34} &&+ P_{45} &&+ P_{46} &&+ P_{47} &= 0\\
-P_{25} &- P_{45} &&+ P_{57} && && &= 0\\
-P_{36} &- P_{46} &&+ P_{67} && && &= 0\\
-P_{47} &- P_{57} &&- P_{67} &&+ && f &= 0
\end{aligned}
$$

$P_{ij} \le P_{ij}^{max}$

$P_{ij} \ge 0$

P_{ij}: dòng chảy trong nhánh ij.

P_{ij}^{max} : lượng dòng chảy lớn nhất có thể chuyển qua nhánh ij.

f: tổng dòng chảy về nút 7.

Bài toán tìm tập cắt nhỏ nhất được viết dưới dạng **Dual** như sau:

Minimize $\sum\sum P_{ij}^{max} h_{ij}$

Với các ràng buộc

$w_i - w_j + h_{ij} \ge 0$

$w_7 - w_1 = 1$

w_i có thể nhỏ hơn 0

$h_{ij} \ge 0$

P_{ij}^{max} : lượng dòng chảy lớn nhất có thể chuyển qua nhánh ij

w_i (bằng 0 hoặc 1): biến kép biểu diễn quan hệ giữa nút i với cạnh h_{ij}.

Ta có thể biểu diễn mọi lát cắt dưới dạng các biểu thức trên.

Ví dụ như lát cắt $S_1 = \{1, 2, 3, 4\}$ và $S_2 = \{5, 6, 7\}$ với dung lượng 70 (bằng tổng dung lượng của các cạnh 2-5, 3-6, 4-5, 4-6 và 4-7) được biểu diễn dưới dạng dual như sau:

$$w_7 = 1,\ w_1 = 0$$

$$h_{25} = h_{36} = h_{45} = h_{46} = h_{47} = 1,$$

$$w_2 = w_3 = w_4 = 0$$

$$w_5 = w_6 = 1.$$

Các ràng buộc đều được thoả mãn

$$w_2 - w_5 + h_{25} \geq 0$$

$$w_1 - w_2 + h_{12} \geq 0$$

Giá trị của hàm mục tiêu là 70.

Cũng như vậy lời giải tối ưu ứng với dòng chảy cực đại được biểu diễn như sau:

$$w_1 = w_2 = w_3 = 0,$$

$$w_4 = w_5 = w_6 = w_7 = 1,$$

$$h_{24} = h_{25} = h_{34} = h_{36} = 1$$

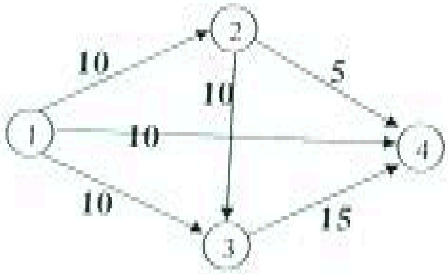
Với giá trị của hàm mục tiêu là 55.

Ta thấy rằng, giá trị của dòng chảy cực đại bằng giá trị của tập cắt nhỏ nhất.

3.1.5 THUẬT TOÁN DÁN NHÃN VÀ PHẦN MỀM TÌM MAX FLOW - MIN CUT

1. Thuật toán dán nhãn:

Xét ví dụ sau:



Hình 3.11

Thuật toán dán nhãn được hiểu như sau:

Đường dẫn khả tăng – Thuật toán dán nhãn

Chọn giá trị ban đầu $P_{ij} = 0$ ở tất cả các nhánh.

1. Đặt L(1) = [- , ∞]
2. Nếu nút i đã được dán nhãn, nút j chưa, $P_{ij} < P_{ij}^{max}$ thì cho L(j) = [+i, Δ_j] trong đó Δ_j = Min {Δ_i, $U_{ij} - P_{ij}$}. Nếu nút i đã được dán nhãn và nút j chưa, $P_{ji} > 0$ thì cho L(j) = [-i, Δ_j] trong đó Δ_j = Min {Δ_i, P_{ji}}. Lặp lại bước 2 cho đến khi nút m được dán nhãn hoặc là không có thêm nút nào được dán nhãn.
3. Nếu nút m chưa được dán nhãn, STOP, đã tìm ra lời giải. Nếu nút m đã được dán nhãn thì cập nhật lời giải như sau:
   Cho ... Nếu nút k là +k, thêm Δ_m vào P_{km}. Nếu nút k là -k, trừ bớt P_{mk} ... là Δ_m. Lặp lại bước này cho đến khi nút 1 được dán nhãn. Quay ... bước 1.

Trong thuật t...

P_{ij}^{max} biểu diễn lượng dòng chảy lớn nhất có thể chảy trong nhánh i-j.

Trong mạng có m nút, nút 1 là nguồn, nút m là tải.

Để làm rõ hơn thuật toán ta xét ví dụ hình 3.9

Vòng lặp 1

Cho tất cả $P_{ij} = 0$

L(1) = [-,∞]. L(2) chưa được dán nhãn và $P_{12} < P_{12}^{max}$. Ta dán nhãn L(2) = [+1, 10]. L(3) chưa được dán nhãn và $P_{23} < P_{23}^{max}$. Ta dán nhãn L(3) = [+2, 10]. L(4) chưa được dán nhãn và $P_{34} < P_{34}^{max}$. Ta dán nhãn L(4) = [+3, 10]. Ta có

$P_{34}^{max} - P_{34} = 15$, ta chọn giá trị nhỏ nhất của (15, L(3), L(4)) có giá trị là 10. Nút m đã được dán nhãn. Ta update các dòng chảy

$$P_{34} = P_{23} = P_{12} = \Delta_4 = 10.$$

<u>**Vòng lặp 2**</u>

L(1) = [-,∞]. L(2) chưa được dán nhãn nhưng $P_{12} = P_{12}^{max}$. Do đó ta không cần xét nút 2. Nút 3 chưa được dán nhãn và $P_{13} < P_{13}^{max}$. Ta dán nhãn L(3) = [+1, 10]. L(4) chưa được dán nhãn và P_{34} (=10) < P_{34}^{max} (=15). Ta dán nhãn L(4) = [+3, 5] do $P_{34}^{max} - P_{34} = 5 <$ L(3). Nút m (4) đã được dán nhãn. Ta cập nhật các dòng chảy như sau:

$$P_{13} = \Delta_4 = 5 \qquad P_{34} = 10 + \Delta_4 = 15.$$

<u>**Vòng lặp 3**</u>

L(1) = [-,∞]. L(2) không được dán nhãn vì $P_{12} = P_{12}^{max}$. Nút 3 chưa được dán nhãn và $P_{13} < P_{13}^{max}$. Cho L(3) = [+1, 5], với $5 = P_{13}^{max} - P_{13}$. L(4) không thể được tính từ nút 3 vì $P_{34} = P_{34}^{max}$. Ta có thể dán nhãn nút 2 từ 3 vì P_{23} là dòng công suất dương. Ta dán nhãn nút 3 như sau L(3) = [-2, 5]. Nút 4 có thể được dán nhãn từ nút 2 do $P_{24} < P_{24}^{max}$. L(4) = [+2, 5].

$$P_{13} = 5 + \Delta_4 = 10.\ P_{23} = 10 - \Delta_4 = 5.\ P_{24} = \Delta_4 = 5.$$

<u>**Vòng lặp 4**</u>

L(1) = [-,∞]. Ta không thể dán nhãn nút 2 vì $P_{12} = P_{12}^{max}$. Ta cũng không thể dán nhãn nút 3 vì $P_{13} = P_{13}^{max}$. Ta có thể dán nhãn nút 4 từ nút 1 vì $P_{14} < P_{14}^{max}$. L(4) = [+1, 10]. Ta cập nhật dòng chảy như sau $P_{14} = \Delta_4 = 10$.

Dòng tổng = 10 + 5 + 5 + 10 = 30

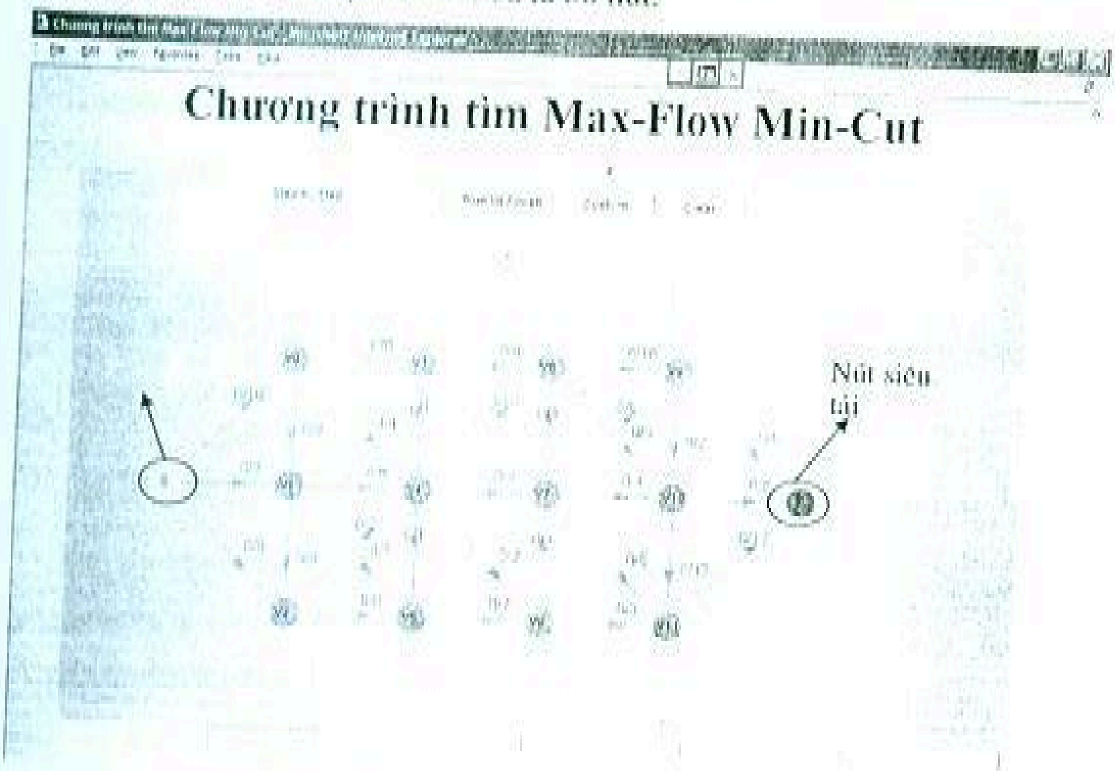
<u>**Vòng lặp 5**</u>

L(1) = [-,∞]. Ta không thể dán nhãn các nút 2 , 3 và 4 do $P_{ij} = P_{ij}^{max}$.

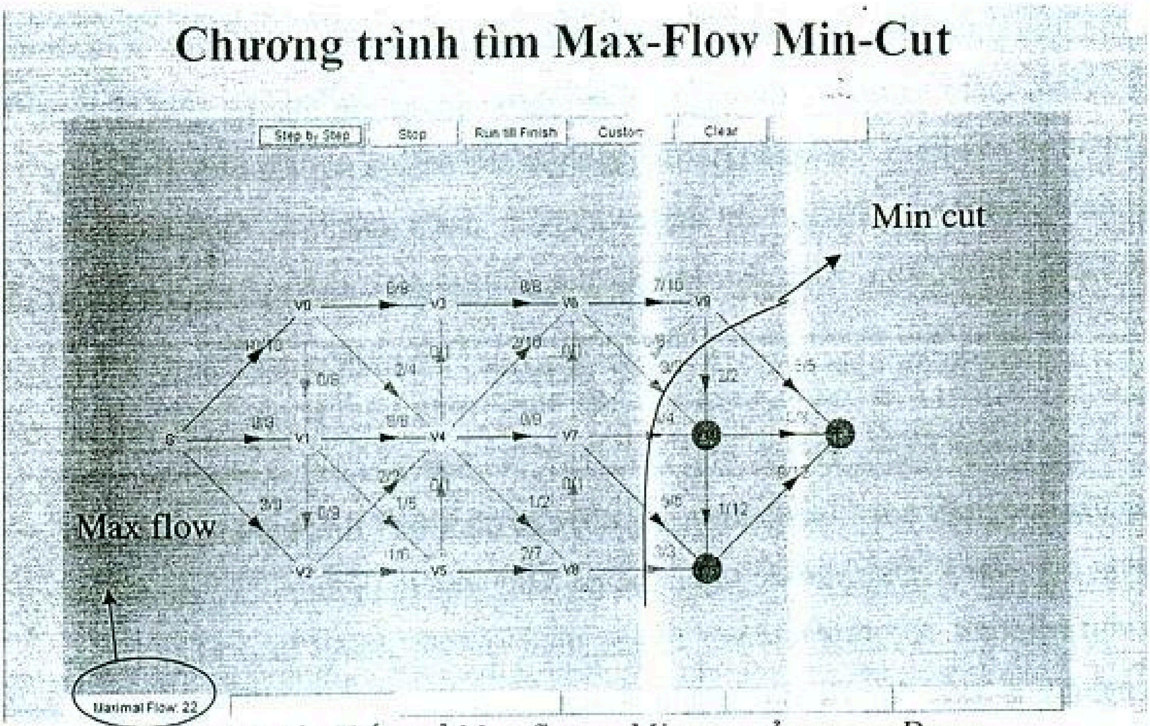
Thuật toán dừng lại với dòng chảy lớn nhất là 30.

