

# MỘT THUẬT TOÁN GEN CHO THIẾT KẾ TOPOLOGY MẠNG CÓ KHẢ NĂNG HỒI PHỤC

NGUYỄN QUÝ MINH HIỀN, PHẠM QUỐC HUY

*Viện Khoa học Kỹ thuật Bưu điện*

**Abstract.** Today, telecommunication technologies and services are deploying rapidly in NGN (Next Generation Network) networks. Quality of Service (QoS) problems in NGN networks are addressed in network designing, deploying, maintaining phases. In network designing phase, one of QoS problems is designing survivable networks to assure that services are not disrupted when a failure happens. On an ATM, MPLS, GMPLS based-on network technologies, there are many approaches for designing survivable networks. This paper introduces a genetic algorithm for topological design of survivable networks by a repetitively-restorable method.

**Tóm tắt.** Hiện nay, với việc triển khai công nghệ mạng NGN, các dịch vụ thông tin truyền thông đang được phát triển rất nhanh chóng với nhiều loại hình dịch vụ. Một trong những vấn đề đặt ra đó là cần phải thiết kế, triển khai mạng có khả năng đảm bảo chất lượng dịch vụ cho các loại dịch vụ triển khai trên đó. Một trong những yêu cầu về chất lượng dịch vụ đặt ra đối với việc thiết kế mạng, đó là mạng phải có khả năng hồi phục để đảm bảo cung cấp dịch vụ với chất lượng như đã cam kết với khách hàng ngay trong những trường hợp mạng có sự cố. Trên cơ sở các công nghệ mạng như ATM, MPLS, GMPLS, có nhiều cách thức tiếp cận để đảm bảo khả năng hồi phục của mạng. Bài báo này giới thiệu một thuật toán thiết kế topology với khả năng hồi phục dựa trên phương pháp hồi phục lặp trên nhiều đường hồi phục.

## 1. GIỚI THIỆU

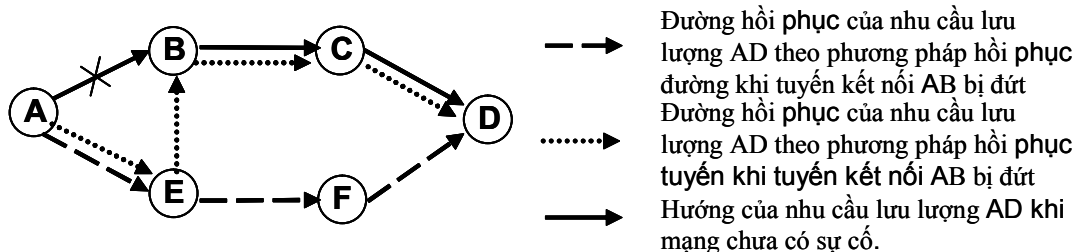
Trên mạng trong quá trình khai thác có thể xuất hiện những sự cố gây trở ngại cho quá trình cung cấp dịch vụ. Những sự cố có thể là đứt cáp, mất nguồn điện, hỏng hóc thiết bị, nằm ngoài sự điều khiển của người vận hành. Một số sự cố khác nằm trong điều hành của người đó là khởi động, cấu hình lại hệ thống, nâng cấp hệ thống, mất điều khiển hệ thống. Những sự cố như vậy có thể quy thành hai nhóm: sự cố tuyến và sự cố nút. Những sự cố tuyến xảy ra khi sự cố dẫn đến tuyến làm việc không chính xác hoặc ngừng các luồng thông tin trên tuyến. Những sự cố nút xuất hiện gây nên sự ngừng trệ thông tin trên nút đó và cho các tuyến kết nối đến nút này. Thực tế trên mạng có thể xuất hiện đồng thời nhiều sự cố, tuy nhiên, xác suất xuất hiện đồng thời hai sự cố trên mạng là rất thấp. Chính vì vậy phần lớn các nghiên cứu tập trung vào việc giải quyết sự cố theo giả thiết tại mỗi thời điểm chỉ có một sự cố.

Một chiến lược hồi phục có thể triển khai ở các lớp khác nhau theo quan điểm giao thức mạng OSI như hồi phục tại lớp vật lý, kết nối, mạng hoặc đồng thời trên nhiều lớp. Tại mỗi lớp, các chiến lược hồi phục có những đặc tính khác nhau. Với lớp vật lý, các chiến lược

hồi phục có thời gian hồi phục nhanh nhất khoảng 50ms. Các chiến lược hồi phục tại lớp kết nối thường dùng cho các mạng hướng kết nối như ATM và MPLS. Các kiến trúc mạng chuyển mạch nhãn thiết lập *đường dẫn chuyển mạch nhãn LSP* (Label Switch Path) chuyển tiếp lưu lượng giữa nút nguồn và đích. Mục tiêu của các chiến lược tại lớp này là hồi phục phần lưu lượng trên các LSP mà bị ảnh hưởng bởi sự cố. Các chiến lược hồi phục tại lớp kết nối có nhiều điểm linh hoạt như không phân nhỏ mạng thành các phần nhỏ hơn hoặc không phải giả thiết rằng topology của mạng phải có cấu trúc. Các tuyến và các nút có thể tham gia trong chiến lược hồi phục. Sử dụng các thuật toán hoạch định trước, thường dùng để xác định cách thức tốt nhất định tuyến lưu lượng khi có sự cố xảy ra, ví dụ như trong chiến lược hồi phục đường, sẽ sử dụng một đường khác để chuyển lưu lượng khi xảy ra sự cố mạng. Mặc dù thời gian cho việc định tuyến lại lưu lượng trong trường hợp thứ nhất lớn hơn trường hợp thứ hai, nhưng các phương pháp trong [2, 5, 8, 12] chỉ ra rằng hiệu suất sử dụng băng thông hồi phục tại lớp kết nối lớn hơn đáng kể so với khi hồi phục tại lớp vật lý.

