Technical Document

In this documentation, we will introduce complexity analysis for the original optimization problem, the result gap analysis and competitive analysis for our SFC-CEB algorithm. The result gap analysis mainly refers to analyze the reasons which cause our SFC-CEB algorithm can not obtain optimal embedding scheme with minimal cost for single SFC request. The difference between the total cost of embedding scheme which is found by SFC-CEB and the total cost of the optimal scheme is mainly determined by the relevant cost parameters (such as VNF operation cost, link forwarding delay, etc.) and the characteristics of the SFC request itself (the traffic size of the request, the expected completion time of the request, etc.). This makes it difficult to generalize an accurate mathematical model to assess specific result gap between our algorithm and theoretical optimal solution, so in this document we only complete qualitative analysis. On the other hand, the competitive analysis mainly refers to analyze the performance gap between SFC-CEB and offline optimal solution, since SFC-CEB is an online algorithm.

I. Optimization Problem Complexity Analysis

Assume we set $T = 1$ and we only consider the VNF running cost in original optimization problem shown in (6), the optimization objective of our problem will convert to following format:

$$C_R^{(t)} = \sum_{n \in [N]} \sum_{f \in [F]} \mu_{f,n} q_{f,n}^{(t)} \tag{1}$$

According to the definition of minimum knapsack problem (MKP): For a given set of items, each item has its own weight and value, the objective is to find a minimum weight subset of items such that the total value of these items meet specified demand. In our optimization problem, we can regard $\mu_{f,n}$ as the weight of item and φ_f as the value of item. Then the demand of MKP is: $\sum_{k\in[K]} \sum_{f'\in[F]} \sum_{n'\in[N]} \sum_{p\in[P^{n,n'}]} X_{k,p,f',n',f,n}^{(t)} R_{k,f}^{(t)}$ is satisfied by constraint (7a) ($q_{f,n}^{(t)}$ will be the set of items). Therefore, our optimization problem can be reduced from MKP. This means that our problem is at least as complex as MKP, which is NP-hard. The proof is complete.

II. Result Gap Analysis for SFC-CEB

In order to obtain optimal SFC request embedding scheme, a better approach is to traverse the entire path from source node to destination node of request in the multi-layer graph, and select the path with minimal cost. However, the computation complexity of this approach is unacceptable when the network topology is large size, especially for online problem. Therefore, we make a trade-off between algorithm complexity and cost optimization effectiveness. By modifying traditional shortest path algorithm, we improve the efficiency of the algorithm to find a feasible embedding scheme for SFC request. On the other hand, the embedding scheme found by the algorithm may not always have the lowest cost. Next, we will analysis the reasons that cause our algorithm may find near-optimal embedding scheme and the cost gap between near-optimal scheme and optimal scheme.

1) **Path Feasibility Misestimate:** Because the traditional shortest path algorithms do not need to consider the capacity limit of nodes or links, if we use them directly in multi-layer graph, the shortest path they find may violate resource capacity limitation. To avoid this problem, we introduce path feasibility checking in our algorithm. But this procedure leads to another problem. As shown in Fig. 1, the ingress node of request in multi-layer graph is A_{m-1}, egress node is F_{m+1}, and it remains two VNFs to be deployed. Assume that $path1 : A_{m-1} \to B_{m-1} \to B_m$ is the path with minimal cost to deploy VNF_1, but if we accept this path, there will not be enough resource capacity on node B to deploy VNF_2 (which means $path2 : A_{m-1} \to B_{m-1} \to B_m \to B_{m+1}$ is not feasible, even though it is the shortest path). In addition to $path2$, suppose $path3 : A_{m-1} \to C_{m-1} \to C_m \to D_m \to B_m \to B_{m+1}$ is the shortest path in multi-layer graph. However, the shortest path from A_{m-1} to B_m has already been stored as $path1$. When we traverse the neighbor nodes of D_m and B_m is traversed, the shortest path to B_m will still not be updated due to the rule of shortest path algorithm. In this case, $path3$ will not be found, and another path with higher embedding cost may be found (i.e. $path4 : A_{m-1} \to C_{m-1} \to C_m \to C_{m+1} \to D_{m+1} \to B_{m+1}$). This causes the shortest path algorithm may not found the optimal solution.