2. Phần mềm tìm Max flow – Min cut

Phần mềm tìm Max flow – Min cut được xây dựng dựa trên thuật toán dán nhãn và mô hình hoá quan hệ Dòng chảy cực đại và tập cắt nhỏ nhất sang quan hệ Primal-Dual. Phần mềm được viết bằng ngôn ngữ Java. Giao diện của phần mềm như ở hình 3.13. Sau khi đã mô hình hoá hệ thống sang mô hình mạng, vẽ bằng phần mềm này, phần mềm sẽ tính toán cho chúng ta giá trị dòng chảy cực đại của mạng và các nhánh nằm trong tập cắt nhỏ nhất. Để dễ hình dung, ta xét mạng Demo như ở hình 3.14. Sau khi tính xong, các đỉnh nào thuộc cùng tập với T sẽ mang màu đỏ, các đỉnh nào thuộc cùng tập với S sẽ mang màu vàng. Dựa vào đó ta có thể dễ dàng xác định được các cạnh nằm trong tập Min cut. Đó là các cạnh có hai đỉnh của nó mang màu khác nhau. Phần mềm tính toán được cho tối đa là 50 nút.



Hình 3.12: Giao diện của phần mềm tìm Max flow – Min cut



Hình 3.13: Kết quả Max flow – Min cut của mạng Demo

Nội dung một số file nguồn Java môdul chính của chương trình sẽ được trình bày ở phần phụ lục.

3.2. THUẬT TOÁN NHÁNH VÀ CẬN

Thuật toán nhánh và cận là một thuật toán tìm kiếm.

Khởi đầu từ một vài nhánh (có thể là phương án, lời giải) khả thi, bằng một phương pháp hoặc thuật toán nào đó ta thực hiện quá trình sinh các nhánh. Các nhánh này nằm trong các lời giải có thể đúng của bài toán.

Song song với quá trình sinh nhánh, ta phải đưa các cận vào để kiểm tra, so sánh nhằm tìm ra lời giải tối ưu.

Nếu một tập hợp các nhánh của một lời giải không thoả mãn một trong các cận thì tập hợp các nhánh đó chắc chắn không phải là lời giải của bài toán và bị loại ra khỏi quá trình tìm kiếm, việc sinh nhánh theo các nhánh này bị dừng lại.

Ngược lại

Branch and Bound is a general search method.

Starting by considering the root problem (the original problem with the complete feasible region), the lower-bounding and upper-bounding procedures are applied to the root problem.

If the bounds match, then an optimal solution has been found and the procedure terminates.

Otherwise, the feasible region is divided into two or more regions, these subproblems partition the feasible region.

The algorithm is applied recursively to the subproblems. If an optimal solution is found to a subproblem, it is a feasible solution to the full problem, but not necessarily globally optimal

If the lower bound for a node exceeds the best known feasible solution, no globally optimal solution can exist in the subspace of the feasible region represented by the node. Therefore, the node can be removed from consideration.

The search proceeds until all nodes have been solved or pruned, or until some specified threshold is met between the best solution found and the lower bounds on all unsolved subproblems

3.3. TỔNG KẾT CHƯƠNG

Chương 3 đã trình bày hai cơ sở lý thuyết để xây dựng nên phương pháp nhánh và cận áp dụng cho bài toán quy hoạch mở rộng lưới điện truyền tải.

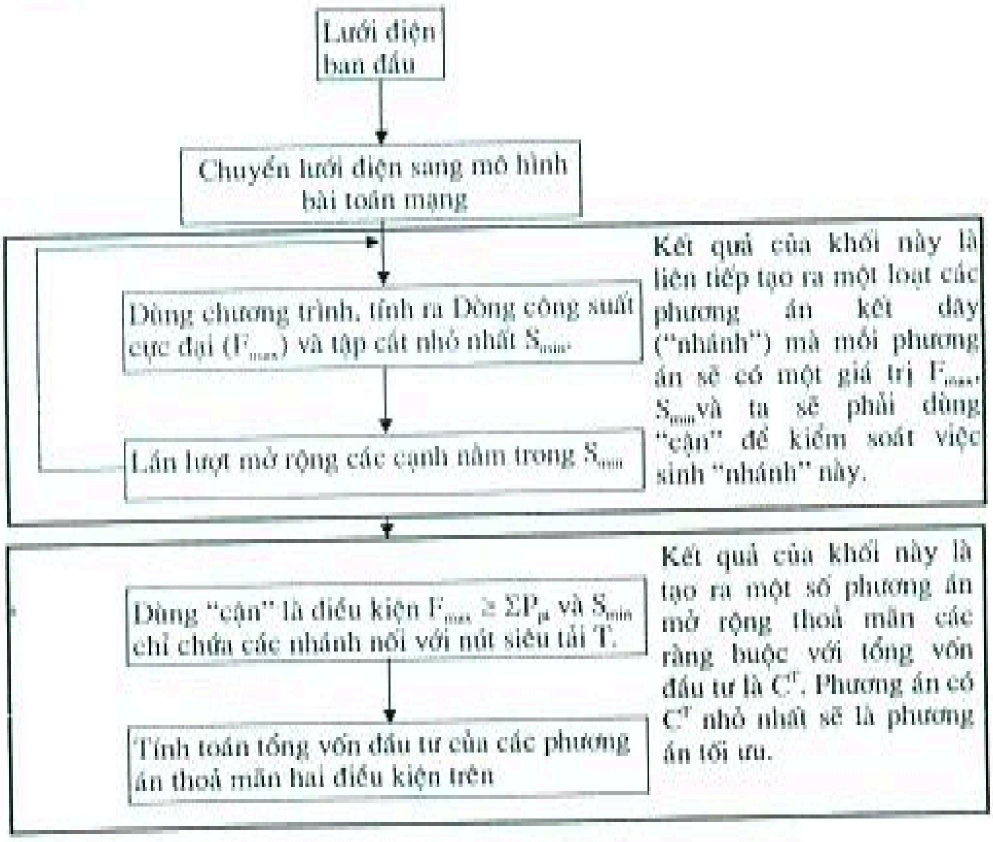
- Định lý Dòng chảy cực đại và Tập cắt nhỏ nhất.
- Phương pháp nhánh và cận tổng quát.

Chương 4

QUY HOẠCH MỞ RỘNG LƯỚI TRUYỀN TẢI BẰNG PHƯƠNG PHÁP NHÁNH VÀ CẬN

Chương 3 đã trình bày một cách đầy đủ các lý thuyết sẽ được sử dụng để xây dựng nên phương pháp nhánh và cận. Trong chương 4 dưới đây sẽ trình bày cụ thể các ứng dụng của các lý thuyết trên để giải quyết bài toán tối ưu hoá vốn đầu tư trong quy hoạch mở rộng lưới điện truyền tải.

Phương pháp nhánh và cận trong bài toán này được hiểu một cách sơ lược như sau:

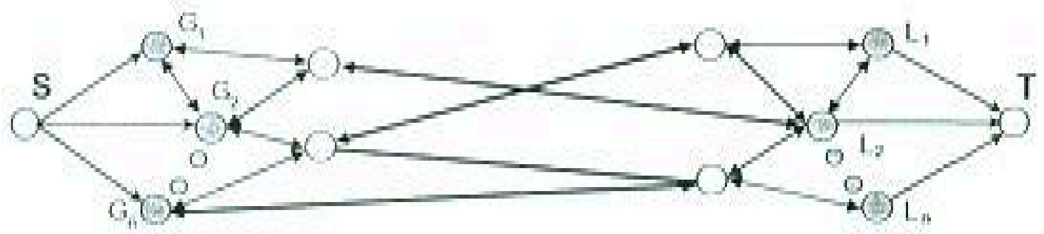


Sau đây ta sẽ đi nghiên cứu cụ thể nội dung của phương pháp

4.1. NỘI DUNG PHƯƠNG PHÁP

4.1.1. CHUYỂN HỆ THỐNG ĐIỆN SANG MÔ HÌNH BÀI TOÁN MẠNG

Để giải quyết bài toán ta sẽ mô hình hoá lưới điện trên thành một network có dạng tổng quát sau:



Hình 4.1: Mô hình tổng quát dạng network của hệ thống điện

Trong đó:

S: được gọi là một siêu nguồn có công suất vô cùng lớn

G: tập hợp các nút có nối với máy phát

L: tập hợp các nút có nối với phụ tải

T: được gọi là một siêu tải có công suất vô cùng lớn

Mạng trên bao gồm có một nút siêu nguồn S và một nút siêu tải T, và một tập hợp các nút trung gian bao gồm các nút có nối với máy phát (G), các nút có nối với phụ tải (L) và các nút không nối với máy phát cũng không nối với tải. Cạnh nối từ S đến các nút có máy phát chỉ có một chiều đi từ S và có khả năng truyền tải lượng công suất lớn nhất bằng công suất phát của máy phát tại nút đó. Cạnh nối các nút có phụ tải đến T chỉ có một chiều đến T và có khả năng truyền tải lượng công suất lớn nhất bằng công suất phát của phụ tải tại nút đó. Cạnh nối giữa các nút không phải là nút S và T có hai chiều và có khả năng truyền tải lượng công suất lớn nhất bằng giới hạn truyền tải của đường dây hoặc trạm biến áp tương ứng với sơ đồ kết dây của lưới điện đang xét. Việc xác định các cạnh nối giữa các nút dựa theo sơ đồ kết dây của lưới điện đang xét.

Gọi:

F_{max} : tổng dòng công suất lớn nhất các phụ tải có thể nhận được

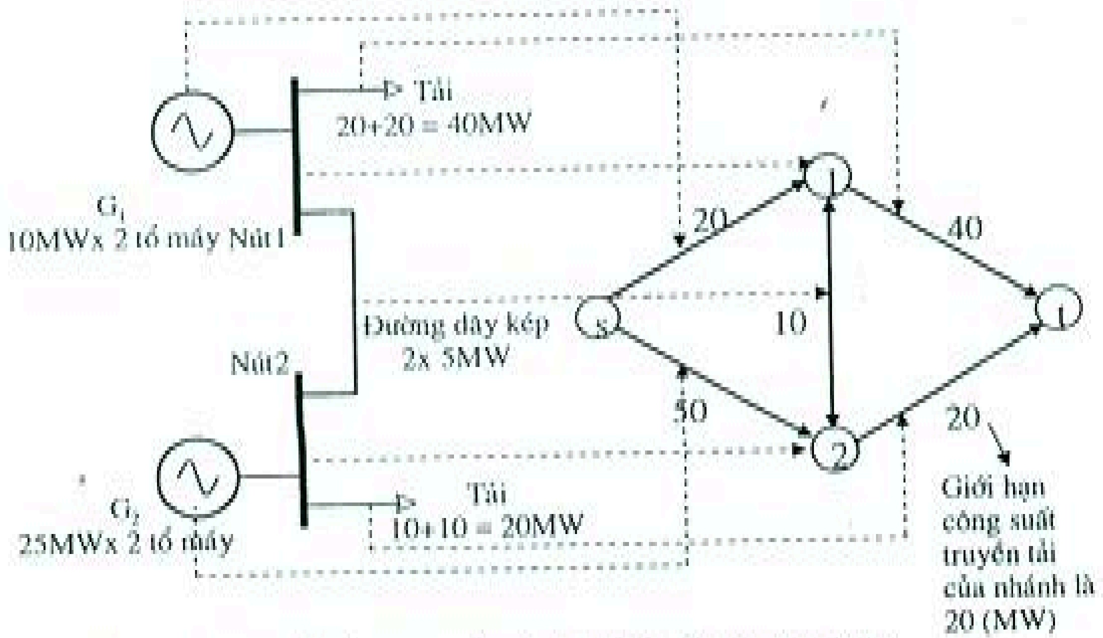
G : tổng công suất phát

L : tổng công suất của phụ tải

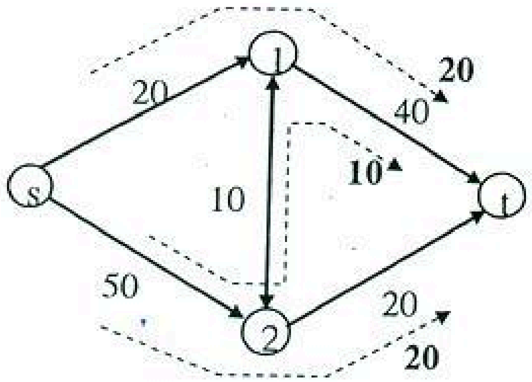
Ba đại lượng trên tồn tại các quan hệ sau:

$F_{max} = L \leq G$	Hệ thống không bị thiếu công suất phát
$F_{max} = G < L$	Thiếu công suất phát
$F_{max} < L \leq G$	Giới hạn truyền tải của đường dây và trạm biến áp bị thiếu khiến cho mặc dù thừa công suất phát nhưng phụ tải vẫn không được cung cấp đủ công suất
$F_{max} < G < L$	Thiếu công suất phát và giới hạn truyền tải của đường dây, trạm

Để dễ hình dung ta xét một mạng điện đơn giản sau:



Hình 4.2: Mô hình network của hệ thống 2 nút đơn giản



Hình 4.3

Ở đây:

$F_{max} = 20 + 10 + 20 = 50$ (MW)

$G = 20 + 50 = 70$ (MW)

$L = 40 + 20 = 60$ (MW)

Như vậy mạng này có

$F_{max} < L \leq G$

Mặc dù thừa công suất phát nhưng tải vẫn bị thiếu 10(MW) do giới hạn truyền tải của đường dây chỉ bằng 10 (MW).