Các chiến lược hồi phục lớp kết nối được dựa trên các kiến trúc mạng hướng kết nối như MPLS, ATM hoặc GMPLS như đã chỉ ra trong nhiều tài liệu. Lấy ví dụ cho mạng MPLS, mặc dù có thể sử dụng các mạng khác, chiến lược hồi phục lớp kết nối đó là chiến lược hồi phục đường dẫn MPLS. Dạng đơn giản nhất của hồi phục đường dẫn là chiến lược *1+1 chuyển mạch đường dẫn tự động APS* (Automatic Path Switching). Trong chiến lược này, hai đường dẫn với cùng băng thông được thiết lập giữa điểm nguồn và điểm đích, tuy nhiên trong quá trình hoạt động, chỉ một đường dịch vụ là chuyển tải lưu lượng và đường còn lại là đường dự phòng (mặc dù đường dự phòng vẫn chuyển tải bản sao của lưu lượng của đường dịch vụ). Đường dự phòng hoàn toàn cách ly khỏi đường dịch vụ, khi có một tuyến nào đó trên đường dịch vụ có sự cố, đường dự phòng sẽ được sử dụng thay thế. Một chiến lược hồi phục khác là chiến lược *1:1 APS* cũng sử dụng hai đường dẫn cách ly nhau hoàn toàn nhưng khác với chiến lược *1+1 APS* đó là chỉ đường dịch vụ chuyển tải lưu lượng, đường dự phòng không chuyển tải. Chiến lược hồi phục *1:1 APS* mất nhiều thời gian để hồi phục hơn chiến lược hồi phục *1+1 APS*. Tuy nhiên, chiến lược hồi phục này lại có ưu điểm là đường dự phòng không chuyển tải lưu lượng do đó trong điều kiện mạng hoạt động bình thường, băng thông không được tận dụng, do vậy làm giảm yêu cầu về băng thông.

Một chiến lược hồi phục khác là hồi phục theo tuyến kết nối. Không như các chiến lược hồi phục *1+1 APS* và *1:1 APS*, hồi phục trên cơ sở đường dẫn ([2, 4, 5, 8, 12]), các chiến lược này dựa trên cơ sở từng tuyến kết nối như trong [10, 12]. Xem xét ví dụ dưới đây.

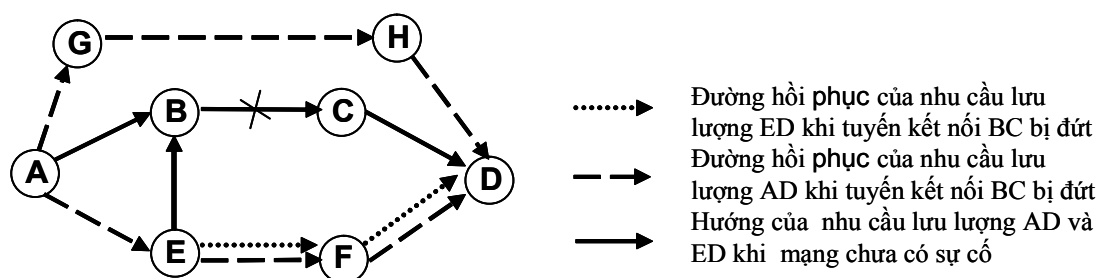


Hình 1. Hồi phục theo tuyến và đường

Trong trường hợp mạng bình thường, lưu lượng giữa A và D được chuyển theo đường

ABCD. Theo chiến lược hồi phục  $1+1$  APS hoặc  $1:1$  APS, đường AEFD được sử dụng làm đường hồi phục. Khi có sự cố, giả sử là tuyến AB bị đứt, các chiến lược hồi phục trên sẽ sử dụng đường hồi phục AEFD để chuyển tải lưu lượng giữa A và D. Một cách khác, nếu theo chiến lược hồi phục tuyến, lưu lượng chuyển vòng qua tuyến bị đứt, cụ thể trong trường hợp này đường AEBCD sẽ được sử dụng. Chiến lược này cũng có thể sử dụng cho các trường hợp nút bị sự cố, khi đó có thể coi không có lưu lượng chuyển qua các tuyến có kết nối tới nút đó. Các kết quả nghiên cứu từ [10, 12] cho thấy chiến lược hồi phục theo đường có hiệu suất sử dụng băng thông cao hơn.

Có hai phương pháp hồi phục đường: phương pháp hồi phục đường theo phương thức phân tách phần lưu lượng trên đường bị ảnh hưởng sang một số đường dự phòng khác nhau và phương thức chuyển tải toàn bộ phần lưu lượng bị ảnh hưởng trên một đường dự phòng khác. Hình 2 dưới đây minh họa cách thức hồi phục này.



Hình 2. Hồi phục đường với nhiều đường hồi phục khác nhau

Xét lại trường hợp khi nhu cầu lưu lượng AD đi theo đường ABCD. Khi tuyến BC gặp sự cố, nhu cầu lưu lượng AD được phân tách làm hai phần: phần thứ nhất được định tuyến lại đi theo đường AEFD, phần thứ hai được đi trên đường AGHD. Điều này là hoàn toàn có thể bởi trong nhiều trường hợp, nhu cầu lưu lượng AD lớn hơn bằng thông còn lại của một trong các tuyến AE, EF, FD trên đường AEFD. Vì vậy chỉ một phần lưu lượng được phân bổ trên đường này. Phần còn lại phải đi trên đường khác (AGHD), mặc dù chi phí cao hơn. Với phương thức hồi phục theo nhiều đường, trong thiết kế, chi phí tổng cộng (bao gồm cả chi phí dự phòng cho hồi phục) giảm, khả năng tận dụng băng thông cao hơn.

## 2. VẤN ĐỀ VÀ PHƯƠNG PHÁP THIẾT KẾ TOPOLOGY MẠNG CÓ KHẢ NĂNG HỒI PHỤC

Mô hình bài toán thiết kế topology mạng hồi phục sẽ được thiết lập trên cơ sở mở rộng mô hình được trình bày trong [1] để tính đến chi phí băng thông hồi phục.

Một mạng được biểu thị qua một graph vô hướng  $G(V, E)$  với  $V$  biểu thị tập các nút mạng với  $n$  phần tử và  $E$  biểu thị tập các đường kết nối giữa các nút. Mỗi một nhu cầu lưu lượng (commodity) được định nghĩa là nhóm lưu lượng cùng chung cặp nguồn đích. Gọi  $D$  là tập các nhu cầu lưu lượng (quy đổi theo đơn vị bps), được tạo nên bởi các phần tử nằm trên đường chéo của một ma trận vuông nhu cầu lưu lượng kích thước  $n$ . Mỗi một nhu cầu lưu lượng có thể được truyền tải trên một tập đường dẫn khác nhau.

