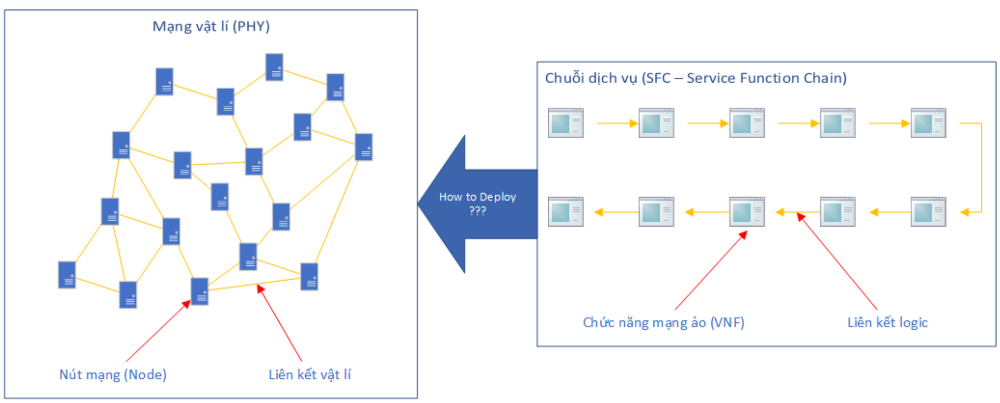
SFC-PHY Mapping Problem Explained

Nguyễn Minh Thành

Ngày 15 tháng 8 năm 2023

1 Giới thiệu

Network Slicing có nghĩa là phân tách mạng vật lí thành nhiều lát mạng nhỏ, mỗi lát mạng phục vụ một chuỗi dịch vụ khác nhau. Mạng vật lí là một hệ thống gồm nhiều các nút mạng được kết nối với nhau. Một chuỗi dịch vụ bao gồm nhiều chức năng mạng ảo được nối với nhau. Vấn đề đặt ra là làm sao để đặt các chức năng mạng ảo trong chuỗi dịch vụ trên các nút vật lí sao cho các chức năng được kết nối đúng như thiết kế, chuỗi dịch vụ hoạt động đúng yêu cầu và đảm bảo tính hiệu quả cao trong vận hành hệ thống (nói đơn giản là có thể đặt nhiều chuỗi dịch vụ với chất lượng vận hành tốt nhất có thể).



Hình 1: Bài toán Network Slicing

Như vậy, bài toán của chúng ta là một bài toán tối ưu, có đầu vào (input) đó là mạng vật lí và chuỗi dịch vụ; đầu ra (output) của bài toán đó là cách đặt chuỗi dịch vụ lên mạng vật lí. Để có thể giải bài toán này, ta cần xây dựng một mô hình toán học để tổng quát hoá bài toán. Để đơn giản nhất, ta sẽ thực hiện tổng quát hoá bài toán dựa trên mô hình toán học tuyến tính. Lúc này, ta sẽ cần xác định hai yếu tố, một là các điều kiện ràng buộc (dưới dạng bất phương trình tuyến tính), hai là hàm tổng quát (dưới dạng hàm đa thức) và yêu cầu về hàm đó (cực đại, cực tiểu, ...).

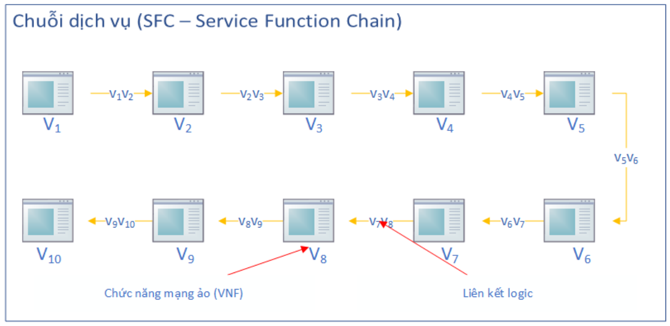
2 Mô hình toán học

Để có thể xây dựng mô hình toán học của bài toán, ta cần toán học và tổng quát hoá các dữ kiện đầu vào và yêu cầu đầu ra.