Ta có thể hình dung trong trường hợp này, đường dây chính là "***nút cổ chai***" của mạng. Chính vì vậy ta cần phải xây thêm đường dây tại vị trí này. Với mạng đơn giản trên ta có thể dễ dàng tìm thấy "***nút cổ chai***" của mạng. Đối với các hệ thống điện có nhiều nút, muốn tìm ra "***nút cổ chai***" ta phải áp dụng thuật toán Ford - Fulkerson và định lý Max flow - Min cut. Thuật toán Ford - Fulkerson sẽ đưa ra cho ta giá trị F_{max} cùng với lát cắt nhỏ nhất của mạng. Định lý Max flow - Min cut đã chứng minh rằng F_{max} bằng tổng các dòng công suất của các nhánh nằm trong lát cắt nhỏ nhất nên nếu F_{max} nhỏ hơn tổng công suất phụ tải L thì có nghĩa là lát cắt nhỏ nhất chính là "***nút cổ chai***" của mạng. Ta sẽ cần phải xây dựng thêm đường dây hoặc trạm biến áp tại các nhánh trong lát cắt nhỏ nhất nhưng với ràng buộc là tổng chi phí xây dựng phải là nhỏ nhất.

4.1.2. HÀM MỤC TIÊU VÀ CÁC RÀNG BUỘC

1. Hàm mục tiêu:

$$\min C^T = \sum_{(u,v)\in B}\left[\sum_{i=1}^{m(u,v)} C^i(u,v)U^i(u,v)\right].$$

trong đó:

* C^T : tổng vốn đầu tư xây dựng thiết bị mới.

* (u,v): nhánh giữa hai nút u và v ∈ B (tập hợp của tất cả các nhánh).

* m(u,v) : số nhánh mới giữa hai nút (u,v).

* $U^i(u,v)$: biến quyết định được định nghĩa như sau:

$$U^i(u,v) = \begin{cases} 1, & P(u,v) = P^{(0)}_{(u,v)} + P^{(i)}_{(u,v)} \\ 0, & P(u,v) \neq P^{(0)}_{(u,v)} + P^{(i)}_{(u,v)} \end{cases}$$

$$C^i(u,v) = \sum_{j=1}^{i} \Delta C^{(j)}_{(u,v)}$$

với $\Delta C^{(j)}_{(u,v)}$:chi phí xây dựng phần tử song song thứ j của nhánh nối giữa hai nút x và y.

2. Các ràng buộc:

a. Ràng buộc không thiếu hụt công suất cho phụ tải

$$P_C(V,\overline{V}) \geq L \quad (s \in V; t \in \overline{V})$$

- $P_C(V,\overline{V})$: tổng các dòng công suất trong các nhánh thuộc lát cắt nhỏ nhất (và cũng bằng giá trị F_{max})
- V : Tập con của N chứa các nút trong đó có nút siêu nguồn (S)
- $\overline{V}$: N-V chứa các nút trong đó có nút siêu tải (T)
- L : tổng phụ tải

b. Ràng buộc về phụ tải

$$\sum_{(u,v)\in(V_i,\overline{V}_i)}\left[P^{(0)}(u,v) + \sum P^{(i)}(u,v)U^i(u,v)\right] \geq L$$

* k : chỉ số của tập cắt nhỏ nhất k =1,2, ..., k, ..., n.

* P (u,v) : công suất truyền tải trên nhánh nối giữa nút u và nút v.

* $P^{(i)}(u,v) = \sum_{j=1}^{i} \Delta P^{(j)}_{(u,v)}$

$\Delta P^{(j)}_{(u,v)}$: dung lượng của phần tử xây thêm thứ j ở nhánh nối giữa nút u và nút v

4.1.3. SƠ ĐỒ KHỐI

Bước 1: Tính Max flow – Min cut của mạng ban đầu ta sẽ thu được một tập hợp các nhánh nằm trong Min cut và đây chính là các nhánh mà ta có khả năng mở rộng.

Bước 2: Cho j=1 (j là số thứ tự của hệ thống – Hệ thống thứ 1 tương đương với hệ thống ban đầu), $j_{tư}$ = 0 (thứ tự của hệ thống tối ưu), j_{max} = 0, $C^T_{tư}$ = 0 (vốn đầu tư tối ưu) và NC_j = 0 (nếu NC_j = 1 có nghĩa là hệ thống thứ j là một tối ưu cục bộ)

Bước 3: Kiểm tra NC_j=1 hay = 0 cho hệ thống thứ #j. Nếu = 1, hệ thống thứ #j được sử dụng làm phương án tối ưu cục bộ, chuyển sang bước 15.

Bước 4: Tính toán Max flow – Min cut cho hệ thống thứ j ta thu được tập các nhánh S_j

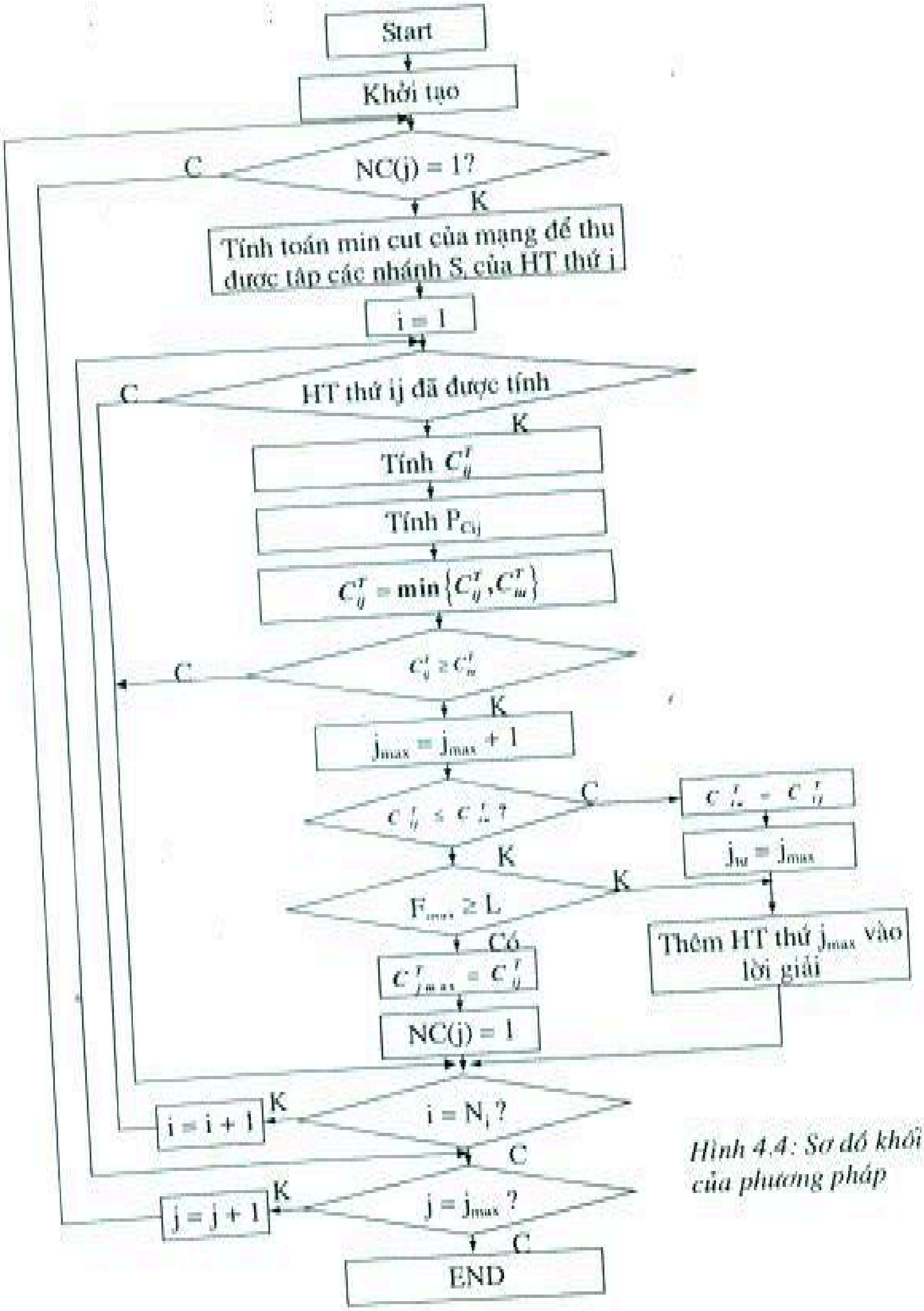
Bước 5: Chọn một nhánh i trong tập S_j thêm vào hệ thống thứ #j. Hệ thống này tạm thời được gọi là hệ thống thứ ij.

Bước 6: Kiểm tra xem hệ thống ij có trùng với lời giải cục bộ không. Nếu có chuyển sang bước 14.

Bước 7: Tính vốn đầu tư $C^T_{ji} = C^T_j + C(P^{(i)}(u,v))$ cho hệ thống thứ #ji.

Bước 8: Tính tổng công suất giới hạn của các cạnh nằm trong Min cut của hệ thống thứ ij. $P^{ij}_C(V,\overline{V})$

Bước 9: Tính C^T_{ji} = minimum $\{C^T_{ji}, C^T_{tư}\}$.



Hình 4.4: Sơ đồ khối của phương pháp

Bước 10: So sánh C^I_{jt} với C^I_{jt}. Nếu C^I_{ji} nhỏ hơn C^I_{opt}, ta không cần phải xét các phương án của hệ thống thứ #ji, chuyển sang bước 14.

Bước 11: Cho $j_{max} = j_{max} + 1$.

Bước 12: Nếu C^I_{ji} nhỏ hơn C^I_{bt} cho $C^I_{bt} = C^I_{ji}$ và $j_{bt} = j_{max}$ và chuyển sang bước 14.

Bước 13: Nếu F_{max} lớn L, cho $C^I_{jmax} = C^I_{ji}$ và $NC_{jmax} = 1$ chuyển sang bước 15. Nếu F_{max} nhỏ hơn L, chuyển sang bước tiếp theo.

Bước 14: Thêm phương án $\#j_{max}$,(#ji) mạng.

Bước 15: Kiểm tra xem tất cả các nhánh trong tập S_i đã được xét hết chưa? Nếu chưa cho i=i+1 chuyển sang bước 5. Nếu có thì chuyển sang bước 16.

Bước 16: Kiểm tra $j = j_{max}$ hay chưa. Nếu chưa cho j = j +1 và sang bước 4.

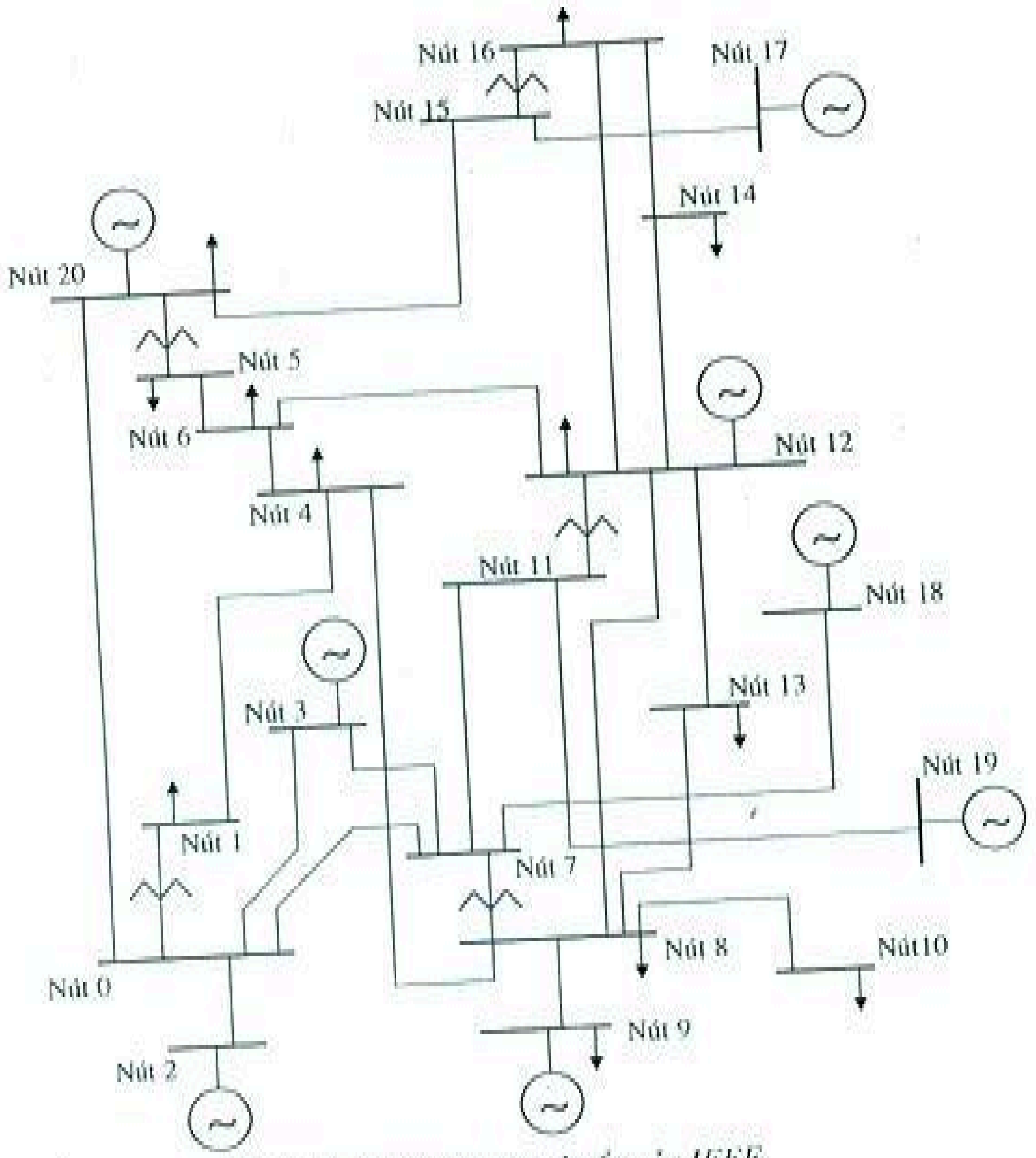
Bước 17: Với $j = j_{max}$, phương án tìm được là phương án tối ưu. j_{bt} cùng với C^I_{bt} nhỏ hơn vốn đầu tư tại bước 12. Kết thúc quá trình tính toán.