Để thuận tiện, các ký hiệu sau được sử dụng:

$l_{ij}$  biểu thị đường kết nối giữa hai nút  $i$  và nút  $j$  với nhau;

$c_{ij}$  là dung lượng đường kết nối  $l_{ij}$ ;

$c_{ij}^s$  biểu thị bằng thông phân bố trên để truyền tải các nhu cầu  $l_{ij}$  khi mạng chưa có sự cố;

$c_{ij}^B$  biểu thị bằng thông phân bố trên cho việc dự phòng  $l_{ij}$  khi mạng có sự cố;

$R_{kl}^0$  biểu thị tập các tuyến  $r$  truyền tải nhu cầu  $d_{kl}$  của hai nút  $k$  và  $l$ ;

$R_{kl}^1$  biểu thị tập các tuyến hồi phục  $b$  dùng để hồi phục nhu cầu  $d_{kl}$ ;

$\lambda_{ij}$  là chi phí cho một đơn vị dung lượng đường  $l_{ij}$ ;

$f_0^r$  là phần lưu lượng của  $d_{kl}$  trên tuyến  $r$  khi mạng chưa bị sự cố;

$f_1^b$  là phần lưu lượng của  $d_{kl}$  cần hồi phục trên tuyến hồi phục  $b$  khi mạng bị sự cố;

$\delta_{ij}^r$  bằng 1 nếu tuyến  $r$  sử dụng đường kết nối  $l_{ij}$  được sử dụng để chuyển tải phần lưu lượng của nhu cầu  $d_{kl}$ , ngược lại bằng 0;

$h_{ij}^r \geq 1$  là hệ số liên quan đến vấn đề trễ cho phần lưu lượng  $f_0^r$  hoặc  $f_1^b$  đi qua đường kết nối  $l_{ij}$ , hiện có giá trị bằng 1;

$0 \leq \mu_{kl} \leq 1$  là hệ số hồi phục phần lưu lượng của  $d_{kl}$  trên tuyến  $r$  sang một tuyến khác, hiện có giá trị bằng 1;

$\sigma_{kl}^r$  có giá trị 1 khi trên tuyến  $r$ , chuyển tải phần lưu lượng của  $d_{kl}$ , có chứa đường kết nối bị sự cố. Ngược lại sẽ có giá trị 0;

$R_{kl}^0$  là tập các tuyến chuyển tải nhu cầu  $d_{kl}$  trên mạng khi chưa có sự cố;

$R_{kl}^1$  là tập các tuyến chuyển tải nhu cầu  $d_{kl}$  trên mạng khi có sự cố;

$S$  là tập nhu cầu lưu lượng phải hồi phục khi có sự cố mạng;

$F$  là tập sự cố mạng có thể xảy;

$E_f$  là tập các đường kết nối có thể ảnh hưởng bởi tập sự cố  $F$ ;

$c_{ij}^{Bf}$  là bằng thông dự phòng cần phân bố cho đường  $l_{ij}$  để hồi phục mạng cho sự cố  $f$  đang xét;

$G_{ij}^s$  là chi phí truyền tải các nhu cầu trên mạng trong điều kiện bình thường;

$G_{ij}^B$  là chi phí cho việc hồi phục các nhu cầu khi có sự cố.

Từ đây có thể lập công thức cho bài toán thiết kế topology mạng có khả năng hồi phục như sau:

$$\text{Tối thiểu hóa} \quad \sum_{i=1}^{n-1} \sum_{i>1}^n \lambda_{ij} (c_{ij}^s + c_{ij}^B) \quad (1)$$

sao cho

$$d_{kl} = \sum_{r \in R_{kl}^0} f_0^r, \quad \forall k, l \in V \quad (2)$$

$$c_{ij}^s = \sum_{d_{kl} \in D} \sum_{r \in R_{kl}^0} \delta_{ij}^r h_{ij}^r f_0^r, \quad \forall i, j, k, l \in V \quad (3)$$

$$c_{ij}^B = \sum_{d_{kl} \in D} \sum_{b \in R_{kl}^1} \delta_{ij}^b \mu_{kl} h_{ij}^b f_1^b, \quad \forall i, j, k, l \in V \quad (4)$$

$$\sum_{b \in R_{kl}^1} f_1^b = \sum_{r \in R_{kl}^0} \sigma_{kl}^r f_0^r, \quad \forall i, j, k, l \in V. \quad (5)$$

Ràng buộc trong (2) nhằm đảm bảo mọi nhu cầu lưu lượng đều được chuyển tải trên mạng. Biểu thức (3) đảm bảo băng thông phân bố trên các đường kết nối là bằng hoặc lớn hơn lưu lượng trên đó. Biểu thức (4) và (5) đảm bảo băng thông phần bổ cho phần lưu lượng cần hồi phục khi mạng có sự cố. Biểu thức (6) đảm bảo phần lưu lượng trên các tuyến có sự cố được hồi phục trên các tuyến khác.

Trong quá trình thiết kế mạng, các thuật toán hồi phục trên lớp kết nối có thể chia làm hai loại, một loại sử dụng phương pháp ILP (Integer Linear Programming) như trong một số tài liệu và loại còn lại sử dụng phương pháp thực nghiệm (heuristic) ([4, 5, 6, 10, 12]). Với phương pháp ILP, các kết quả đạt được rất tốt, tuy nhiên mất rất nhiều thời gian, cho nên thường phù hợp với những mạng nhỏ, có số lượng nút ít. Những thuật toán thực nghiệm mất ít thời gian hơn, dù vậy các kết quả đạt được cũng gần với các kết quả từ phương pháp ILP. Trong những mạng có số lượng nút nhiều, lên đến hàng vài chục nút, thời gian cần thiết để giải quyết bài toán là rất lớn, đặc biệt trong những trường hợp số đường kết nối lớn như mạng lưới đầy đủ (full mesh) cần sử dụng những thuật toán thực nghiệm. Các phương pháp thực nghiệm như ([4, 5, 6, 10, 12]) giải quyết vấn đề hồi phục mạng dựa theo hai cách tiếp cận là định tuyến lại dựa trên đường ngắn nhất ([5, 6, 10]) hoặc trên nguyên tắc luồng cực đại ([4]). Tuy nhiên, đối với các phương pháp này đều thường đưa ra giả thiết khi xây dựng thuật toán. Các thuật toán trong [5, 6, 10, 12] định trước một tập các tuyến hồi phục và sử dụng trong quá trình hồi phục mạng với giả thiết rằng số lượng tuyến hồi phục là nhỏ. Mặc dù trong [5] chỉ ra rằng việc sử dụng tập tuyến hồi phục này ảnh hưởng đến kết quả và mở rộng hơn tập tuyến hồi phục, hoặc trong [4] cho rằng các tập cắt (cut sets) là nhỏ thì việc sử dụng tập tuyến hồi phục (hay các tập cắt) được định trước sẽ làm giảm khả năng tìm kiếm giải pháp. Những giả thiết như vậy có thể không phù hợp cho các mạng lớn mà số lượng các tuyến hồi phục hoặc số lượng lát cắt cho các cặp nút là rất lớn.