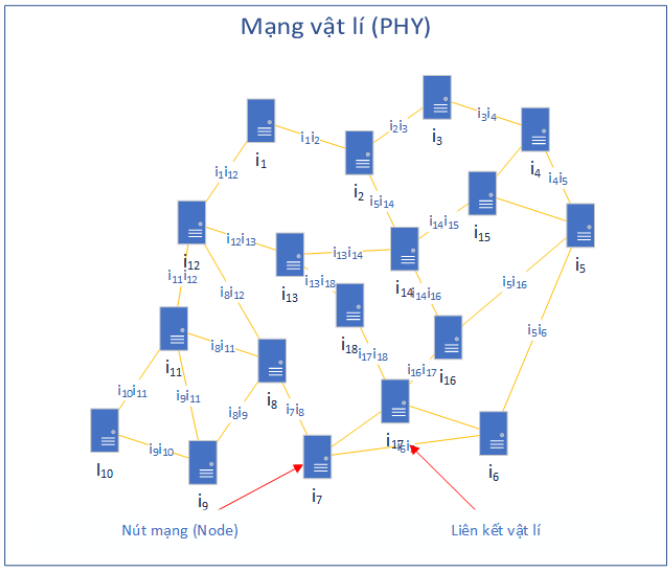
2.1 Dữ kiện đầu vào

Mỗi một chuỗi dịch vụ (Service Functions Chain - SFC) có thể được mô tả dưới dạng một đồ thị $\mathcal{G}_s(\mathcal{N}_s, \mathcal{E}_s)$ như trong Hình 2. Trong đó, $\mathcal{N}_s = \{v_1, v_2, ...\}$ là tập hợp gồm các chức năng được ảo hoá (Virtual Network Function - VNF) và $\mathcal{E}_s = \{v_1v_2, v_2v_3, ...\}$ là tập hợp gồm liên kết giữa các cặp VNF. Mỗi một VNF sẽ có một yêu cầu nhất định về tài nguyên (CPU, MEM, STORAGE, ...), ta sẽ kí hiệu chung là r_v, mỗi một liên kết giữa một cặp VNF cũng sẽ có những yêu cầu nhất định khi truyền dẫn (Bandwidth, Latency, Jitter, ...), ta sẽ kí hiệu chung là r_{vw}.

Một hệ thống mạng vật lí (Physical Network - PHY) có thể được mô tả dưới dạng một đồ thị $\mathcal{G}(\mathcal{N}, \mathcal{E})$ như trong Hình 3. Trong đó, $\mathcal{N} = \{i_1, i_2, ...\}$ là tập hợp gồm các nút mạng (máy chủ, trạm, điểm kết nối, ...) vật lí và



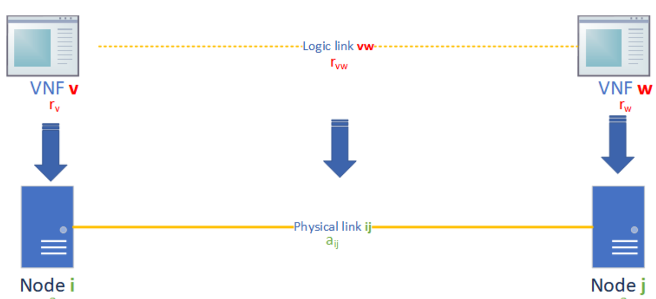
Hình 2: Cấu trúc SFC



Hình 3: Cấu trúc mạng vật lí

$\mathcal{E} = \{i_1i_2, i_2i_3, ...\}$ là tập hợp gồm liên kết giữa các cặp nút mạng. Mỗi một nút vật lí sẽ có lượng tài nguyên nhất định (CPU, MEM, STORAGE, ...), kí hiệu chung là a_i. Mỗi một liên kết giữa một cặp nút mạng sẽ có một khả năng tối đa nhất định (Bandwidth, Latency, Jitter,...), ta kí hiệu chung là a_{ij}.