4.2. ÁP DỤNG PHƯƠNG PHÁP TÍNH TOÁN CHO MẠNG 21 NÚT CHUẨN CỦA HỘI ĐỒNG ĐIỆN & ĐIỆN TỬ QUỐC TẾ (IEEE)

Theo quy định của IEEE, các nghiên cứu về quy hoạch lưới điện đều phải được tính toán, kiểm tra trên các mạng chuẩn của IEEE. Mạng chuẩn ở đây có thể hiểu là mạng chung với số liệu chuẩn trên đó các nghiên cứu theo các hướng khác nhau cùng tiến hành để dễ dàng đánh giá tính chính xác của các nghiên cứu.

4.2.1. SỐ LIỆU ĐẦU VÀO:

Mạng 21 nút chuẩn của IEEE:



Hình 4.5 : Mạng 21 nút chuẩn của IEEE

Với các số liệu cho ở bảng sau: P[MW], C[M$]

Bảng 4.1 : Số liệu công suất giới hạn và chi phí xây dựng của các nhánh

TT	NĐ	NC	Loại	P(0)	P(1)	P(2)	P(3)	P(4)	C(0)	C(1)	C(2)	C(3)	C(4)
1	S	2	MF	850	0	0	0	0	0	0	0	0	0
2	S	20	MF	600	0	0	0	0	0	0	0	0	0
3	S	3	MF	850	0	0	0	0	0	0	0	0	0
4	S	9	MF	900	0	0	0	0	0	0	0	0	0
5	S	19	MF	800	0	0	0	0	0	0	0	0	0
6	S	17	MF	850	0	0	0	0	0	0	0	0	0
7	S	12	MF	460	0	0	0	0	0	0	0	0	0
8	S	18	MF	725	0	0	0	0	0	0	0	0	0
9	5	20	TBA	1020	510	510	0	0	0	132	132	0	0
10	15	16	TBA	1020	510	510	0	0	0	124	124	0	0
11	11	12	TBA	1020	510	510	0	0	0	123	130	0	0
12	7	8	TBA	800	800	0	0	0	0	155	0	0	0
13	0	1	TBA	800	800	0	0	0	0	151	0	0	0
14	20	0	ĐD	500	500	500	0	0	0	90	90	0	0
15	1	4	ĐD	220	220	0	0	0	0	54	0	0	0
16	0	3	ĐD	300	300	0	0	0	0	73	0	0	0
17	0	7	ĐD	400	400	0	0	0	0	70	0	0	0
18	0	2	ĐD	1000	250	250	250	250	0	20	20	20	20
19	3	7	ĐD	300	300	0	0	0	0	63	0	0	0
20	4	8	ĐD	220	220	0	0	0	0	82	0	0	0
21	4	6	ĐD	220	220	0	0	0	0	77	0	0	0
22	6	5	ĐD	220	220	0	0	0	0	85	0	0	0
23	20	15	ĐD	1000	250	250	250	250	0	30	0	0	0
24	6	12	ĐD	220	220	0	0	0	0	88	0	0	0
25	12	16	ĐD	220	220	0	0	0	0	69	0	0	0
26	12	14	ĐD	220	220	0	0	0	0	83	0	0	0
27	15	17	ĐD	1320	330	330	330	330	0	32	32	32	32
28	8	12	ĐD	220	220	0	0	0	0	71	0	0	0
29	8	13	ĐD	220	220	0	0	0	0	65	0	0	0
30	7	18	ĐD	620	620	0	0	0	0	64	0	0	0
31	11	19	ĐD	1240	310	310	310	310	0	28	28	28	28
32	11	7	ĐD	400	400	0	0	0	0	62	0	0	0
33	8	9	ĐD	240	240	0	0	0	0	81	0	0	0
34	8	10	ĐD	340	340	0	0	0	0	45	0	0	0
35	14	16	ĐD	220	220	0	0	0	0	80	0	0	0
36	12	13	ĐD	220	220	0	0	0	0	80	0	0	0
37	20	T	Tải	785	0	0	0	0	0	0	0	0	0
38	5	T	Tải	750	0	0	0	0	0	0	0	0	0
39	1	T	Tải	850	0	0	0	0	0	0	0	0	0
40	8	T	Tải	595	0	0	0	0	0	0	0	0	0
41	9	T	Tải	17	0	0	0	0	0	0	0	0	0
42	10	T	Tải	550	0	0	0	0	0	0	0	0	0
43	13	T	Tải	190	0	0	0	0	0	0	0	0	0
44	12	T	Tải	710	0	0	0	0	0	0	0	0	0
45	14	T	Tải	450	0	0	0	0	0	0	0	0	0
46	16	T	Tải	870	0	0	0	0	0	0	0	0	0
47	6	T	Tải	290	0	0	0	0	0	0	0	0	0
48	4	T	Tải	70	0	0	0	0	0	0	0	0	0

Bảng 4.2: Dự báo phụ tải

Nút có phụ tải	Công suất phụ tải cũ (MW)	Công suất phụ tải mới (MW)	Tăng thêm (MW)
20	500	785	285
5	450	750	300
1	525	850	325
8	360	595	235
10	325	550	225
12	400	710	310
16	550	870	320
Các nút khác	1017	1017	0
Tổng	4127	6127	2000

Bảng 4.3: Dự báo nguồn

Nút có nguồn	Công suất nguồn cũ (MW)	Công suất nguồn mới (MW)	Tăng thêm (MW)
Nút 20	600	900	300
Nút 19	800	1200	400
Nút 12	400	760	360
Nút 18	725	950	225
Các nút khác	3450	3450	0
Tổng	5975	7260	1285

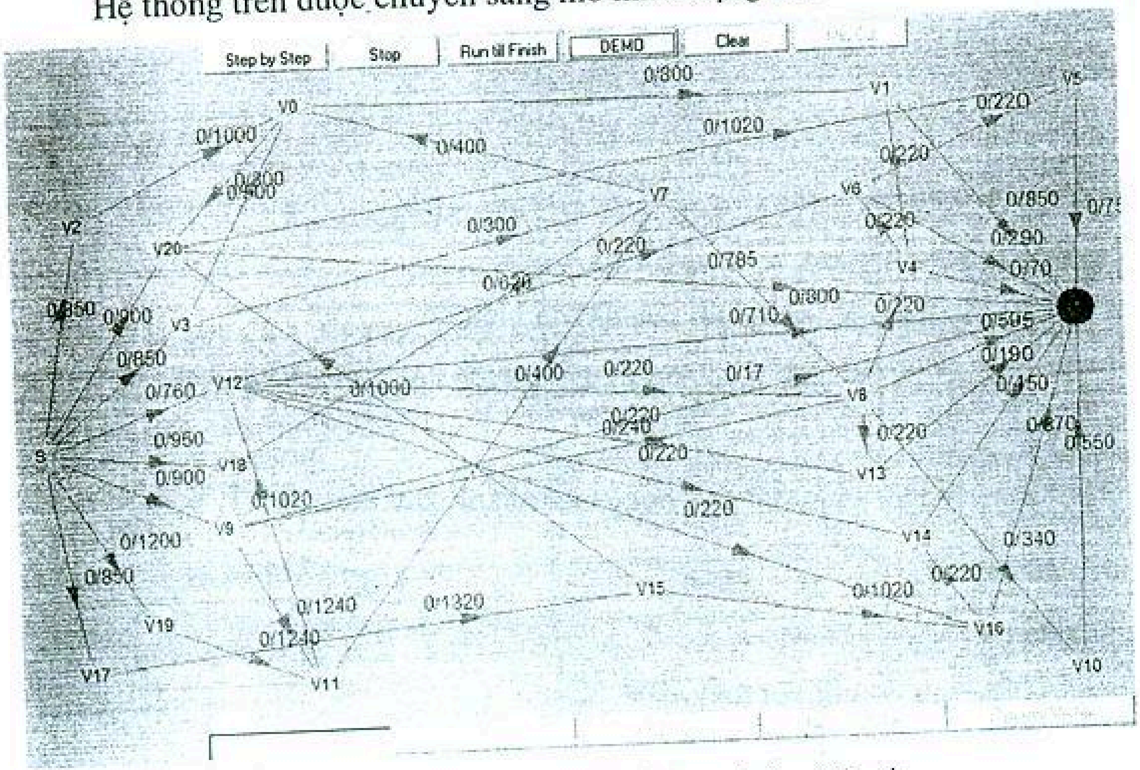
Ký hiệu:

- ND: nút đầu
- NC: nút cuối
- MF: máy phát
- TBA: trạm biến áp
- ĐD: đường dây
- P(0): giới hạn công suất truyền tải của nhánh hiện có
- P(1), P(2), P(3), P(4): giới hạn công suất truyền tải của các nhánh xây thêm thứ 1, 2, 3, 4
- C(1), C(2), C(3), C(4): chi phí xây dựng của nhánh xây thêm thứ 1, 2, 3, 4
- S: nút siêu nguồn
- T: nút siêu tải
- G: tập hợp các nút có nối với máy phát
  (ở đây G = {2, 3, 20, 9, 19, 17, 12, 18})
- L: tập hợp các nút có nối với phụ tải
  (ở đây L = {20, 5, 1, 8, 9, 10, 13, 12, 14, 16, 6, 4})
- Tổng công suất phát là:
  $G = 850 + 900 + 850 + 900 + 1200 + 850 + 760 + 950 = 7260$ (MW)
- Tổng công suất của phụ tải là:
  $L = 785 + 750 + 850 + 595 + 17 + 550 + 190 +$
  $+ 710 + 450 + 870 + 290 + 70 = 6127$ (MW)

Ta thấy rằng, tổng công suất phát lớn hơn tổng công suất của phụ tải, nhưng các phụ tải vẫn không được cung cấp đủ do lưới truyền tải của hệ thống. Bài toán đặt ra là phải xây dựng thêm các đường dây và trạm biến áp với chi phí nhỏ nhất.

4.2.2. CHUYỂN HỆ THỐNG SANG MÔ HÌNH BÀI TOÁN MẠNG:

Hệ thống trên được chuyển sang mô hình mạng như ở hình 4.6 dưới đây



Hình 4.6: Mô hình mạng của hệ thống 21 nút

Nút siêu nguồn S có màu vàng.

Nút siêu tải T có màu đỏ.

Các nút trên hệ thống được biểu diễn bằng các nút có cùng số

Các nhánh nối từ S đến các nút thuộc G = {2, 3, 20, 9, 19, 17, 12, 18} luôn có chiều đi từ S.

Các nhánh nối từ T đến các nút thuộc L= {20, 5, 1, 8, 9, 10, 13, 12, 14, 16, 6, 4} luôn có chiều kết thúc ở T.

Nhánh nối giữa các nút khác trong hệ thống có hai chiều, trên mỗi nhánh ghi giới hạn công suất truyền tải của nó và dòng công suất chạy trên nhánh. Ví dụ nhánh V20-V5 có 0/1020 nghĩa là dòng công suất trong nhánh hiện tại là 0(MW) và giới hạn truyền tải của nhánh là 1020(MW)

Sau khi tính được min cut, các nút nằm cùng tập với S sẽ có màu vàng, các nút nằm cùng tập với T sẽ có màu đỏ.

Trên phần mềm

Step by Step: chạy chương trình từng bước

Stop: dừng

Run still Finish: chạy chương trình đến khi kết thúc

Demo: chạy mạng demo

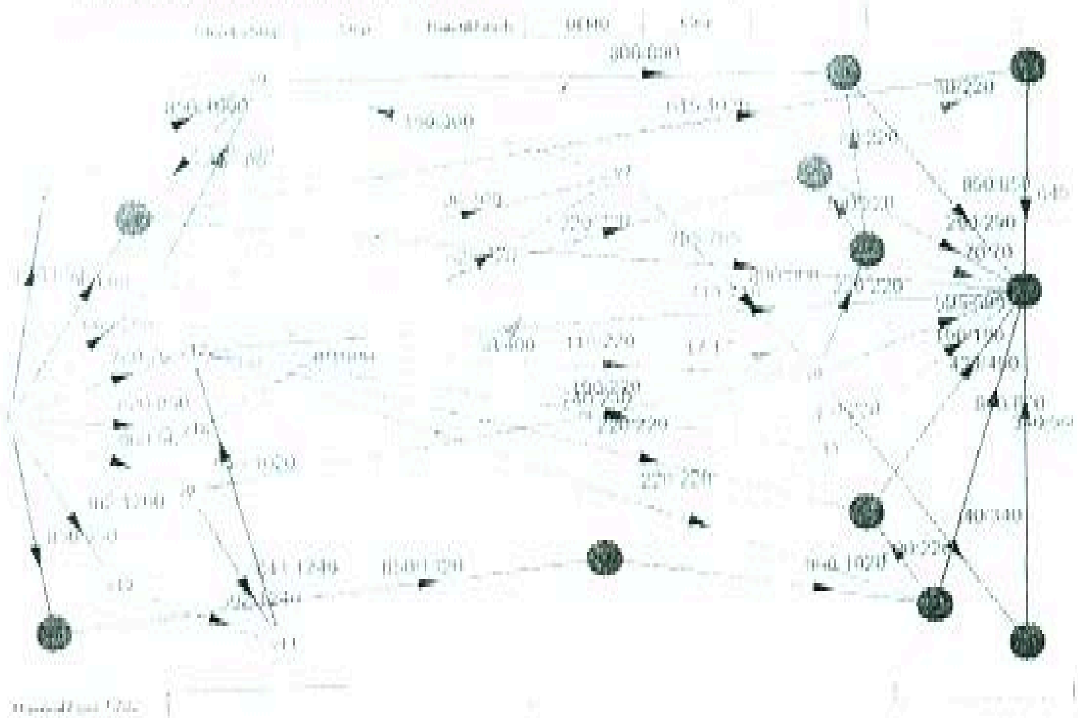
Clear: xoá sạch màn hình

Min cut: chương trình đưa ra min cut của mạng.