Vì vậy, phương pháp đề xuất ở đây là không sử dụng các tập tuyến hồi phục, hay các tập cắt định trước để giải quyết bài toán. Với mỗi topology mạng đưa ra, các tuyến hồi phục được tính toán trên cơ sở đường ngắn nhất. Để tối ưu việc thiết lập tuyến hồi phục, một trong những vấn đề quan trọng cần giải quyết đó là làm thế nào để tăng hiệu suất sử dụng các kênh hồi phục, tức là tăng tối đa việc sử dụng chung băng thông hồi phục trong các trường hợp mạng gặp sự cố. Phương án đề xuất ở đây là, lặp lại (một số lần nhất định) quá trình thiết lập và phân bố băng thông hồi phục, với mỗi lần lặp là quá trình lựa chọn ngẫu nhiên các sự cố trên mạng. Quá trình lặp và lấy ngẫu nhiên các sự cố sẽ cho phép tìm ra các giải pháp có hiệu suất sử dụng băng thông hồi phục hiệu quả hơn.

Phương pháp thiết kế được đề xuất như sau: Sử dụng thuật toán di truyền để tìm một topology mạng đảm bảo truyền tải tập  $D$  các nhu cầu, sau đó thực hiện quá trình định tuyến lại các nhu cầu lưu lượng cần hồi phục theo phương pháp hồi phục đường trên nhiều đường hồi phục khác nhau. Quá trình xác định các đường hồi phục sẽ được lặp với một số lần xác định. Chi phí tổng cộng cho cả hai quá trình này là cơ sở để tính mức độ phù hợp (fitness) trong thuật toán gen phục vụ cho quá trình lặp trong các thế hệ tiếp theo.

Thuật toán gen [1], với phương pháp mã hóa theo thứ tự đưa nhu cầu lưu lượng lên mạng

cho những mạng mà dung lượng đường kết nối là hữu hạn, đạt hiệu quả hơn các phương pháp mã hóa khác trong các tài liệu [7, 9, 11]. Mỗi nhu cầu sẽ được gán một số xác định. Chuỗi các số này sẽ tạo nên một trật tự đưa các nhu cầu lưu lượng trên mạng. Ví dụ, giả sử một mạng gồm 4 nút với các nhu cầu lưu lượng là  $D_{12}, D_{13}, D_{14}, D_{23}, D_{24}, D_{34}$  và được gán số lần lượt là 1, 2, 3, 4, 5, 6. Như vậy nếu theo một chuỗi trật tự ví dụ 234516 sẽ tạo ra một topology mạng khi lần lượt sắp xếp các nhu cầu lưu lượng  $D_{13}, D_{14}, D_{23}, D_{24}, D_{12}, D_{34}$  lên mạng. Các nhu cầu lưu lượng được sắp xếp trên mạng trên cơ sở thuật toán đường ngắn nhất. Trong những mạng có dung lượng đường kết nối vô hạn hoặc rất lớn so với nhu cầu lưu lượng thì trật tự đưa các nhu cầu lưu lượng lên mạng không làm ảnh hưởng đến chi phí tổng cộng. Tuy nhiên, ở những mạng có dung lượng hữu hạn, trật tự đưa nhu cầu lưu lượng tác động rất nhiều đến chi phí tổng cộng.

Khởi tạo thể hệ khởi đầu được thực hiện bằng cách sinh ra các chuỗi với trật tự ngẫu nhiên của các số gán cho các nhu cầu lưu lượng. Ví dụ như các chuỗi 234165, 623415, 342165... sẽ tạo nên các topology mạng khác nhau.

Toán tử lai ghép, dựa trên cơ sở sử dụng chuỗi trật tự như trên, giữa hai cá thể có kích thước chuỗi  $N$ , lai ghép tại vị trí  $L$  ( $L < N$ ) bất kỳ, ví dụ  $L = 3$ , sẽ được thực hiện như sau:

Cá thể 1 thế hệ  $n$ : **136425**

Cá thể 2 thế hệ  $n$ : **234156**

Cá thể thế hệ  $n + 1$ : **136245**

Như vậy nguyên tắc lai ghép là từ hai cá thể thế hệ đầu kích thước  $N$ , lai ghép tại vị trí  $L$  ( $L < N$ ), thế hệ sau được tạo ra với kích thước  $N$  và chuỗi được thiết lập với  $L$  phần tử đầu là  $L$  phần tử của chuỗi thứ nhất và  $N - L$  phần tử tiếp theo lấy từ chuỗi thứ hai, bằng cách duyệt theo trật tự của chuỗi thứ hai, phần tử được lấy là phần tử không xuất hiện trong  $L$  phần tử đầu.

Toán tử đột biến, thực hiện trên một cá thể nào đó, tại hai vị trí  $K$  và  $L$  ( $K < N$  và  $L < N$ ), được thực hiện bằng cách hoán đổi hai vị trí  $K$  và  $L$  cho nhau, ví dụ  $K = 2$  và  $L = 5$ .