Giả sử ta đã đặt một VNF v lên một nút mạng i và VNF w lên nút mạng j. Trong đó v,$w$là hai VNF liền kề trong SFC, i,j là hai nút mạng kề nhau trong mạng vật lí; liên kết logic vw của hai VNF được đặt trong liên kết vật lí ij của hai nút mạng (Hình 4).



Hình 4: Ánh xạ SFC lên mạng vật lí

Ta đặt ϕ_i^v là một biến biểu thị việc VNF v có được đặt tại nút i hay không. Khi đó, $\phi_i^v = 1$ khi v được đặt tại nút i, ngược lại, $\phi_i^v = 0$.

Ta đặt ϕ_{ij}^{vw} là một biến biểu thị việc liên kết logic vw có được đặt tại liên kết vật lí ij hay không. Khi đó,

$\phi_{ij}^{vw} = 1$ khi vw được đặt tại ij, ngược lại, $\phi_{ij}^{vw} = 0$.

Như vậy ta đã toán học hoá toàn bộ các dữ kiện đầu vào và đầu ra của bài toán. Cụ thể:

Đầu vào (input)

- Đồ thị chuỗi dịch vụ (SFC): $\mathcal{G}_s(\mathcal{N}_s, \mathcal{E}_s)$
- Đồ thị mạng vật lí (PHY): $\mathcal{G}\left(\mathcal{N}, \mathcal{E}\right)$

Đầu ra (output)

- Vị trí đặt các VNF trên nút mạng: $\phi_i^v = \begin{cases} 1 & \text{khi } v \text{ được đặt tại } i \\ 0 & \text{khi } v \text{ không được đặt tại } i \end{cases}$
- Vị trí đặt các liên kết logic trên các liên kết vật lí: $\phi_{ij}^{vw} = \begin{cases} 1 & \text{khi } vw \text{ được đặt tại } ij \\ 0 & \text{khi } vw \text{ không được đặt tại } ij \end{cases}$

Bảng 1: Các tham số của mô hình

Tham số	Ý nghĩa
$\mathcal{G}_s(\mathcal{N}_s, \mathcal{E}_s)$	Đồ thị chuỗi dịch vụ (SFC)
$\mathcal{N}_s$	Tập các VNF
$\mathcal{E}_s$	Tập các kết nối logic giữa các VNF
$\mathcal{G}(\mathcal{N}, \mathcal{E})$	Đồ thị mạng vật lí (PHY)
$\mathcal{N}$	Tập các nút vật lí
$\mathcal{E}$	Tập các liên kết vật lí

Bảng 2: Các biến của mô hình

Biến	Ý nghĩa
ϕ_i^v	Trạng thái đặt VNF v tại nút vật lí i
ϕ_{ij}^{vw}	Trạng thái đặt liên kết logic vw tại liên kết vật lí ij

2.2 Các điều kiện ràng buộc

Trong bài toán tối ưu, trước khi thực hiện bất kì bước nào, ta cần phải xác định các điều kiện ràng buộc giữa các dữ kiện và biến.

Đầu tiên, muốn một VNF có thể được đặt vào một nút vật lí, thì lượng tài nguyên sẵn có của nút đó phải tối thiểu bằng, hoặc có nhiều hơn lượng tài nguyên mà VNF yêu cầu. Một nút có thể được đặt nhiều VNF, do đó tổng lượng tài nguyên yêu cầu của các VNF phải ít hơn hoặc tối đa là bằng lượng tài nguyên mà nút đó có sẵn.

$$\sum_{v \in \mathcal{N}_s} \phi_i^v . r_v \leq a_i \qquad \forall i \in \mathcal{N} \tag{1}$$