To solve this problem, we introduce the DFS procedure when traversing nodes in multi-layer graph. After adopting DFS procedure, when we the traverse neighbor nodes of D_m again, it will continue to traverse the neighbor nodes of the neighbor nodes of D_m. Suppose the depth of DFS procedure is 1, when B_m is traversed, it will continue to traverse the neighbor nodes of B_m. Because B_{m+1} will be traversed at this time, the shortest path to node B_{m+1} will be updated as $path3$. According to this example, we can see that DFS procedure can handle the problem that the optimal embedding scheme can not be found due to path feasibility checking.

2) **Weight Calculation Misestimate:** We introduce a new way of weight representation and calculation in modified shortest path algorithm, this may also cause our algorithm can not found optimal embedding scheme for SFC request. An example is shown in Fig. 1, there are two paths $path5 : A_{m-1} \to$

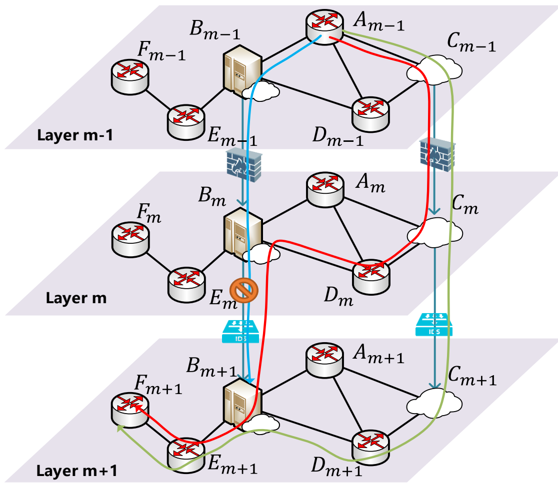

Fig. 1. SFC-CEB Misestimate Example

$C_{m-1} \rightarrow C_m \rightarrow D_m \rightarrow B_m \rightarrow B_{m+1} \rightarrow E_{m+1} \rightarrow F_{m+1}$ and $path6 : A_{m-1} \rightarrow C_{m-1} \rightarrow C_m \rightarrow C_{m+1} \rightarrow D_{m+1} \rightarrow B_{m+1} \rightarrow E_{m+1} \rightarrow F_{m+1}$ from ingress node A_{m-1} to egress node F_{m+1}. Assume that the weight factor quintuple at E_{m+1} is $[172, 15, 41, 38, 5]$ for $path5$, and $[190, 0, 39, 40, -3]$ for $path6$. According to the updating rule of our algorithm, $path5$ will be stored at E_{m+1}, since it has lower embedding cost (172 + 15 + 41 + 38 = 266 ¡ 190 + 0 + 39 + 40 = 269). If the forwarding delay is 3, bandwidth cost is 5 and resource weight is 1 for link $E_{m+1} \rightarrow F_{m+1}$, the weight factor quintuple will be $[172, 24, 46, 39, 8]$ for $path5$, and $[190, 0, 44, 41, 0]$ for $path6$ (suppose τ_k is 3). At this time, $path6$ has the lower cost. However, $path6$ can not be found because the shortest path at E_{m+1} has been stored as $path5$. This also causes that our algorithm can not found the optimal solution.

In fact, DFS procedure can also handle this problem. When E_{m+1} is traversed following the step of $path6$, if we continue to traverse the neighbor nodes of E_{m+1}, $path6$ can still be found. However, the misestimate caused by weight calculation may need deeper DFS procedure (in this example, if F_{m+1} is not the egress node, there are more nodes behind F_{m+1}, it needs to traverse more than 1 depth to make sure that optimal embedding scheme can be found). Larger depth in DFS procedure may cause vast computation complexity, this require us to make a trade-off between the accuracy of the results and the computation complexity. Please refer to the performance evaluation section of the original paper for the relationship between the depth of DFS procedure, the probability of finding the optimal solution and computation complexity.