Cá thể  $i$  thế hệ  $n$ : **234156**

Cá thể  $i$  sau khi thực hiện đột biến: **254136**

Ngoài ra có thể thực hiện toán tử đột biến thông qua việc quay vòng phải hoặc vòng trái tại vị trí  $L$  của chuỗi  $N$  như sau:

Quay vòng phải tại vị trí  $L = 5$  với cá thể  $i$ : **234156**, kết quả tạo ra cá thể: **562341**

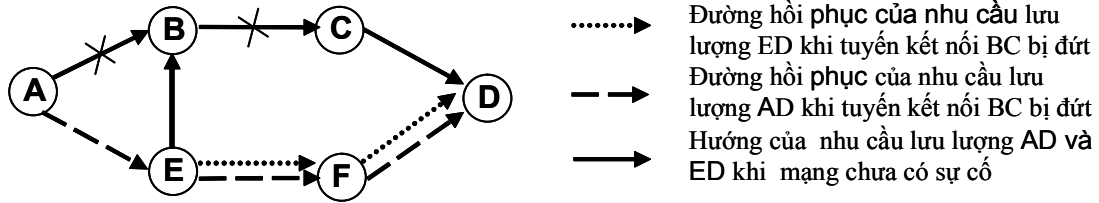
Quay vòng trái tại vị trí  $L = 3$  với cá thể  $i$ : **234156**, kết quả tạo ra cá thể: **156234**

Tại giai đoạn này, tập các đường kết nối  $l_{ij}$  cùng với bảng thông phân bố  $c_{ij}^s$  và chi phí  $G_{ij}^s$  để đảm bảo truyền tải các nhu cầu lưu lượng là được xác định.

Tiếp theo, ta tính bảng thông dự phòng  $c_{ij}^B$  và chi phí cho việc cấp băng thông dự phòng  $G_{ij}^B$  cho việc hồi phục mạng khi có sự cố. Để tính toán bảng thông dự phòng, xem Hình 3 sau.

Từ Hình 3 cho thấy, giả sử tập sự cố bao gồm tuyến AB và BC có thể có sự cố. Giả sử, khi tuyến kết nối BC bị sự cố, nhu cầu lưu lượng  $D_{AD}$  và  $D_{ED}$  sẽ được chuyển từ đường dẫn ABCD và EBCD sang đường dẫn Aefd và EFD. Việc chuyển sang các đường dẫn mới cũng được thực hiện theo thuật toán đường ngắn nhất. Trong trường hợp này, giả sử nhu cầu lưu lượng  $D_{AD}$  và  $D_{ED}$  nằm trong tập  $S$  (tập cần phải bảo vệ) thì, bảng thông dự

phòng trên tuyến kết nối EF và FD cho sự cố tuyến BC đứt lần lượt là  $C_{EF}^B$  và  $C_{FD}^B$  bằng tổng nhu cầu lưu lượng  $D_{AD}$  và  $D_{ED}$ . Bảng thông dự phòng trên tuyến kết nối AE là  $C_{AE}^B$  sẽ bằng nhu cầu lưu lượng  $D_{AD}$ .



Hình 3. Hướng nhu cầu lưu lượng  $D_{ED}$  và  $D_{AD}$  thay đổi khi mạng có sự cố

Tiếp tục xét tiếp trường hợp sự cố là tuyến kết nối AB đứt, khi đó chỉ có nhu cầu lưu lượng  $D_{AD}$  được chuyển sang đường dẫn là AEFD. Trong trường hợp này, bảng thông dự phòng sẽ được phân bổ trên các tuyến AE, EF, FD lần lượt là  $C_{AE}^B$ ,  $C_{EF}^B$  và  $C_{FD}^B$  bằng với nhu cầu lưu lượng  $D_{AD}$ . Tuy nhiên, do bảng thông dự phòng đã được phân bổ từ trước cho trường hợp sự cố trước (BC đứt) lớn hơn bảng thông dự phòng cần cho trường hợp này, nên bảng thông dự phòng không phải phân bổ thêm. Và như vậy, trường hợp sau không có chi phí cho việc phân bổ thêm bảng thông dự phòng. Ngược lại, trong trường hợp sự cố đòi hỏi bằng thông dự phòng lớn hơn bảng thông dự phòng đã cấp, thì bảng thông dự phòng sẽ bằng bảng thông dự phòng mới và chi phí sẽ tăng thêm một lượng bằng chi phí cho phần bảng thông tăng thêm.

Từ đó ta có:

$$c_{ij}^B = c_{ij}^{Bf} \quad \text{nếu} \quad c_{ij}^{Bf} \geq c_{ij}^B \quad \text{và} \quad c_{ij}^{Bf} \leq c_{ij} - c_{ij}^s, \quad (6)$$

$$c_{ij}^B = c_{ij}^B \quad \text{nếu} \quad c_{ij}^{Bf} \leq c_{ij}^B, \quad (7)$$

khi đó chi phí  $G_{ij}^B$  cho việc cấp bằng thông hồi phục sẽ được tính trên cơ sở cho các trường hợp (6) và (7).

Thuật toán tính toán bằng thông và chi phí cho việc hồi phục mạng được mô tả như sau:

Procedure Resilient\_Cost()

$C_{ij}^B = 0;$

for  $N = 1$  to  $N = N_{\max}$  do

for  $|F|$  sự cố trong tập  $F$  do

Lựa chọn ngẫu nhiên một sự cố  $F_i;$

Xác định tập các đường kết nối  $E_f$  bị ảnh hưởng bởi sự cố;

Xác định các nhu cầu lưu lượng  $S_{ij}$  cần phục hồi sử dụng các đường kết nối  $E_f;$

Cập nhật lại  $G(V, E)$  trên cơ sở loại bỏ các đường kết nối  $E_f;$

Định tuyến lại các nhu cầu lưu lượng  $S_{ij}$  trên  $G(V, E)$  theo đường ngắn nhất;

Xác định bằng thông hồi phục;

Xác định chi phí cho bằng thông hồi phục;

Trả lại các đường kết nối  $E_f$  trong  $G(V, E);$

end

Xác định chi phí bằng thông hồi phục tối thiểu;

end

Quá trình tìm kiếm các đường dẫn mới sẽ loại bỏ các đường kết nối mà không còn bằng thông cho việc hồi phục.