4.2.3. CÁC BƯỚC TÍNH TOÁN:

Bước 1:

Tính max flow (F_{max}) và min cut ban đầu của mạng ta có:



Hình 4.7: Kết quả tính max flow và min cut ban đầu của mạng

Kết quả thu được như sau:

$F_{max} = 5782\ (MW) < L$

$$\text{min cut} = \begin{array}{l} \{S\overset{900}{-}20, S\overset{850}{-}17, 8\overset{595}{-}T, 13\overset{190}{-}T, 12\overset{710}{-}T, 9\overset{17}{-}T, 0\overset{500}{-}20, 12\overset{220}{-}6, 8\overset{220}{-}4, 0\overset{800}{-}1, \\ 12\overset{220}{-}14, 12\overset{220}{-}16, 8\overset{340}{-}10\} \end{array}$$

Kiểm tra ta thấy tổng các dòng công suất chạy trong các nhánh thuộc min cut = 900 + 850 + 595 + 190 + 710 + 17 + 500 + 220 + 220 + 800 + 220 + 220 + 340 = 5782 (MW)

Max flow < Tổng phụ tải do đó ta phải tăng max flow lên ($F_{max} < L$). Để tăng max flow thì ta phải tăng giới hạn truyền tải của các nhánh nằm trong tập min cut. Do nguồn và tải là cố định nên trong tập min cut các nhánh nối với S và T không được xét để tăng giới hạn truyền tải.

Như vậy các nhánh có thể được gọi là nút cổ chai ở trong mạng sẽ là

$$\{0\overset{500}{-}20, 12\overset{220}{-}6, 8\overset{220}{-}4, 0\overset{800}{-}1, 12\overset{220}{-}14, 12\overset{220}{-}16, 8\overset{340}{-}10\}$$

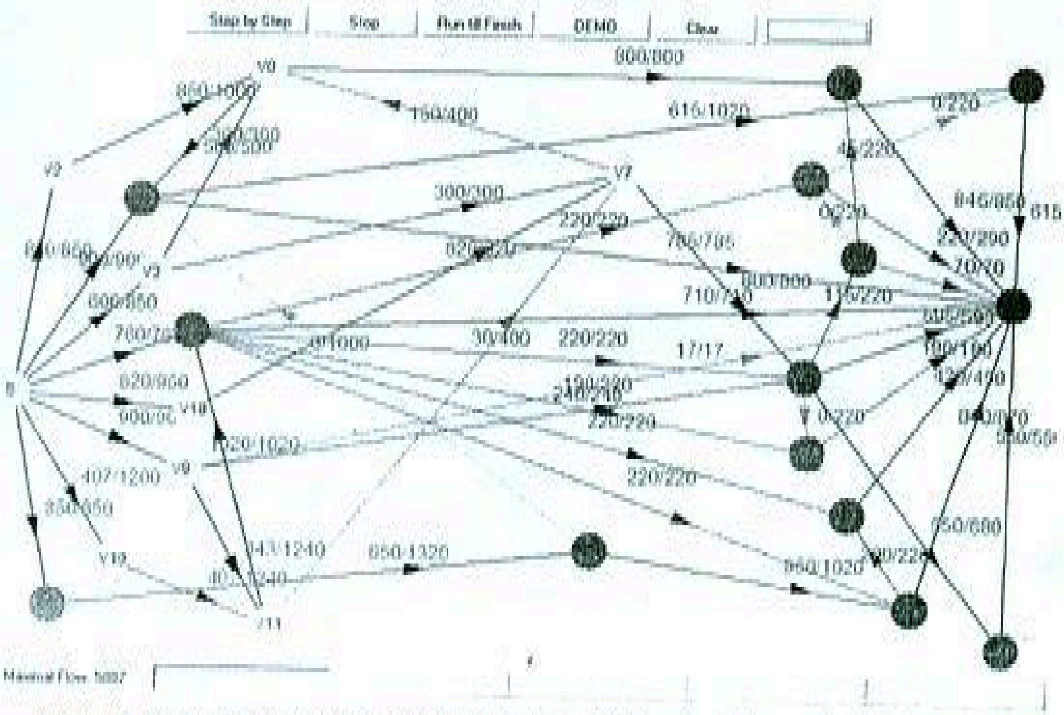
Chi phí xây dựng thêm các đường dây sẽ lần lượt là

$$\{0\overset{90}{-}20, 12\overset{88}{-}6, 8\overset{82}{-}4, 0\overset{151}{-}1, 12\overset{83}{-}14, 12\overset{69}{-}16, 8\overset{45}{-}10\}$$

Để tăng tốc độ hội tụ của bài toán ta sẽ chọn nhánh có chi phí xây dựng thêm các đường dây nhỏ nhất ở đây là nhánh 8-10 với C(1) = 45 (M$).

Bước 2:

Sau khi xây thêm nhánh 8-10 ta phải tính lại max flow và min cut của mạng và được kết quả như sau:



Hình 4.8: Kết quả tính max flow-min cut khi xây thêm tại nhánh 8-10

F_{max} = 5[illegible]7 (MW) < L

min cut = {S - 20, [illegible] $\overset{550}{-}$ 17, 9 $\overset{17}{-}$ T, 0 $\overset{500}{-}$ 20, 0 $\overset{800}{-}$ 1, 9 $\overset{240}{-}$ 8, 11 $\overset{1020}{-}$ 12, 7 $\overset{800}{-}$ 8}

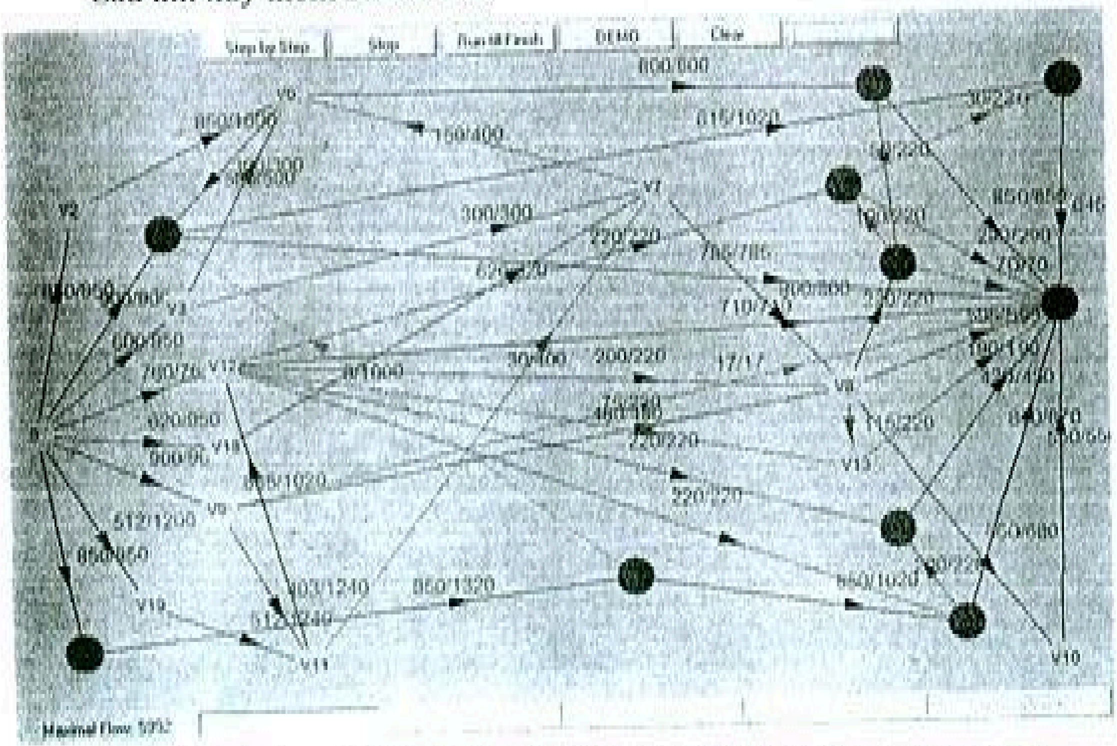
Chi phí xây dựng thêm các nhánh là:

{0 - 20, 0 - 1, 9 $\overset{81}{-}$ 8, 11 $\overset{124}{-}$ 12, [illegible]

Ta thấy chi phí xây thêm tại nhánh [illegible] nhỏ nhất do đó ta sẽ xây thêm nhánh [illegible] min cut [illegible] mạng mới được kết quả như sau:

Bước 3:

Sau khi xây thêm nhánh 9-8



Hình 4.9: Kết quả tính max flow-min cut khi xây thêm tại nhánh 9-8

F_{max} = 5992 (MW) < L

min cut =

$$\{S\overset{900}{-}20, S\overset{850}{-}17, 9\overset{17}{-}T, 8\overset{17}{-}T, 10\overset{17}{-}T, 13\overset{17}{-}T, 0\overset{800}{-}20, 0\overset{800}{-}1, 12\overset{220}{-}14, 12\overset{220}{-}6, 12\overset{220}{-}16, 8\overset{220}{-}4\}$$

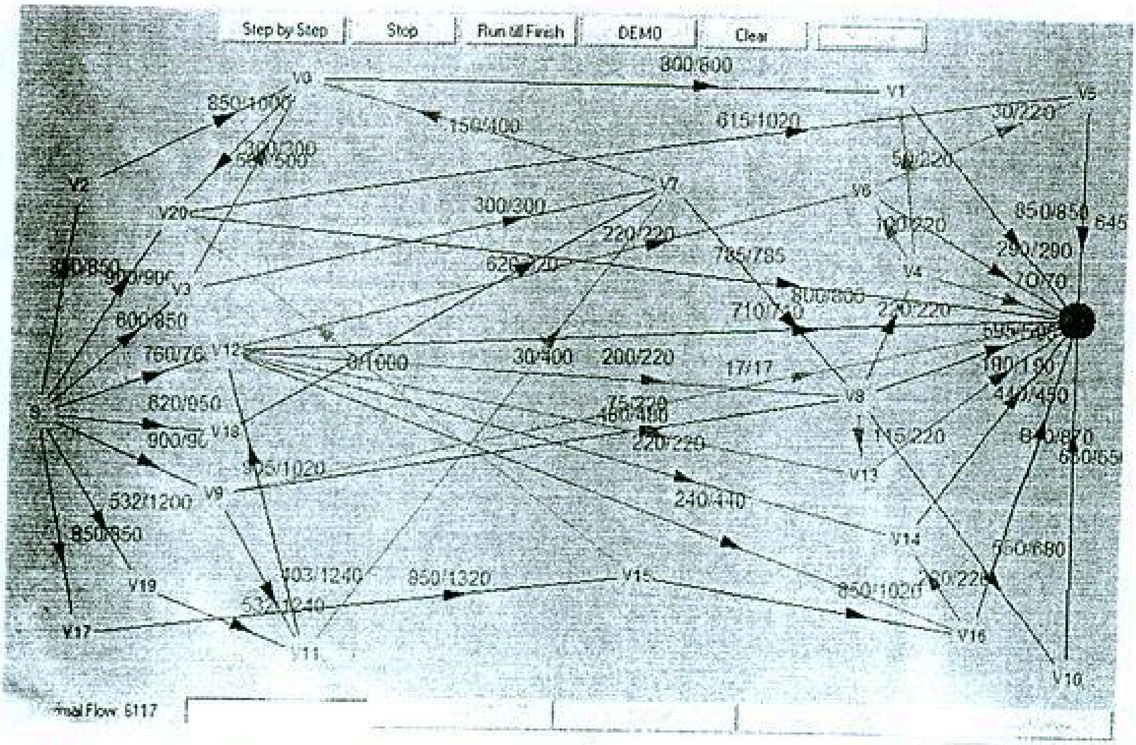
Chi phí xây dựng thêm các nhánh là:

$$\{0\overset{90}{-}20, 0\overset{151}{-}1, 12\overset{83}{-}14, 12\overset{88}{-}6, 12\overset{69}{-}16, 8\overset{83}{-}4\}$$

Ta thấy chi phí xây dựng thêm tại nhánh 12-16 là nhỏ nhất do đó ta sẽ xây dựng tại nhánh này và tính max flow-min cut thu được kết quả sau:

Bước 4:

Sau khi xây thêm nhánh12-16



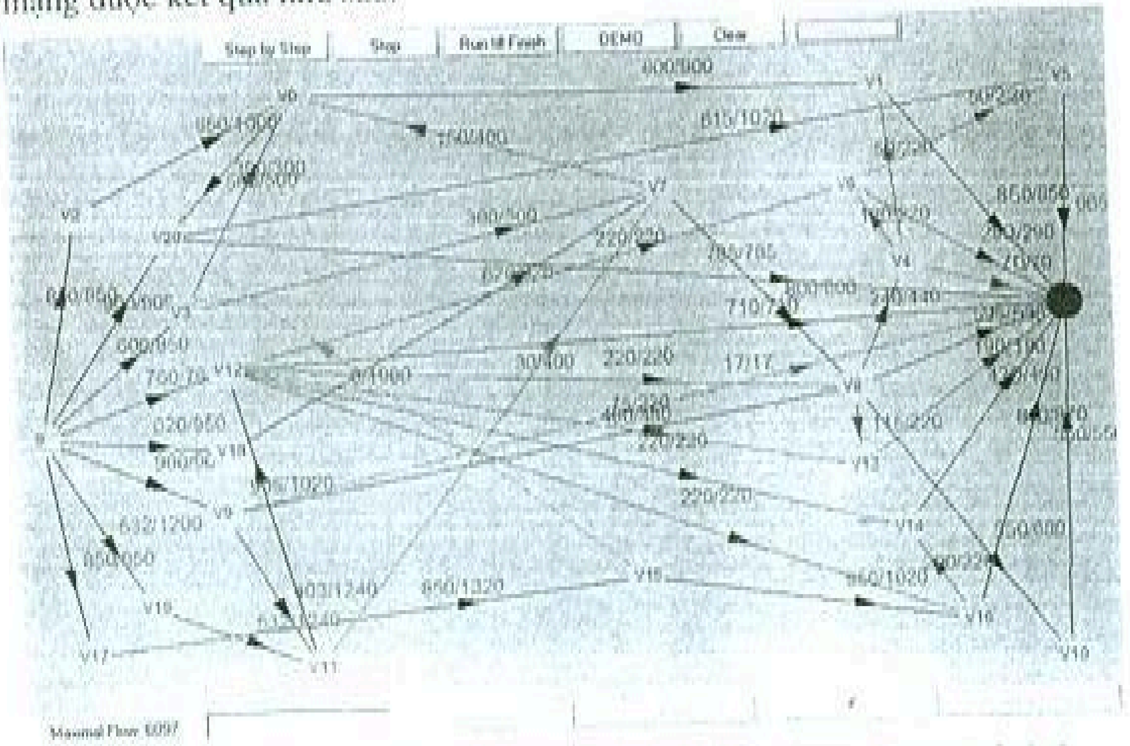
Hình ... Kết quả tính max flow-min cut khi xây thêm tại nhánh 12-16

F_{max} = 6117 (MW) < L

Nhìn vào hình vẽ ta thấy lát cắt nhỏ nhất sẽ đi qua các nhánh nối với T, như vậy có thể hiểu là không thể mở rộng thêm được nữa. Thế nhưng max flow vẫn nhỏ hơn tổng phụ tải. Như vậy việc xây dựng thêm tại nhánh này là không đạt yêu cầu cung cấp đủ cho phụ tải.

Bước 5:

Ta chuyển sang xét cho nhánh 8-4 và tính max flow – min cut cho mạng được kết quả như sau:



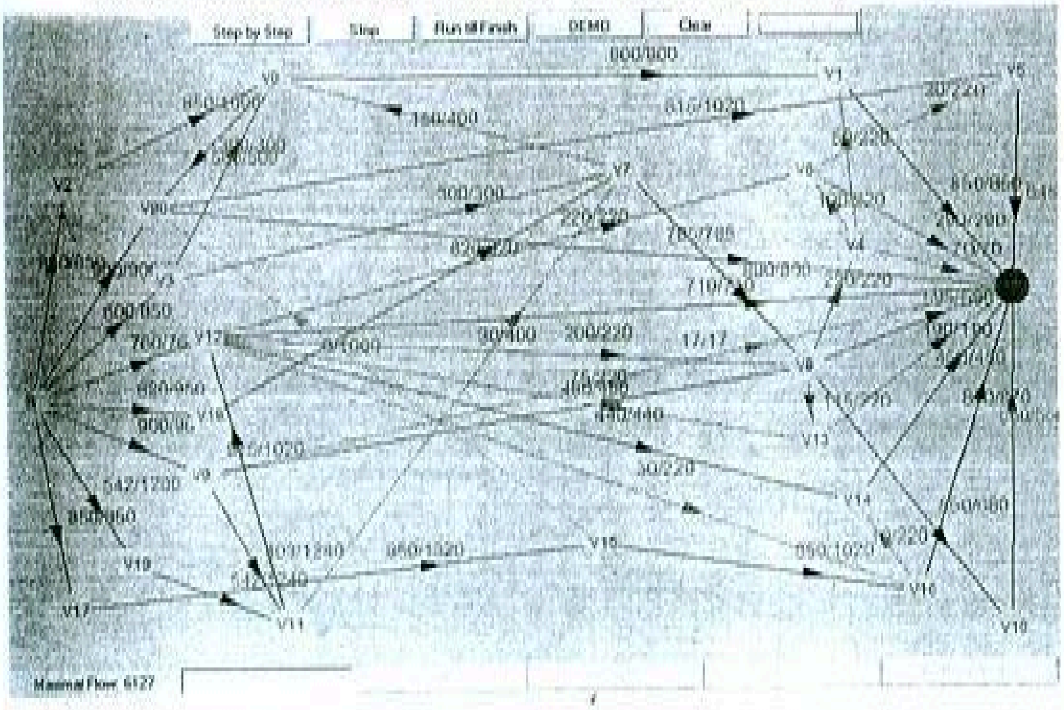
Hình 4.11: Kết quả tính max flow-min cut khi xây thêm tại nhánh 8-4

$F_{max} = 6097$ (MW) < L.

Cũng giống như trường hợp trước, việc xây thêm tại nhánh này là không thoả mãn ràng buộc cung cấp đủ công suất cho phụ tải.

Bước 6

Ta xét nhánh 12-14, tính max flow và min cut ta thu được kết quả sau:

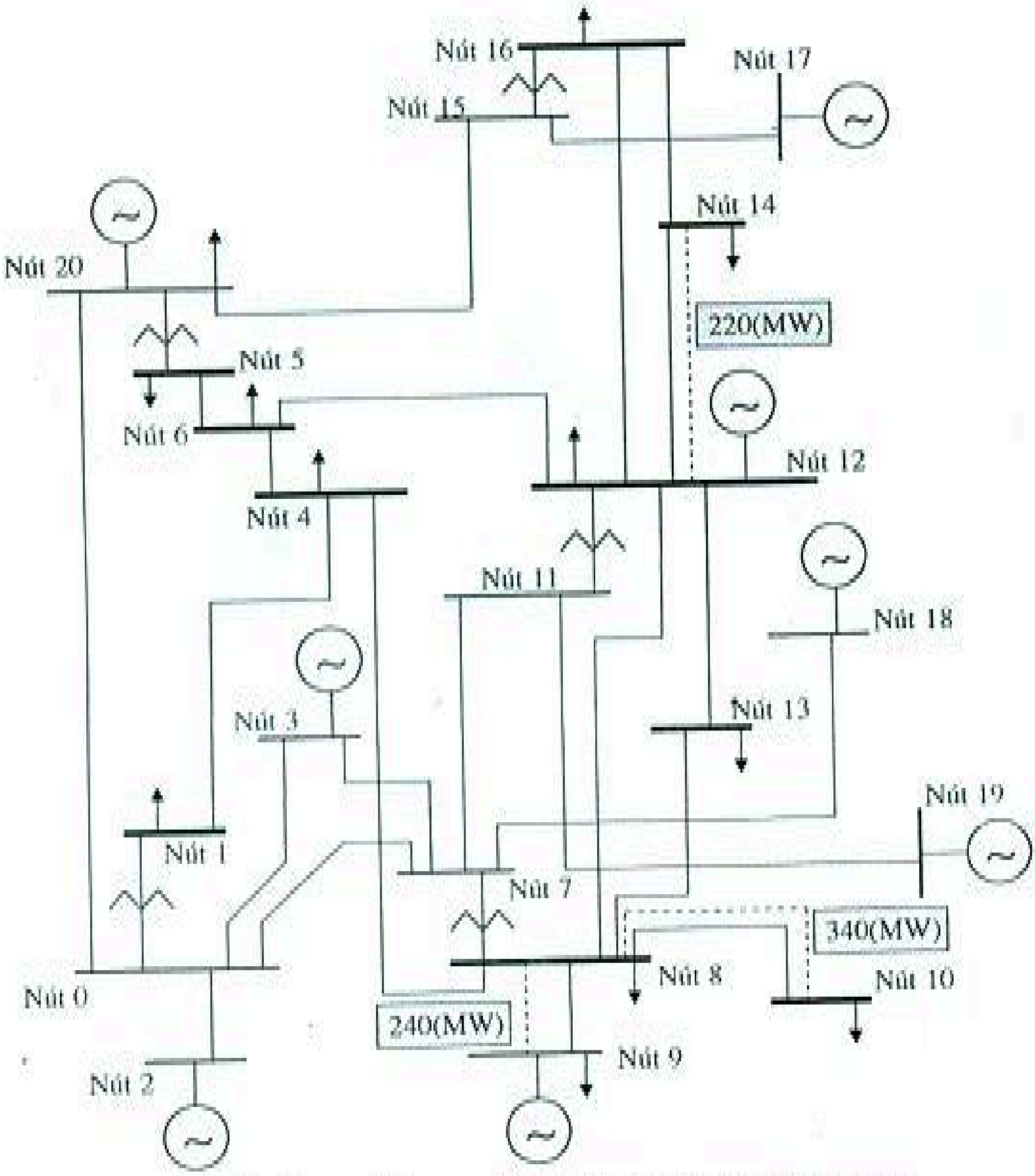


Hình 4.12: Kết quả tính max flow-min cut khi xây thêm tại nhánh 12-14

F_{max} = 6127 (MW) = L

Quá trình giải bài toán dừng lại tại đây.

4.3. KẾT QUẢ VÀ ĐÁNH GIÁ



Hình 4.13: Mạng điện sau khi đã được quy hoạch mở rộng

Bảng 4.4: Kết quả bài toán

Loại thiết bị	Phương án tối ưu	Vốn đầu tư (M$)
Máy biến áp	0	0
Đường dây	**8-10', 8-9', 12-14'**	209
Tổng vốn đầu tư		209

Nếu ta xây dựng tại mỗi nhánh 8-10, 8-9, 12-14 thêm một nhánh có công suất giới hạn tương ứng là 340(MW), 240(MW), 220(MW) thì ta sẽ thu được giá trị của hàm mục tiêu là nhỏ nhất, thoả mãn mọi ràng buộc của bài toán.

Tốc độ hội tụ của bài toán là chậm do khối lượng tính toán lớn. Bởi vì, khi thêm một nhánh vào hệ thống ta phải tính lại Max flow – Min cut và lại thu được một tập hợp các nhánh có thể được xây mới.

KẾT LUẬN

1. Bài toán quy hoạch mở rộng lưới điện truyền tải là bài toán tối ưu đa mục tiêu phức tạp với nội dung xác định xem ta sẽ xây dựng thêm ở đâu, khi nào, thiết bị mới loại gì để đáp ứng kịp với kết quả dự báo nguồn và tải với vốn đầu tư và chi phí vận hành nhỏ nhất.
2. Việc giải song song hai bài toán tối ưu vốn đầu và chi phí vận hành là rất phức tạp nên trên thực tế người ta tách hai bài toán ra giải riêng biệt.
3. Hiện nay có rất nhiều phương pháp giải bài toán quy hoạch mở rộng lưới điện truyền tải dựa trên các công cụ toán học như: các thuật toán trí tuệ nhân tạo (Tìm kiếm Tabu, Mô phỏng tôi), thuật toán nhánh và cận, lý thuyết trò chơi…
4. Phương pháp đề xuất trong luận văn này là sự kết hợp của phương pháp nhánh và cận với lý thuyết bài toán mạng thích hợp để giải bài toán quy hoạch mở rộng lưới điện truyền tải có đặc điểm: tĩnh, tối ưu vốn đầu tư. Phần mềm tìm Max flow – Min cut sẽ chỉ ra các tập hợp các nhánh cần phải mở rộng, phương pháp nhánh và cận sẽ tìm ra phương án có vốn đầu tư nhỏ nhất.
5. Nếu kết hợp phương pháp với các thuật toán trí tuệ nhân tạo như hệ mờ, phương pháp có khả năng ứng dụng vào quy hoạch hệ thống điện thực tế. Ngoài ra phương pháp cũng có thể áp dụng cho quy hoạch mạng lưới viễn thông hoặc đường giao thông, …