Thứ hai, tương tự, muốn một liên kết logic có thể được đặt vào một liên kết vật lí thì khả năng của liên kết vật lí đó phải tối thiểu bằng hoặc cao hơn mức yêu cầu của liên kết logic. Trong một liên kết vật lí có thể có nhiều liên kết logic, do đó tổng yêu cầu yêu cầu của các liên kết logic phải thấp hơn hoặc tối đa là bằng khả năng tối đa của liên kết vật lí.

$$\sum_{vw \in \mathcal{E}_s} \phi_{ij}^{vw} . r_{vw} \leq a_{ij} \qquad \forall ij \in \mathcal{E} \tag{2}$$

Thứ ba, mặc dù trên lí thuyết, ta có thể đặt nhiều VNF của cùng một SFC trên một nút mạng duy nhất, tuy nhiên, trong thực tế thì ta không làm như vậy. Bởi lẽ nếu đặt nhiều VNF của cùng một SFC trên một nút mạng thì trong trường hợp khi mà nút mạng đó ngoại tuyến, ngắt kết nối khỏi mạng (do lỗi đường truyền, lỗi phần cứng, lỗi phần mềm, thiên tai, ...) thì việc khôi phục lại chuỗi dịch vụ tốn nhiều thời gian cũng như công sức. Do đó, mỗi một nút vật lí chỉ chứa tối đa 1 VNF của một chuỗi dịch vụ. Lúc này, với mỗi nút vật lí, tổng các biến chỉ vị trí đặt các VNF trên nút đó phải nhỏ hơn 1.

$$\sum_{v \in \mathcal{N}_s} \phi_i^v \leq 1 \qquad \forall i \in \mathcal{N} \tag{3}$$

Thứ tư, muốn một chuỗi dịch vụ có thể hoạt động thì toàn bộ các VNF của chuỗi hoạt động đó phải được triển khai lên mạng vật lí. Tức là với mỗi một VNF, tổng các biến chỉ vị trí đặt VNF đó trên các nút vật lí đó phải bằng 1.

$$\sum_{i \in \mathcal{N}} \phi_i^v = 1 \qquad \forall v \in \mathcal{N}_s \tag{4}$$

Cuối cùng, để chuỗi dịch vụ hoạt động đúng theo thiết kế và yêu cầu, thứ tự đặt các VNF trên mạng vật lí phải tuân đúng theo logic đã được đặt ra bởi chuỗi dịch vụ.

$$\sum_{j \in \mathcal{N}} \phi_{ij}^{vw} - \sum_{j \in \mathcal{N}} \phi_{ji}^{vw} = \phi_i^v - \phi_i^w \qquad \forall i \in \mathcal{N}, \forall vw \in \mathcal{E}_s \tag{5}$$

Như vậy, ta đã xác định được toàn bộ các điều kiện ràng buộc của bài toán là các Hệ thức 1, 2, 3, 4, và 5.

2.3 Hàm mục tiêu

Từ các dữ kiện đầu vào, ta có thể đặt ra một số bài toán kèm các hàm toán học của chúng.

- Tổng số nút và liên kết vật lí là nhỏ nhất:

$$\min\left(\phi_{\mathcal{G}}^{\mathcal{G}_s}\right) = \min\left(\sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{N}_s} \phi_i^v + \sum_{ij \in \mathcal{E}} \sum_{vw \in \mathcal{E}_s} \phi_{ij}^{vw}\right) \tag{6}$$

- Tổng lượng tài nguyên cần sử dụng là ít nhất:

$$\min\left(r_{\mathcal{G}}^{\mathcal{G}_s}\right) = \min\left(\sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{N}_s} \phi_i^v r_v + \sum_{ij \in \mathcal{E}} \sum_{vw \in \mathcal{E}_s} \phi_{ij}^{vw} r_{vw}\right) \tag{7}$$