Các kết quả tính toán chi phí bằng thông  $G_{ij}^S$ ,  $G_{ij}^B$  được sử dụng để tính mức độ phù hợp của các cá thể. Như vậy, toàn bộ thuật toán thiết kế được mô tả như sau:

```

Procedure Algorithm_Resilent_Network()
  Khởi tạo thế hệ đầu  $P$  với số lượng  $N$ ;
  for gen =1 to maxGen do
    for  $j = 1$  to  $N/2$  do
      Lựa chọn ngẫu nhiên cá thể bất kỳ trong  $P$ ;
      Chọn hai cá thể có mức độ phù hợp cao nhất trong ba cá thể;
      Lai ghép hai cá thể;
    end
    Tạo các cá thể đột biến từ  $P$  với một số lượng xác định;
    Khởi tạo một số cá thể mới với số lượng xác định;
    Tính toán mức độ phù hợp của các cá thể mới; // tính cả chi phí hồi phục
    Lựa chọn  $N$  cá thể mới có mức độ phù hợp cao nhất trong toàn bộ tập cá thể;
    Đưa ra các kết quả;
  End
  
```

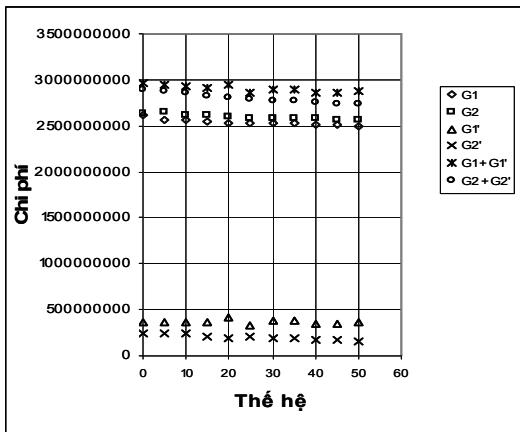
Một cách tiếp cận khác dựa trên một cách tiếp cận đã được đề xuất đó là sử dụng thuật toán di truyền tìm một topology mạng với chi phí thấp nhất (chưa có khả năng phục hồi) trong việc đảm bảo truyền tải tập  $D$  các nhu cầu lưu lượng. Mức độ phù hợp (fitness) trong thuật toán di truyền chỉ được tính cho giai đoạn đầu. Ở giai đoạn sau sẽ định tuyến lại các nhu cầu lưu lượng cần phục hồi và chi phí cuối cùng là tổng của hai giai đoạn. Việc so sánh hai cách tiếp cận này sẽ được thể hiện qua một số kết quả tính toán dưới đây.

### 3. MỘT SỐ KẾT QUẢ TÍNH TOÁN THỰC NGHIỆM

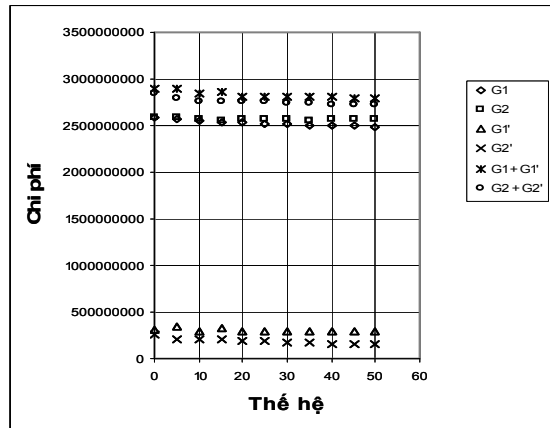
Một số kết quả thu được trong quá trình thực hiện hai phương pháp tiếp cận trên được đưa ra ở đây. Tất cả các kết quả được thực hiện trên cơ sở lặp lại một số lần. Để so sánh tính hiệu quả giữa việc sử dụng chi phí thiết lập mạng chưa có dự phòng và chi phí tổng cộng (gồm cả chi phí dự phòng) để tính mức độ phù hợp cho thuật toán gen, một mạng 20 nút được sử dụng. Các ma trận nhu cầu lưu lượng, dung lượng các đường kết nối, chi phí dung lượng đường kết nối,...được sinh ngẫu nhiên bởi chương trình. Các nhu cầu lưu lượng được quy đổi có giá trị từ 0.5Mbps đến 2.5Mbps; dung lượng đường kết nối có giá trị từ 3Mbps đến 5Mbps; chi phí cho một đơn vị dung lượng kênh kết nối từ 5000 đến 35000 cho 1Kbps; kích thước dân số được đặt với giá trị 25, số thế hệ là 50, tỷ lệ lai ghép là 45%, tỷ lệ đột biến là 30%, tỷ lệ tạo cá thể mới là 55%.

Trên các Hình 3a và Hình 3b trình bày các kết quả tính toán. Chi phí  $G1$  và  $G1'$  ứng với trường hợp thiết kế topology mạng với mức độ phù hợp trong thuật toán gen được dựa trên cơ sở chi phí  $G1$  mà không tính đến  $G1'$ . Ngược lại, chi phí  $G2$  và  $G2'$  là cho trường hợp mức độ phù hợp dựa trên chi phí tổng cộng  $G2 + G2'$ . Các chi phí  $G1$  và  $G2$  là chi phí thiết lập mạng chưa có dự phòng. Các chi phí  $G1'$  và  $G2'$  là các chi phí cho việc hồi phục mạng.





Hình 3a.

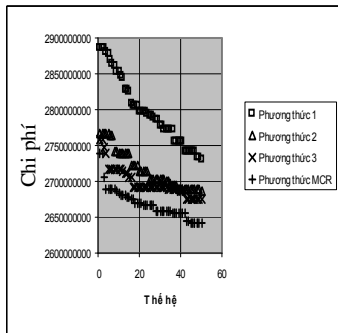


Hình 3b.