PHẦN PHỤ LỤC

1. Phụ lục 1:

FILE NGUỒN JAVA MÔDUL TÍNH TOÁN MAX FLOW – MIN CUT

```
public class ComputeMaxFlow {

  private BasicGraph Gr;
  private Vector residEdge;
  public TreeSearch flowBFS;

  public ComputeMaxFlow(BasicGraph basicgraph) {
    Gr = basicgraph;
    residEdge = new Vector();
    init();
  }
  public void init() {
    int i = Gr.nEdge();
    for(int j = 0; j < i; j++) {
      SuperEdge superedge = (SuperEdge)Gr.obtainEdge(j);
      superedge.defineFlow(0);
    }
    i = residEdge.size();
    for(int k = 0; k < i; k++) {
      SuperEdge superedge1 = (SuperEdge)residEdge.elementAt(k);
      superedge1.defineFlow(0);
    }
}
protected void goOneStep() {
  SuperVer superver5 = (SuperVer)Gr.returnSource();
  flowBFS = new TreeSearch(Gr);
  flowBFS.begin();
  SuperVer superver = (SuperVer)Gr.obtainSink();
  if(!isOver()) {
    int i = 0x7fffffff;
    SuperVer superver1 = superver;
    SuperVer superver3;
    do {
      superver3 = superver1.obtainFormer();
      SuperEdge superedge = (SuperEdge)superver3.obtainEdgeTo(superver1);
      i = Math.min(superedge.resCap(), i);
      superver1 = superver3;
    } while(superver3 != superver5);
```

```
        SuperVer superver4;
        for(SuperVer superver2 = superver; superver2 != superver5; superver2 =
superver4) {
            superver4 = superver2.obtainFormer();
            SuperEdge superedge1 = (SuperEdge)superver4.obtainEdgeTo(superver2);
            SuperEdge superedge2 = (SuperEdge)superver2.obtainEdgeTo(superver4);
            if(superedge2 == null) {
                superedge2 = new SuperEdge();
                superedge2.defineVerDestiny(superver4);
                superedge2.defineVerOrigin(superver2);
                residEdge.addElement(superedge2);
            }
                superedge1.defineFlow(superedge1.obtainFlow() + i);
                superedge2.defineFlow(-superedge1.obtainFlow());
            }
        }
    }
    public void initThisGraph() {
        int i = residEdge.size();
        for(int j = 0; j < i; j++) {
            SuperEdge superedge = (SuperEdge)residEdge.elementAt(j);
            SuperVer superver = (SuperVer)superedge.obtainVerOrigin();
            superver.increaseOutEdge(superedge);
            superver = (SuperVer)superedge.obtainVerDestiny();
            superver.increaseInEdge(superedge);
        }
    }

    public void resGraphObtained() {
        int i = residEdge.size();
        for(int j = 0; j < i; j++) {
            SuperEdge superedge = (SuperEdge)residEdge.elementAt(j);
            SuperVer superver = (SuperVer)superedge.obtainVerOrigin();
            superver.killEdgeOut(superedge);
            superver = (SuperVer)superedge.obtainVerDestiny();
            superver.killEdgeIn(superedge);
        }
    }
    public void stepsRun(int i) {
        initThisGraph();
        int j = 0;
        do {
            if(j >= i)
                break;
            goOneStep();
            if(isOver())
                break;
```

```
            j++;
        } while(true);
        resGraphObtained();
    }

    public void runUntilFinish() {
        initThisGraph();
        do
            goOneStep();
        while(!isOver());
        resGraphObtained();
    }
    public boolean isOver() {
        return ((SuperVer)Gr.obtainSink()).obtainFormer() == null;
    }
    public int thisFlow() {
        int i = 0;
        SuperVer superver = (SuperVer)Gr.obtainSink();
        int j = superver.nEdgeIn();
        for(int k = 0; k < j; k++) {
            SuperEdge superedge = (SuperEdge)superver.obtainEdgeIn(k);
            i += superedge.obtainFlow();
        }
        return i;
    }
}
```

2. Phụ lục 2:

FILE NGUỒN JAVA MÔDUL GIAO DIỆN VÀ LIÊN KẾT

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.EventObject;

public class Gui extends Applet
    implements ActionListener {

    private int speedofRun;
    private boolean isAlgWork;
    public BasicGraph gArray[];
    private int thisGraph;
    private static final Color BGDEFAULT;
    private static final int DEMOINDEX = 0;
    private static final int MYGRAPHINDEX = 1;
    private static final int NEWGRAPHS = 2;
    private Button ButtonClear;
    private Button ButtonStepByStep;
    private Button ButtonCustom;
    private Button BottonRunToFinish;
    private Button ButtonStop;
    public Button ButtonMinCut;
    private boolean stop;
    public Button buttonDelEdge;
    public Button buttonOK;
    public Button buttonCancel;
    public TextField fillInEdgeCAP;
    public Button buttonDelVer;
    protected GraphPanel canvaslet;
    private Panel bPanel;
    private Panel stPanel;
    private Label stLargestFlow;
    private ComputeMaxFlow alg;
    private static final int WDDEFAULT = 800;
    private static final int HGTDEFAULT = 600;
    private static final int MINIMUMHEIGHT = 600;
    private static final int MINIMUMWIDTH = 800;
    private static final int SPDEFAULT = 1;
    private static final int SPAUSE = 1000;
    public int currentVX;
    public int currentVY;
    public ExtendVer currentVer;
```

```
public Gui() {
    speedofRun = 1;
    isAlgWork = false;
    gArray = new BasicGraph[2];
    stop = false;
}

public void init() {
    gArray[0] = Demo();
    canvaslet = new GraphPanel(this, gArray[thisGraph]);
    gArray[1] = CreateNewGraph();
    thisGraph = 0;
    bPanel = new Panel();
    ButtonStepByStep = new Button("   Step by Step  ");
    BottonRunToFinish = new Button(" Run till Finish ");
    ButtonCustom = new Button("    Custom     ");
    ButtonStop = new Button("     Stop      ");
    ButtonClear = new Button("     Clear      ");
    ButtonMinCut = new Button("     Min Cut     ");
    bPanel.setLayout(new FlowLayout());
    bPanel.add(ButtonStepByStep);
    bPanel.add(ButtonStop);
    bPanel.add(BottonRunToFinish);
    bPanel.add(ButtonCustom);
    bPanel.add(ButtonClear);
    bPanel.add(ButtonMinCut);
    ButtonStop.setEnabled(false);
    ButtonMinCut.setEnabled(false);
    ButtonClear.setEnabled(true);
    stPanel = new Panel();
    buttonDelEdge = new Button("Remove Edge");
    buttonOK = new Button("     OK     ");
    buttonCancel = new Button("   Cancel  ");
    fillInEdgeCAP = new TextField(" ", 8);
    buttonDelVer = new Button("Remove Vertex");
    stPanel.setLayout(new GridLayout(1, 10));
    stLargestFlow = new Label(gArray[thisGraph].strGraph);
    stPanel.add(stLargestFlow);
    stPanel.add(fillInEdgeCAP);
    stPanel.add(buttonOK);
    stPanel.add(buttonCancel);
    stPanel.add(buttonDelEdge);
    stPanel.add(buttonDelVer);
    buttonOK.setEnabled(false);
    buttonCancel.setEnabled(false);
    buttonDelEdge.setEnabled(false);
    buttonDelVer.setEnabled(false);
```

```
        fillInEdgeCAP.setEnabled(false);
        setLayout(new BorderLayout());
        add("South", stPanel);
        add("Center", canvaslet);
        add("North", bPanel);
        resize(800, 600);
        setBackground(BGDEFAULT);
        ButtonStepByStep.addActionListener(this);
        buttonOK.addActionListener(this);
        fillInEdgeCAP.addActionListener(this);
        ButtonMinCut.addActionListener(this);
        buttonCancel.addActionListener(this);
        buttonDelEdge.addActionListener(this);
        buttonDelVer.addActionListener(this);
        ButtonClear.addActionListener(this);
        ButtonStop.addActionListener(this);
        ButtonCustom.addActionListener(this);
        BottonRunToFinish.addActionListener(this);
    }

    public void actionPerformed(ActionEvent actionevent) {
        boolean flag = false;
        if(actionevent.getSource().equals(ButtonStepByStep))
            handleEventStepByStep();
        if(actionevent.getSource() == buttonOK || actionevent.getSource() == fillInEdgeCAP)
            handleEventOKorFillEdge();
        if(actionevent.getSource().equals(ButtonMinCut))
            handleEventMINCUT(currentVer);
        if(actionevent.getSource() == buttonCancel)
            handleEventCancel();
        if(actionevent.getSource() == buttonDelEdge)
            handleEventDELEDGE();
        if(actionevent.getSource() == buttonDelVer)
            handleEventDelVer();
        if(actionevent.getSource().equals(ButtonClear) && !isAlgWork)
            handleEventClear();
        if(actionevent.getSource().equals(ButtonStop))
            handleEventStop();
        if(actionevent.getSource().equals(ButtonCustom) && !isAlgWork)
            handleEventCustom();
        if(actionevent.getSource().equals(BottonRunToFinish)) {
            handleEventRunToFinish();
            ButtonMinCut.setEnabled(true);
        }
    }

    protected BasicGraph Demo() {
```

```
BasicGraph basicgraph = CreateNewGraph();
ExtendVer extendver = (ExtendVer)basicgraph.obtainSink();
ExtendVer extendver1 = (ExtendVer)basicgraph.returnSource();
Point point = extendver.position();
Point point1 = extendver1.position();
int i = point.x - point1.x;
ExtendVer extendver2 = new ExtendVer(point1.x + i / 5, point.y - 120);
ExtendVer extendver3 = new ExtendVer(point1.x + i / 5, point.y);
ExtendVer extendver4 = new ExtendVer(point1.x + i / 5, point.y + 120);
ExtendVer extendver5 = new ExtendVer(point1.x + (2 * i) / 5, point.y - 120);
ExtendVer extendver6 = new ExtendVer(point1.x + (2 * i) / 5, point.y);
ExtendVer extendver7 = new ExtendVer(point1.x + (2 * i) / 5, point.y + 120);
ExtendVer extendver8 = new ExtendVer(point1.x + (3 * i) / 5, point.y - 120);
ExtendVer extendver9 = new ExtendVer(point1.x + (3 * i) / 5, point.y);
ExtendVer extendver10 = new ExtendVer(point1.x + (3 * i) / 5, point.y + 120);
ExtendVer extendver11 = new ExtendVer(point1.x + (4 * i) / 5, point.y - 120);
ExtendVer extendver12 = new ExtendVer(point1.x + (4 * i) / 5, point.y);
ExtendVer extendver13 = new ExtendVer(point1.x + (4 * i) / 5, point.y + 120);
basicgraph.increaseVer(extendver2);
basicgraph.increaseVer(extendver3);
basicgraph.increaseVer(extendver4);
basicgraph.increaseVer(extendver5);
basicgraph.increaseVer(extendver6);
basicgraph.increaseVer(extendver7);
basicgraph.increaseVer(extendver8);
basicgraph.increaseVer(extendver9);
basicgraph.increaseVer(extendver10);
basicgraph.increaseVer(extendver11);
basicgraph.increaseVer(extendver12);
basicgraph.increaseVer(extendver13);
ExtendEdge extendedge = new ExtendEdge(extendver1, extendver2, 10);
ExtendEdge extendedge1 = new ExtendEdge(extendver1, extendver3, 9);
ExtendEdge extendedge2 = new ExtendEdge(extendver1, extendver4, 9);
ExtendEdge extendedge3 = new ExtendEdge(extendver2, extendver3, 8);
ExtendEdge extendedge4 = new ExtendEdge(extendver2, extendver5, 9);
ExtendEdge extendedge5 = new ExtendEdge(extendver2, extendver6, 4);
ExtendEdge extendedge6 = new ExtendEdge(extendver3, extendver4, 9);
ExtendEdge extendedge7 = new ExtendEdge(extendver3, extendver6, 8);
ExtendEdge extendedge8 = new ExtendEdge(extendver3, extendver7, 5);
ExtendEdge extendedge9 = new ExtendEdge(extendver4, extendver7, 6);
ExtendEdge extendedge10 = new ExtendEdge(extendver5, extendver8, 8);
ExtendEdge extendedge11 = new ExtendEdge(extendver6, extendver5, 1);
ExtendEdge extendedge12 = new ExtendEdge(extendver6, extendver8, 10);
ExtendEdge extendedge13 = new ExtendEdge(extendver6, extendver9, 9);
ExtendEdge extendedge14 = new ExtendEdge(extendver6, extendver10, 2);
ExtendEdge extendedge15 = new ExtendEdge(extendver7, extendver6, 1);
ExtendEdge extendedge16 = new ExtendEdge(extendver7, extendver10, 7);
```

```
    ExtendEdge extendedge17 = new ExtendEdge(extendver8, extendver11, 10);
    ExtendEdge extendedge18 = new ExtendEdge(extendver9, extendver8, 1);
    ExtendEdge extendedge19 = new ExtendEdge(extendver8, extendver12, 3);
    ExtendEdge extendedge20 = new ExtendEdge(extendver9, extendver12, 4);
    ExtendEdge extendedge21 = new ExtendEdge(extendver9, extendver13, 5);
    ExtendEdge extendedge22 = new ExtendEdge(extendver10, extendver9, 1);
    ExtendEdge extendedge23 = new ExtendEdge(extendver10, extendver13, 3);
    ExtendEdge extendedge24 = new ExtendEdge(extendver11, extendver12, 2);
    ExtendEdge extendedge25 = new ExtendEdge(extendver11, extendver, 5);
    ExtendEdge extendedge26 = new ExtendEdge(extendver12, extendver13, 12);
    ExtendEdge extendedge27 = new ExtendEdge(extendver12, extendver, 8);
    ExtendEdge extendedge28 = new ExtendEdge(extendver13, extendver, 12);
    ExtendEdge extendedge29 = new ExtendEdge(extendver4, extendver6, 2);
    ExtendEdge extendedge30 = new ExtendEdge(extendver9, extendver11, 6);
    basicgraph.increaseEdge(extendedge);
    basicgraph.increaseEdge(extendedge1);
    basicgraph.increaseEdge(extendedge2);
    basicgraph.increaseEdge(extendedge3);
    basicgraph.increaseEdge(extendedge4);
    basicgraph.increaseEdge(extendedge5);
    basicgraph.increaseEdge(extendedge6);
    basicgraph.increaseEdge(extendedge7);
    basicgraph.increaseEdge(extendedge8);
    basicgraph.increaseEdge(extendedge9);
    basicgraph.increaseEdge(extendedge10);
    basicgraph.increaseEdge(extendedge11);
    basicgraph.increaseEdge(extendedge12);
    basicgraph.increaseEdge(extendedge13);
    basicgraph.increaseEdge(extendedge14);
    basicgraph.increaseEdge(extendedge15);
    basicgraph.increaseEdge(extendedge16);
    basicgraph.increaseEdge(extendedge17);
    basicgraph.increaseEdge(extendedge18);
    basicgraph.increaseEdge(extendedge19);
    basicgraph.increaseEdge(extendedge20);
    basicgraph.increaseEdge(extendedge21);
    basicgraph.increaseEdge(extendedge22);
    basicgraph.increaseEdge(extendedge23);
    basicgraph.increaseEdge(extendedge24);
    basicgraph.increaseEdge(extendedge25);
    basicgraph.increaseEdge(extendedge26);
    basicgraph.increaseEdge(extendedge27);
    basicgraph.increaseEdge(extendedge28);
    basicgraph.increaseEdge(extendedge29);
    basicgraph.increaseEdge(extendedge30);
    return basicgraph;
}
```