III. Competitive Analysis for SFC-CEB

In order to analyze the competitive ratio between SFC-CEB and the offline optimal solution, we assume that SFC-CEB can find the embedding scheme with the lowest total embedding cost for each SFC request. Firstly, start with a simple scenario which includes two SFC requests $request\ A$ and $request\ B$. And the network environment only contains one remote public cloud and one edge cloud, that means there are only two embedding schemes can be selected for any request: a) Only deploy VNF instances on edge cloud; b) Deploy at least one VNF instance on remote public cloud. These schemes are classified on the basis that the traffic routing cost is lower if we only use the edge cloud to deploy VNF instances, and higher if the public cloud is used to deploy VNF instances.

To simplify representation, we use following variables to describe the characteristics of the request and different type costs. For each SFC request k, it has its own request size $size_k$, expected completion time ddl_k and timeout penalty τ_k. For VNF operation cost (includes VNF deployment cost and VNF running cost), we use D_P and D_E to denote the cost incurred by VNF instances for handling per unit size of request on public cloud and edge cloud respectively. For traffic routing cost, we use R_P and B_P to denote the end-to-end delay and bandwidth cost when request passes through public cloud, and R_E and B_E to denote the end-to-end delay and bandwidth cost when request only passes through edge cloud. By using these variable, the total embedding cost for any SFC request can be denoted by following equations:

If the SFC request passes through public cloud:

$$\begin{cases} D_P \cdot size_k + B_P & \text{if } ddl_k \geq R_P \\ D_P \cdot size_k + (R_P - ddl_k) \cdot \tau_k + B_P & \text{otherwise} \end{cases} \tag{2}$$

If the SFC request only passes through edge cloud:

$$\begin{cases} D_E \cdot size_k + B_E & \text{if } ddl_k \geq R_E \\ D_E \cdot size_k + (R_E - ddl_k) \cdot \tau_k + B_E & \text{otherwise} \end{cases} \tag{3}$$

We also introduce $C_{P,k}$, $C^{'}_{P,k}$, $C_{E,k}$ and $C^{'}_{E,k}$ to denote $D_P \cdot size_k + B_P$, $D_P \cdot size_k + (R_P - ddl_k) \cdot \tau_k + B_P$, $D_E \cdot size_k + B_E$ and $D_E \cdot size_k + (R_E - ddl_k) \cdot \tau_k + B_E$ respectively.

Suppose $request\ A$ arrives before $request\ B$, and edge cloud can only accommodate one request. We can analyze the competitive ratio as following:

1) For $request\ A$, if $D_P \cdot size_k + B_P$ (namely $C_{P,k}$) is the embedding scheme with minimal total cost ($ddl_A \geq R_P$), then no matter what the lowest-cost embedding scheme to $request\ B$ is, SFC-CEB has a competitive ratio of 1 to the offline optimal solution. The reason is that in this case there is no competition for resources on the edge cloud.

2) For $request\ A$, if $D_E \cdot size_k + B_E$ (namely $C_{E,k}$) is the embedding scheme with minimal total cost ($R_P \geq ddl_A \geq R_E$), resource competition may occur on the edge cloud depending on the situation of $request\ B$:

- If $D_P \cdot size_k + B_P$ (namely $C_{P,k}$) is the embedding scheme with minimal total cost ($ddl_B \geq R_P$) for $request\ B$, there will not happen resource competition on edge cloud. Hence, the competitive ratio between SFC-CEB and offline optimal solution is 1.
- If $D_E \cdot size_k + B_E$ (namely $C_{E,k}$) is the embedding scheme with minimal total cost ($R_P \geq ddl_B \geq R_E$) for $request\ B$, that means resource competition will occur. Considering edge cloud can only accommodate only one SFC request and $request\ A$ arrives before

than $request\ B$, $request\ B$ can only deploy required VNF instances on public cloud. In this case, if the final embedding scheme which embed $request\ A$ on edge cloud and $request\ B$ on public cloud has minimal total embedding cost, the competitive ratio will be 1. Otherwise, the competitive ratio will be:

$$\frac{C_{E,A}+C'_{P,B}}{C'_{P,A}+C_{E,B}} \leq \frac{C_{E,A}+C'_{P,B}}{C_{E,A}+C_{E,B}} \leq \frac{C'_{P,A}+C'_{P,B}}{C_{E,A}+C_{E,B}}$$