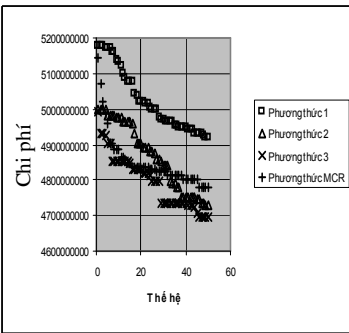
Hồi phục đường trên một đường hồi phục

Hồi phục đường trên nhiều đường hồi phục

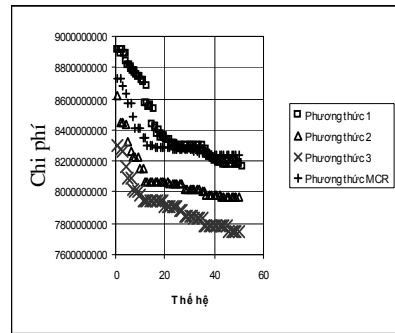
Các kết quả tính toán cho hai trường hợp hồi phục đường cho thấy rằng việc sử dụng chi phí tổng cộng làm cơ sở tính mức độ phù hợp của thuật toán gen sẽ cho kết quả tốt hơn. Trên cơ sở này, các tính toán tiếp theo được thực hiện nhằm so sánh ba phương thức: hồi phục đường trên một đường hồi phục (non-bifurcation) - phương thức 1, hồi phục đường trên nhiều đường hồi phục (bifurcation) - phương thức 2 và phục hồi lặp là phương thức được đề xuất ở đây dựa trên cơ sở lặp lại một số lần quá trình phục hồi đường trên nhiều đường hồi phục - phương thức 3.



Hình 4a. Trường hợp nhu cầu lưu lượng cần hồi phục chiếm 12% lưu lượng toàn bộ mạng



Hình 4b. Trường hợp nhu cầu lưu lượng cần hồi phục chiếm 40% lưu lượng toàn bộ mạng



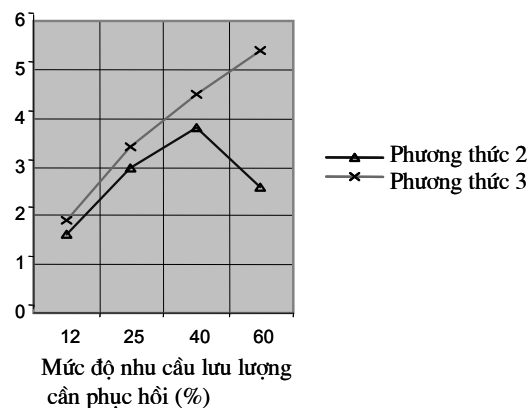
Hình 4c. Trường hợp nhu cầu lưu lượng cần dự phòng chiếm 60% lưu lượng toàn bộ mạng

Trong [12] đưa ra thuật toán hồi phục MCR (Minimum Cost Restoration) với năm bước thực hiện. Hai bước đầu về nội dung tương tự như phương thức 1 trình bày ở trên. Ba bước sau thực chất là quá trình tối ưu tiếp dựa trên quá trình tìm kiếm cục bộ từ kết quả của hai bước đầu. Phương thức hồi phục theo thuật toán MCR được xây dựng nhằm làm cơ sở so sánh với thuật toán hồi phục lặp được đề xuất tại đây. Trên Hình 4a, 4b và 4c thể hiện các kết quả được thực hiện trong điều kiện khác nhau về yêu cầu mức độ phục hồi tại các mức độ 12%, 40% và 60%.

Kết quả cho thấy việc thực hiện phục hồi theo phương thức lặp cho kết quả tốt nhất.

Những trường hợp mức độ yêu cầu phục hồi của mạng là cao từ khoảng 40% trở lên, các phương thức hồi phục 2 và 3 đang xem xét cho kết quả tốt hơn đôi chút so với thuật toán MCR ở [12]. Nguyên nhân là do trong những trường hợp mà yêu cầu mức độ hồi phục cao, việc sử dụng phương pháp hồi phục nhiều đường cho kết quả tốt hơn phương pháp hồi phục không trên nhiều đường mà thuật toán MCR áp dụng. Với trường hợp mức độ nhu cầu hồi phục thấp khoảng trên dưới 10% hoặc tương đương với các trường hợp mà dung lượng các đường kết nối còn dư thừa nhiều khi so sánh tương đối với nhu cầu cần hồi phục, thì thuật toán MCR cho kết quả trội hơn đôi chút (Hình 4a). Kết quả này do khi mạng còn dư thừa bằng thông, phương pháp hồi phục trên nhiều đường mà phương thức 2 và 3 áp dụng không vượt trội so với phương thức MCR áp dụng. Mặt khác, về bản chất, thuật toán MCR cũng áp dụng một quá trình tìm kiếm tối ưu để cải tiến chất lượng của giải pháp đạt được. Tuy nhiên, cũng cần lưu ý rằng, khác với thuật toán đề xuất ở đây thực hiện theo biên đa thức thời gian (polynomial-time bound), thuật toán MCR là thuật toán thực hiện không theo biên đa thức thời gian. Do vậy thời gian chi phí cho việc thực hiện thuật toán MCR rất lớn, gấp nhiều lần thuật toán đề xuất ở đây.

Gọi chi phí cho việc thiết kế topology mạng có khả năng hồi phục cho trường hợp 1 là  $C^1$ , tương tự cho phương thức 2 và 3 là  $C^2, C^3$ . Trong Hình 5 dưới đây, trục tung được tính từ tỷ số của  $(C^1 - C^2)/C^1$  và  $(C^1 - C^3)/C^1$  tương ứng với phương thức 2 và 3 nhằm so sánh mức độ tích kiệm chi phí của các phương thức hồi phục tại các mức độ yêu cầu hồi phục khác nhau. Từ Hình 5 có thể thấy rõ việc sử dụng phương thức lập - phương thức 3 có mức độ tích kiệm chi phí hơn hẳn phương thức 2 khi mức độ yêu cầu hồi phục tăng từ khoảng 40% trở lên. Phương thức hồi phục trên nhiều đường - phương thức 2 có nhiều hiệu quả hơn so với phương thức hồi phục trên một đường - phương thức 1, tùy thuộc vào từng mạng, mức độ hiệu quả tăng dần theo mức độ tăng của nhu cầu lưu lượng cần hồi phục và đạt đến một giá trị nào đó. Sau đó, mức độ hiệu quả lại giảm khi tăng mức độ nhu cầu lưu lượng cần hồi phục. Điều này là do khi tăng quá mức độ nhu cầu lưu lượng cần phục hồi, cần thiết lập các tuyến mới, đối với các phương thức đều không tích kiệm được chi phí.