```
public BasicGraph CreateNewGraph() {
   BasicGraph basicgraph = new BasicGraph();
   if(canvaslet != null) {
      Dimension dimension = canvaslet.getSize();
      basicgraph.defineSource(new ExtendVer(99, 258));
      basicgraph.defineSink(new ExtendVer(698, 258));
   } else {
      basicgraph.defineSource(new ExtendVer(99, 258));
      basicgraph.defineSink(new ExtendVer(698, 258));
   }
   return basicgraph;
}

public void refresh(Graphics g) {
   if(getSize().width < 800 || getSize().height < 600)
      resize(800, 600);
   super.update(g);
}

public void handleEventMINCUT(ExtendVer extendver) {
   alg = new ComputeMaxFlow(gArray[thisGraph]);
   alg.runUntilFinish();
   ExtendVer extendver1 = (ExtendVer)gArray[thisGraph].source;
   ExtendVer extendver2 = (ExtendVer)gArray[thisGraph].obtainSink();
   for(int i = 0; i < gArray[thisGraph].nVer(); i++) {
      ExtendVer extendver3 = (ExtendVer)gArray[thisGraph].vers.elementAt(i);
      extendver3.changeColor(Color.red);
   }

   Vector vector = new Vector();
   vector.add(extendver1);
   minCut(extendver1, extendver2, vector);
   ButtonMinCut.setEnabled(false);
   buttonDelVer.setEnabled(false);
   canvaslet.reDraw();
}

public void minCut(ExtendVer extendver, ExtendVer extendver1, Vector vector) {
   ExtendVer extendver2 = extendver;
   int i = extendver2.outEdge.size();
   int j = extendver2.inEdge.size();
   int k = extendver2.inEdge.size() + extendver2.outEdge.size();
   extendver2.changeColor(Color.orange);
   Vector vector1 = new Vector();
   for(int l = 0; l < i; l++) {
      SuperEdge superedge = (SuperEdge)extendver.obtainEdgeOut(l);
```

```
        if(superedge.resCap() > 0) {
            ExtendVer extendver3 = (ExtendVer)superedge.obtainVerDestiny();
            addElement(extendver3, vector1, vector);
        }
    }

    for(int i1 = 0; i1 < j; i1++) {
        SuperEdge superedge1 = (SuperEdge)extendver.obtainEdgeIn(i1);
        if(!isExistInVec((ExtendVer)superedge1.obtainVerOrigin(),         vector)         &&
Math.abs(superedge1.flow) > 0) {
            ExtendVer extendver4 = (ExtendVer)superedge1.obtainVerOrigin();
            addElement(extendver4, vector1, vector);
        }
    }

    for(int j1 = 0; j1 < vector1.size(); j1++)
        minCut((ExtendVer)vector1.elementAt(j1), extendver1, vector);

}

public boolean isExistInVec(ExtendVer extendver, Vector vector) {
    boolean flag = false;
    int i = 0;
    do {
        if(i >= vector.size())
            break;
        ExtendVer extendver1 = (ExtendVer)vector.elementAt(i);
        if(extendver1.center.x    ==    extendver.center.x    &&    extendver1.center.y    ==
extendver.center.y) {
            flag = true;
            break;
        }
        flag = false;
        i++;
    } while(true);
    return flag;
}

public void addElement(ExtendVer extendver, Vector vector, Vector vector1) {
    boolean flag = false;
    int i = 0;
    do {
        if(i >= vector1.size())
            break;
        ExtendVer extendver1 = (ExtendVer)vector1.elementAt(i);
        if(extendver1.center.x    ==    extendver.center.x    &&    extendver1.center.y    ==
extendver.center.y) {
```

```
            flag = true;
            break;
          }
          flag = false;
          i++;
        } while(true);
        if(!flag) {
          vector.add(extendver);
          vector1.add(extendver);
        }
      }

      public boolean handleEventStepByStep() {
        int i = 0;
        BottonRunToFinish.setEnabled(false);
        ButtonCustom.setEnabled(false);
        ButtonStop.setEnabled(true);
        ButtonClear.setEnabled(false);
        if(!isAlgWork) {
          alg = new ComputeMaxFlow(gArray[thisGraph]);
          alg.runUntilFinish();
          int j = gArray[thisGraph].LargestEdgeFlow();
          canvaslet.defineMaxEdgeFlow(j);
          gArray[thisGraph].largestEdge = j;
          alg.init();
          canvaslet.reDraw();
          isAlgWork = true;
          canvaslet.setEnabled(false);
          gArray[thisGraph].isModified = false;
        }
        int k = speedofRun;
        canvaslet.AugPathMarking(true);
        i = alg.thisFlow();
        Thread thread = new Thread();
        do {
          alg.stepsRun(1);
          canvaslet.reDraw();
          if(k > 1)
            try {
              Thread _tmp = thread;
              Thread.sleep(1000L);
            }
            catch(InterruptedException interruptedexception) { }
          k--;
        } while(!alg.isOver() && k > 0);
        if(alg.isOver()) {
          isAlgWork = false;
```

```
            canvaslet.setEnabled(true);
            canvaslet.AugPathMarking(false);
            gArray[thisGraph].strGraph = "Maximal Flow: " + (new
Integer(alg.thisFlow())).toString();
            BottonRunToFinish.setEnabled(false);
            stLargestFlow.setAlignment(1);
            stLargestFlow.setText(gArray[thisGraph].strGraph);
            ButtonStepByStep.setEnabled(false);
            ButtonMinCut.setEnabled(true);
            ButtonClear.setEnabled(true);
        } else {
            gArray[thisGraph].strGraph = "Flow Growth = " + (new Integer(alg.thisFlow() -
i)).toString();
            stLargestFlow.setAlignment(1);
            stLargestFlow.setText(gArray[thisGraph].strGraph);
        }
        canvaslet.reDraw();
        return true;
    }

    public boolean handleEventCancel() {
        buttonOK.setEnabled(false);
        buttonCancel.setEnabled(false);
        buttonDelEdge.setEnabled(false);
        fillInEdgeCAP.setText(" ");
        fillInEdgeCAP.setEnabled(false);
        canvaslet.reDraw();
        return true;
    }

    public boolean handleEventDELEDGE() {
        ExtendEdge extendedge = canvaslet.edgeHeld;
        gArray[thisGraph].removeEdge(extendedge);
        buttonOK.setEnabled(false);
        buttonCancel.setEnabled(false);
        buttonDelEdge.setEnabled(false);
        buttonDelVer.setEnabled(false);
        fillInEdgeCAP.setText(" ");
        fillInEdgeCAP.setEnabled(false);
        canvaslet.reDraw();
        return true;
    }

    public boolean handleEventOKorFillEdge() {
        ExtendEdge extendedge = canvaslet.edgeHeld;
        String s = fillInEdgeCAP.getText();
        if(s != null) {
```

```
        int i;
        try {
            i = Integer.parseInt(s);
        }
        catch(NumberFormatException numberformatexception) {
            i = extendedge.obtainCap();
        }
        if(i > 0) {
            extendedge.defineCap(i);
            buttonOK.setEnabled(false);
            buttonCancel.setEnabled(false);
            buttonDelEdge.setEnabled(false);
            buttonDelVer.setEnabled(false);
            fillInEdgeCAP.setText(" ");
            fillInEdgeCAP.setEnabled(false);
            canvaslet.reDraw();
        }
    }
    return true;
}

public void handleEventDelVer() {
    int i = gArray[thisGraph].nVer();
    int j = 0;
    do {
        if(j >= i)
            break;
        ExtendVer extendver = (ExtendVer)gArray[thisGraph].returnVer(j);
        if(extendver.isInside(currentVX, currentVY)) {
            gArray[thisGraph].removeVertex(extendver);
            canvaslet.verHeld = null;
            break;
        }
        j++;
    } while(true);
    buttonOK.setEnabled(false);
    buttonCancel.setEnabled(false);
    buttonDelEdge.setEnabled(false);
    buttonDelVer.setEnabled(false);
    fillInEdgeCAP.setText(" ");
    fillInEdgeCAP.setEnabled(false);
    canvaslet.reDraw();
}

public boolean handleEventClear() {
    if(!gArray[thisGraph].isModified) {
        gArray[thisGraph].largestEdge = 0;
```

```
        canvaslet.defineMaxEdgeFlow(0);
        gArray[thisGraph].isModified = true;
        alg = new ComputeMaxFlow(gArray[thisGraph]);
        BasicGraph _tmp = gArray[thisGraph];
        gArray[thisGraph].strGraph = "";
        stLargestFlow.setText(gArray[thisGraph].strGraph);
    }
    if(thisGraph == 1)
        gArray[thisGraph] = CreateNewGraph();
    else
        gArray[thisGraph] = Demo();
    canvaslet.defineThisGraph(gArray[thisGraph]);
    canvaslet.reDraw();
    return true;
}

public void handleEventStop() {
    stop = true;
    alg.init();
    canvaslet.AugPathMarking(false);
    gArray[thisGraph].strGraph = "";
    stLargestFlow.setText(gArray[thisGraph].strGraph);
    isAlgWork = false;
    BottonRunToFinish.setEnabled(true);
    ButtonCustom.setEnabled(true);
    ButtonStepByStep.setEnabled(true);
    canvaslet.setEnabled(true);
    canvaslet.reDraw();
}

public boolean handleEventCustom() {
    if(thisGraph == 1) {
        thisGraph = 0;
        ButtonCustom.setLabel("Custom");
    } else {
        thisGraph = 1;
        ButtonCustom.setLabel(" DEMO ");
    }
    canvaslet.defineMaxEdgeFlow(gArray[thisGraph].largestEdge);
    canvaslet.defineThisGraph(gArray[thisGraph]);
    stLargestFlow.setText(gArray[thisGraph].strGraph);
    canvaslet.reDraw();
    return true;
}

public boolean handleEventRunToFinish() {
    int i = 0;
```

```
ButtonStepByStep.setEnabled(false);
ButtonCustom.setEnabled(false);
ButtonStop.setEnabled(false);
ButtonClear.setEnabled(false);
if(!isAlgWork) {
   alg = new ComputeMaxFlow(gArray[thisGraph]);
   alg.runUntilFinish();
   int j = gArray[thisGraph].LargestEdgeFlow();
   canvaslet.defineMaxEdgeFlow(j);
   gArray[thisGraph].largestEdge = j;
   alg.init();
   canvaslet.reDraw();
   isAlgWork = true;
   canvaslet.setEnabled(false);
   gArray[thisGraph].isModified = false;
}
int k = speedofRun;
k = 0x7fffffff;
canvaslet.AugPathMarking(true);
i = alg.thisFlow();
Thread thread = new Thread();
do {
   alg.stepsRun(1);
   canvaslet.reDraw();
   if(k > 1)
      try {
         Thread _tmp = thread;
         Thread.sleep(1000L);
      }
      catch(InterruptedException interruptedexception) { }
   k--;
} while(!alg.isOver() && k > 0);
if(alg.isOver()) {
   isAlgWork = false;
   canvaslet.setEnabled(true);
   canvaslet.AugPathMarking(false);
   gArray[thisGraph].strGraph = "Maximal Flow: " + (new
Integer(alg.thisFlow())).toString();
   ButtonStepByStep.setEnabled(true);
   ButtonCustom.setEnabled(true);
   ButtonStop.setEnabled(true);
   ButtonClear.setEnabled(true);
   stLargestFlow.setAlignment(1);
   stLargestFlow.setText(gArray[thisGraph].strGraph);
} else {
   gArray[thisGraph].strGraph = "Flow Growth = " + (new Integer(alg.thisFlow() -
i)).toString();
```

```
            stLargestFlow.setAlignment(1);
            stLargestFlow.setText(gArray[thisGraph].strGraph);
        }
        canvaslet.reDraw();
        return true;
    }

    static {
        BGDEFAULT = Color.lightGray;
    }
}
```

TÀI LIỆU THAM KHẢO

iếng Việt

TS. Nguyễn Lân Tráng (2003), "Quy hoạch hệ thống điện", Nhà xuất bản Khoa học và kỹ thuật, Hà Nội.

PGS. Bùi Minh Trí (2001), "Quy hoạch toán học", Nhà xuất bản Khoa học và kỹ thuật.

Ngọc Anh Thư Press (2002), "Giáo trình thuật toán", Nhà xuất bản Thống kê

iếng Anh

. R. Romero, R. A. Gallego, A. Monticelli, "Transmission System Expansion Planning by Simulated Annealing," IEEE Trans. Power Syst., vol. 11, pp. 364–369, Feb. 1996.

. Contreras J., Wu F.F., "A Kernel-Oriented Algorithm for Transmission Expansion Planning," IEEE Transactions, Vol.15, pp.1434 – 1440, Nov.2000.