- Tổng chi phí cần thiết sử dụng là nhỏ nhất (Giả sử giá của mỗi đơn vị tài nguyên sử dụng trên nút vật lí i là c_i, giá của mỗi đơn vị tài nguyên sử dụng trên liên kết vật lí ij là c_{ij}):

$$\min\left(c_{\mathcal{G}}^{\mathcal{G}_s}\right) = \min\left(\sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{N}_s} \phi_i^v c_i r_v + \sum_{ij \in \mathcal{E}} \sum_{vw \in \mathcal{E}_s} \phi_{ij}^{vw} c_{ij} r_{vw}\right) \tag{8}$$

- Độ trễ toàn phần của SFC là nhỏ nhất (Giả sử mỗi liên kết ij có một độ trễ không đổi là d_{ij}):

$$\min\left(d_{\mathcal{G}}^{\mathcal{G}_s}\right) = \min\left(\sum_{ij \in \mathcal{E}} \sum_{vw \in \mathcal{E}_s} \phi_{ij}^{vw} d_{ij}\right) \tag{9}$$

Ta hoàn toàn có thể áp dụng và mô hình nêu trên để giải các bài toán khác trong lí thuyết và thực tế.

2.4 Tổng quát hoá mô hình

Trong mô hình nêu trên, ta đã xử lí bài toán đơn giản nhất với một SFC và mạng vật lí. Mô hình hoàn toàn có thể mở rộng ra thành một mô hình tổng quan, với đầu vào là tập gồm nhiều SFC cần đặt lên mạng vật lí. Cụ thể:

Đầu vào (input)

- Tập hợp gồm các đồ thị chuỗi dịch vụ (SFC): $\mathcal{S} = \{s = \mathcal{G}_s(\mathcal{N}_s, \mathcal{E}_s)\}$
- Đồ thị mạng vật lí (PHY): $\mathcal{G}(\mathcal{N}, \mathcal{E})$

Đầu ra (output)

- Vị trí đặt các VNF trên nút mạng: $\phi_i^{v,s} = \begin{cases} 1 & \text{khi } v \text{ của SFC } s \text{ được đặt tại } i \\ 0 & \text{khi } v \text{ của SFC } s \text{ không được đặt tại } i \end{cases}$
- Vị trí đặt các liên kết logic trên các liên kết vật lí: $\phi_{ij}^{vw,s} = \begin{cases} 1 & \text{khi } vw \text{ của SFC } s \text{ được đặt tại } ij \\ 0 & \text{khi } vw \text{ của SFC } s \text{ không được đặt tại } ij \end{cases}$
- Trạng thái đặt SFC trên mạng vật lí: $\phi_s = \begin{cases} 1 & \text{khi SFC } s \text{ được đặt thành công trên mạng vật lí} \\ 0 & \text{khi SFC } s \text{ không được đặt trên mạng vật lí} \end{cases}$

Bảng 3: Các tham số của mô hình tổng quát

Tham số	Ý nghĩa
$\mathcal{S}$	Tập các SFC
$\mathcal{G}_s(\mathcal{N}_s, \mathcal{E}_s)$	Đồ thị chuỗi dịch vụ s
$\mathcal{N}_s$	Tập các VNF của chuỗi dịch vụ s
$\mathcal{E}_s$	Tập các kết nối logic giữa các VNF của chuỗi dịch vụ s
$\mathcal{G}(\mathcal{N}, \mathcal{E})$	Đồ thị mạng vật lí (PHY)
$\mathcal{N}$	Tập các nút vật lí
$\mathcal{E}$	Tập các liên kết vật lí

Bảng 4: Các biến của mô hình tổng quát

Biến	Ý nghĩa
$\phi_i^{v,s}$	Trạng thái đặt VNF vcủa SFC s tại nút vật lí i
$\phi_{ij}^{vw,s}$	Trạng thái đặt liên kết logic vw của SFC s tại liên kết vật lí ij
ϕ_s	Trạng thái đặt SFC s lên mạng vật lí (PHY)