- If $D_P \cdot size_k + (R_P - ddl_k) \cdot \tau_k + B_P$ (namely $C'_{P,k}$) is the embedding scheme with minimal total cost ($ddl_B \leq R_P$) for $request\ B$, there will not happen resource competition on edge cloud. The competitive ratio between SFC-CEB and offline optimal solution is 1.
- If $D_E \cdot size_k + (R_E - ddl_k) \cdot \tau_k + B_E$ (namely $C'_{E,k}$) is the embedding scheme with minimal total cost ($ddl_B \leq R_E$) for $request\ B$, resource competition may also occur. If the final embedding scheme which embed $request\ A$ on edge cloud and $request\ B$ on public cloud has minimal total embedding cost, the competitive ratio will be 1. Otherwise, the competitive ratio will be:

$$\frac{C_{E,A}+C'_{P,B}}{C'_{P,A}+C'_{E,B}} \leq \frac{C_{E,A}+C'_{P,B}}{C_{E,A}+C'_{E,B}} \leq \frac{C'_{P,A}+C'_{P,B}}{C_{E,A}+C'_{E,B}}$$

3) For $request\ A$, if $D_P \cdot size_k + (R_P - ddl_k) \cdot \tau_k + B_P$ (namely $C'_{P,k}$) is the embedding scheme with minimal total cost ($ddl_A \leq R_P$), similar to case 1), there is no competition for resources on the edge cloud. The competitive ratio between SFC-CEB and offline optimal solution is always 1.

4) For $request\ A$, if $D_E \cdot size_k + (R_E - ddl_k) \cdot \tau_k + B_E$ (namely $C'_{E,k}$) is the embedding scheme with minimal total cost ($ddl_A \leq R_E$), similar to case 2), resource competition may occur on the edge cloud:

- If $D_P \cdot size_k + B_P$ (namely $C_{P,k}$) is the embedding scheme with minimal total cost ($ddl_B \geq R_P$) for $request\ B$, there will not happen resource competition on edge cloud. Hence, the competitive ratio between SFC-CEB and offline optimal solution is 1.
- If $D_E \cdot size_k + B_E$ (namely $C_{E,k}$) is the embedding scheme with minimal total cost ($R_P \geq ddl_B \geq R_E$) for $request\ B$, that means resource competition will occur. Considering edge cloud can only accommodate only one SFC request and $request\ A$ arrives before than $request\ B$, $request\ B$ can only deploy required VNF instances on public cloud. In this case, if the final embedding scheme which embed $request\ A$ on edge cloud and $request\ B$ on public cloud has minimal total embedding cost, the competitive ratio will be 1. Otherwise, the competitive ratio will be:

$$\frac{C'_{E,A}+C'_{P,B}}{C'_{P,A}+C_{E,B}} \leq \frac{C'_{E,A}+C'_{P,B}}{C_{E,A}+C_{E,B}} \leq \frac{C'_{P,A}+C'_{P,B}}{C_{E,A}+C_{E,B}}$$

- If $D_P \cdot size_k + (R_P - ddl_k) \cdot \tau_k + B_P$ (namely $C'_{P,k}$) is the embedding scheme with minimal total cost ($ddl_B \leq R_P$) for $request\ B$, there will not happen resource competition on edge cloud. The competitive ratio between SFC-CEB and offline optimal solution is 1.
- If $D_E \cdot size_k + (R_E - ddl_k) \cdot \tau_k + B_E$ (namely $C'_{E,k}$) is the embedding scheme with minimal total cost ($ddl_B \leq R_E$) for $request\ B$, resource competition may also occur. If the final embedding scheme which embed $request\ A$ on edge cloud and $request\ B$ on public cloud has minimal total embedding cost, the competitive ratio will be 1. Otherwise, the competitive ratio will be:

$$\frac{C'_{E,A}+C'_{P,B}}{C'_{P,A}+C'_{E,B}} \leq \frac{C'_{E,A}+C'_{P,B}}{C_{E,A}+C'_{E,B}} \leq \frac{C'_{P,A}+C'_{P,B}}{C_{E,A}+C'_{E,B}}$$

According to the above simple example, we can see that the main reason causes the competitive ratio between SFC-CEB and offline optimal solution greater than 1 is: SFC-CEB always assigns optimal embedding schemes for first arriving requests. Due to the resources limitation of edge clouds, if some early requests exhaust these limited resources, it may result in higher costs for subsequent requests. However, the offline optimal solution does not have this problem because it has a global view of the arrival of requests.