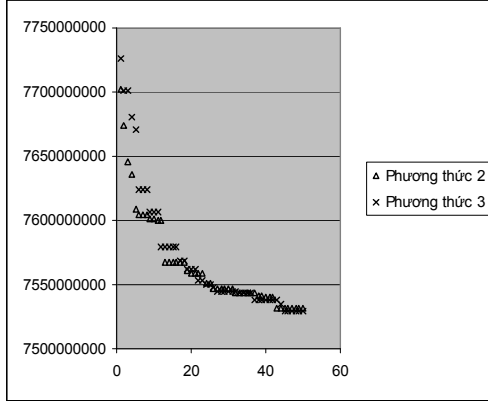


Hình 5. Chênh lệch chi phí thiết kế khi sử dụng phương thức hồi phục 1 với 2 và 3

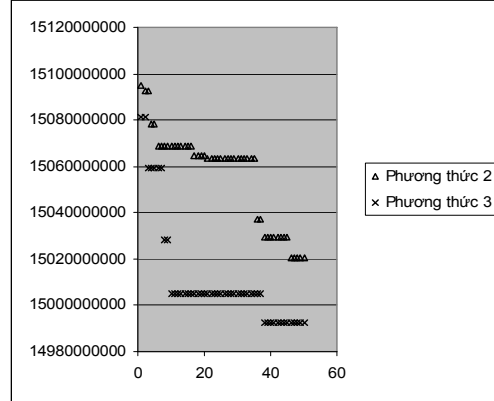
Với các kết quả tính toán này, chúng tôi thực hiện thiết kế topology mạng có khả năng hồi phục cho trường hợp mạng 35 nút và mạng 50 nút với mức độ nhu cầu lưu lượng cần hồi phục là 12% cho hai phương thức: phục hồi trên nhiều đường và phục hồi lập. Các kết

quả được thể hiện tại Hình 6a và 6b dưới đây.

Từ kết quả tính toán sơ bộ cho thấy, trong một số trường hợp thực nghiệm như trên, phương pháp hồi phục lặp cho kết quả khả quan.



Hình 6a Trường hợp mạng lưới đầy đủ 35 nút



Hình 6b Trường hợp mạng lưới đầy đủ 50 nút

#### 4. KẾT LUẬN

Qua các kết quả tính toán ở trên có thể thấy, thuật toán hồi phục lặp được đề xuất ở đây cho kết quả tính toán tốt hơn so với một số thuật toán khác đã được đề cập trước đây. Mức độ tích kiệm chi phí của thuật toán càng lớn khi mức độ yêu cầu phục hồi lưu lượng mạng càng lớn. Mặt khác, quá trình thiết kế mạng với việc kết hợp giữa quá trình thiết kế mạng trong điều kiện bình thường và quá trình tính toán phân bổ băng thông hồi phục đạt kết quả tốt hơn việc phân tách hai quá trình tính toán thiết kế riêng biệt. Quá trình tính toán cũng cho thấy, việc sử dụng các phương pháp hồi phục đường trên nhiều đường hồi phục cho phép sử dụng băng thông hiệu quả hơn, tích kiệm chi phí, khả năng thiết kế mạng linh hoạt hơn. Thuật toán thiết kế topology mạng hồi phục như trên có khả năng thiết kế cho các mạng lưới đầy đủ, dung lượng hữu hạn với số lượng nút lớn, đáp ứng nhu cầu thiết kế các mạng lớn trong thực tế.

*Giới thiệu chương trình:*

Thuật toán được thực hiện qua chương trình xây dựng trên ngôn ngữ lập trình C<sup>++</sup>. Chương trình cho phép nhập hoặc tự sinh các dữ liệu đầu vào như ma trận nhu cầu, ma trận kết nối, dung lượng đường kết nối, giá đơn vị cho các đường kết nối, tập các đường kết nối có thể bị sự cố, tập các nhu cầu cần bảo vệ. Các tùy chọn khác được hỗ trợ nhằm tăng khả năng tìm kiếm trong phần thuật toán gen của chương trình như đặt kích thước dân số, tỷ lệ lai ghép, tỷ lệ đột biến, tỷ lệ tạo cá thể mới.

#### TÀI LIỆU THAM KHẢO

- [1] L. Berry, B. Murtagh, G. McMahon, S. Sugden, and M. Randall, *Fast Network Design for Telecommunications*, 1999.

- [2] G. Li, D. Wang, C. Kalmanek, and R. Doverspike, Efficient distributed restoration path selection for shared mesh restoration, *IEEE/ACM Transactions on Networking* **11** (5) (October 2003).
- [3] D. E. Golberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.
- [4] H. Sakauchi, Y. Nishimura, and S. Hasegawa, A self-healing network with an economical spare-channel assignment, *Proceedings of IEEE GLOBE-COM*, San Diego, USA (December 1990).
- [5] R. Iraschko, M. MacGregor, and W. Grover, Optimal capacity placement for path restoration in mesh survivable networks, *Proceedings of IEEE ICC*, Dallas, USA (June 1996).
- [6] K. Murakami and H. Kim, Joint optimization of capacity and flow assignment for self-healing ATM networks, *Proceedings of IEEE ICC*, Seattle, USA (June 1995).
- [7] A. Konak, and A. Smith, A hybrid genetic algorithm approach for backbone design of communication networks, *Conf. Proc. CEC 99*, IEEE press, Piscataway, N.J., S. (1999) 1817–1823.
- [8] M. Kodialam, and T. V. Lakshman, Dynamic routing of restorable bandwidth guaranteed tunnels using aggregated network resource usage information, *IEEE/ACM Transactions on Networking* **11** (3)(June 2003).
- [9] C. C. Palmer, “An Approach to a Problem in Network Design Using Genetic Algorithms” PhD Thesis, Polytechnic University, Computer Scientific Departement, Brookly, NewYork, 1994.
- [10] M. Herzberg, S. Bye, and A. Utano, The hop-limit approach for spare-capacity assignment in survivable networks, *IEEE/ACM Transactions on Networking* **3** (6) (1995) 775–784.
- [11] G. R. Raidl, and B. A. Julstrom, Edge-sets: An effective evolutionary coding of spanning trees, *IEEE Transactions on Evolutionary Computation* **7** (3) (2003) 225–239.
- [12] Y. Xiong and L. Mason, Restoration strategies and spare capacity requirements in self-healing ATM networks, *IEEE/ACM Transactions on Networking* **7** (1) (1999) 98–110.

*Nhận bài ngày 5 - 9 - 2006*

*Nhận lại sau sửa ngày 15 - 1 - 2007*