Các điều kiện ràng buộc

- Tổng tài nguyên yêu cầu tại một nút vật lí không vượt quá tổng tài nguyên có sẵn tại nút vật lí đó:

$$\sum_{s \in S} \sum_{v \in N_s} \phi_i^{v,s}.r_v \leq a_i \qquad \forall i \in \mathcal{N} \tag{10}$$

- Tổng tài nguyên yêu cầu tại một liên kết vật lí không vượt quá tổng tài nguyên mà nút vật lí đó có:

$$\sum_{s \in S} \sum_{vw \in E_s} \phi_{ij}^{vw,s}.r_{vw} \leq a_{ij} \qquad \forall ij \in \mathcal{E} \tag{11}$$

- Mỗi nút vật lí chỉ chứa tối đa một VNF của một SFC:

$$\sum_{v \in \mathcal{N}_s} \phi_i^{v,s} \leq \phi_s \qquad \forall i \in \mathcal{N}, \forall s \in \mathcal{S} \tag{12}$$

- Mỗi VNF của mỗi SFC đều được chứa trong một nút vật lí:

$$\sum_{i \in \mathcal{N}} \phi_i^{v,s} = \phi_s \qquad \forall v \in \mathcal{N}_s, \forall s \in \mathcal{S} \tag{13}$$

- Mọi SFC đều được bảo toàn về chiều dịch vụ:

$$\sum_{j \in N} \phi_{ij}^{vw,s} - \sum_{j \in N} \phi_{ji}^{vw,s} = \phi_i^{v,s} - \phi_i^{w,s} \qquad \forall i \in \mathcal{N}, \forall vw \in \mathcal{E}_s, \forall s \in \mathcal{S} \tag{14}$$

Các bài toán tổng quát có thể đặt ra

- Tổng số nút và liên kết vật lí là nhỏ nhất:

$$\min\left(\phi_{\mathcal{G}}^{\mathcal{S}}\right) = \min\left(\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{N}_s} \phi_i^{v,s} + \sum_{s \in \mathcal{S}} \sum_{ij \in \mathcal{E}} \sum_{vw \in \mathcal{E}_s} \phi_{ij}^{vw,s}\right) \tag{15}$$

- Tổng lượng tài nguyên cần sử dụng là ít nhất:

$$\min\left(r_{\mathcal{G}}^{\mathcal{S}}\right) = \min\left(\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{N}_s} \phi_i^{v,s} r_v + \sum_{s \in \mathcal{S}} \sum_{ij \in \mathcal{E}} \sum_{vw \in \mathcal{E}_s} \phi_{ij}^{vw,s} r_{vw}\right) \tag{16}$$

- Tổng chi phí cần thiết sử dụng là nhỏ nhất (Giả sử giá của mỗi đơn vị tài nguyên sử dụng trên nút vật lí i là c_i, giá của mỗi đơn vị tài nguyên sử dụng trên liên kết vật lí ij là c_{ij}):

$$\min\left(c_{\mathcal{G}}^{\mathcal{S}}\right) = \min\left(\sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{N}} \sum_{v \in \mathcal{N}_s} \phi_i^{v,s} c_i r_v + \sum_{s \in \mathcal{S}} \sum_{ij \in \mathcal{E}} \sum_{vw \in \mathcal{E}_s} \phi_{ij}^{vw,s} c_{ij} r_{vw}\right) \tag{17}$$

- Độ trễ toàn phần của SFC là nhỏ nhất (Giả sử mỗi liên kết ij có một độ trễ không đổi là d_{ij}):

$$\min\left(d_{\mathcal{G}}^{\mathcal{S}}\right) = \min\left(\sum_{s \in \mathcal{S}} \sum_{ij \in \mathcal{E}} \sum_{vw \in \mathcal{E}_s} \phi_{ij}^{vw} d_{ij}\right) \tag{18}$$

- Tổng số SFC được đặt là lớn nhất:

$$\min\left(-\sum_{s \in \mathcal{S}} \phi_s\right) \tag{19}$$

Tương tự, ta cũng có thể đặt thêm những bài toán tổng quát khác để giải quyết các vấn đề khác nhau trong lí thuyết và thực tế.

3 Mô phỏng bài toán

Có nhiều phương án để triển khai bài toán này. Ở đây để đơn giản và nhanh chóng, chúng ta sẽ sử dụng Python Notebook (Jupyter) cùng thư viện NetworkX và PuLP để tìm nghiệm của bài toán này.

3.1 Bài toán đơn giản

Bài toán đơn giản nhất sẽ có các biến và tham số như đã đề cập trong Bảng 1 và 2, các điều kiện ràng buộc là các Phương trình 1, 2, 3, 4 và 5, với hàm mục tiêu là Phương trình 6.

Mã nguồn và kết quả thu được Mã nguồn của chương trình giải bài toán nêu trên được trình bày tại https://github.com/jerapiblaze/sfc-phy-mapping/blob/master/mapping-solo/mapping-solo.ipynb

3.2 Bài toán đơn giản dạng tổng quát

Bài toán đơn giản nhất sẽ có các biến và tham số như đã đề cập trong Bảng 3 và 4, các điều kiện ràng buộc là các Phương trình 10, 11, 12, 13 và 14, với hàm mục tiêu là Phương trình 15.

Mã nguồn và kết quả thu được Mã nguồn của chương trình giải bài toán nêu trên được trình bày tại `https://github.com/jerapiblaze/sfc-phy-mapping/blob/master/mapping-combine-maxima/mapping-combine-maxima.ipynb`

3.3 Nhận xét chung

Với phương pháp giải trực tiếp mô hình toán bằng các trình tối ưu toán học, kết quả thu được luôn là kết quả tốt nhất, tối ưu nhất. Tuy nhiên, thời gian cần thiết để thu được kết quả tối ưu này lại rất dài, lượng tài nguyên tiêu tốn cũng không nhỏ. Đặc biệt, với những bài toán cỡ lớn (mạng PHY phức tạp, có hàng trăm nút và liên kết, số lượng SFC, VNF cần đặt lớn, ...) sẽ tạo ra một mô hình toán vô cùng cồng kềnh và các trình tối ưu toán học không thể giải quyết trực tiếp được. Do đó, trong thực tế, người ta thường hạn chế giải quyết trực tiếp mô hình toán học bằng các trình tối ưu toán học chính xác mà thay vào đó sẽ sử dụng các cách tiếp cận khác. Một số phương án tiếp cận khác có thể kể đến như:

- Sử dụng GNN để tìm ra điểm tương đồng giữa mạng SFC và PHY, từ đó thực hiện mapping;
- Sử dụng RL để tìm ra phương án đặt mạng SFC vào mạng PHY;
- Sử dụng GNN để tăng tốc DRL trong quá trình tìm phương án đặt mạng SFC vào mạng PHY;
- Sử dụng GNN để tăng tốc quá trình giải mô hình của trình tối ưu toán học;

Các phương án này có ưu điểm là thời gian giải nhanh chóng, lượng tài nguyên sử dụng ít hơn so với việc tối ưu bằng trình tối ưu toán học chính xác. Tuy nhiên chúng đều sẽ không thể đưa ra các phương án tối ưu nhất mà chỉ đưa ra các kết quả cận tối ưu. Có nhiều tài liệu đã hiện thực hoá các phương án nêu trên, mỗi tài liệu lại có những hướng tiếp cận khác nhau, và có những ưu, nhược điểm khác nhau, nhưng chung quy lại đều hướng tới một mục tiêu duy nhất đó là tìm ra kết quả gần với kết quả tối ưu nhất với thời gian và tài nguyên cần thiết là ít nhất.