Now, we consider a set of requests, assume that if the request enters into the network, it will not leave anymore. The number of these requests is n, and the edge clouds in the network can hold s requests at the same time. For each request, there are three types of embedding scheme: 1) only use edge clouds to deploy VNF instances; 2) only use public cloud to deploy VNF instances and 3) use both edge clouds and public cloud to deploy VNF instances. When evaluate the specific embedding cost, we use $C_{E,k}$ and $C'_{E,k}$ to represent the scheme that only use edge clouds, and use $C_{P,k}$ and $C'_{P,k}$ to represent the scheme that only use public cloud or use both public cloud and edge clouds (Because there are higher routing costs whenever the public cloud is used).

When these n requests arrive in order, besides some requests require only edge clouds to deploy VNF instances in order to achieve minimal embedding cost, there are some requests whose optimal embedding scheme is use public cloud to deploy VNF instances. Considering this situation, we suppose that the edge cloud resources are exhausted when the number of requests reaches s', where $s' \geq s$. Among these s' requests, SFC-CEB find the optimal embedding schemes which only use edge clouds for s_1 requests ($s_1 \leq s$). And then, there are $s - s_1$ requests whose optimal embedding scheme use both edge clouds and edge cloud, and the residual $s' - s$ requests have the optimal embedding schemes which only use public cloud. Now the total embedding cost for all n requests achieved by SFC-CEB is shown as following:

$$
\overbrace{C_E^1 + C_E^2 + \cdots + C_E^k}^{s_1} + \overbrace{C_P^1 + C_P^2 + \cdots + C_P^k}^{s-s_1} +
\overbrace{C_P^1 + C_P^2 + \cdots + C_P^k}^{s^{'}-s} + \overbrace{C_P^1 + C_P^2 + \cdots + C_P^k}^{n-s^{'}} =
\overbrace{C_E^1 + C_E^2 + \cdots + C_E^k}^{s_1} + \overbrace{C_P^1 + C_P^2 + \cdots + C_P^k}^{s-s_1} +
\overbrace{C_P^1 + C_P^2 + \cdots + C_P^k}^{n-s} \tag{4}
$$

In this equation, C_E^k represents $C_{E,k}$ or $C_{E,k}^{'}$, C_P^k represents $C_{P,k}$ and $C_{P,k}^{'}$. The exact value of C_E^k and C_P^k depend on the actual situation.

If the above embedding scheme is optimal, the competitive ratio will be 1. If not, the offline optimal solution must find a different embedding scheme from SFC-CEB. In this situation, we suppose that offline optimal solution find s_2 requests whose optimal embedding scheme only use edge clouds ($s_2 \neq s_1$, and $s_2 \leq s$). Then, there are $s - s_2$ requests whose optimal embedding scheme use both edge clouds and edge cloud, and the residual $s^{'} - s$ requests have the optimal embedding schemes which only use public cloud. the total embedding cost for all n requests achieved by offline optimal solution is shown as following:

$$
\overbrace{C_E^1 + C_E^2 + \cdots + C_E^k}^{s_2} + \overbrace{C_P^1 + C_P^2 + \cdots + C_P^k}^{s-s_2} +
\overbrace{C_P^1 + C_P^2 + \cdots + C_E^k}^{n-s} \tag{5}
$$

Next, we can analyze the competitive ratio as following:

$$
\begin{aligned}
&\frac{\overbrace{C_E^1 + \cdots + C_E^k}^{s_1} + \overbrace{C_P^1 + \cdots + C_P^k}^{s-s_1} + \overbrace{C_P^1 + \cdots + C_P^k}^{n-s}}{\overbrace{C_E^1 + \cdots + C_E^k}^{s_2} + \overbrace{C_P^1 + \cdots + C_P^k}^{s-s_2} + \overbrace{C_P^1 + \cdots + C_E^k}^{n-s}} \\
&\leq \frac{|s_1| \cdot max(C_E^k) + |n - s_1| \cdot max(C_P^k)}{|s_2| \cdot min(C_E^k) + |n - s_2| \cdot min(C_P^k)} \\
&\leq \frac{|s_1| \cdot max(C_E^k, C_P^k) + |n - s_1| \cdot max(C_E^k, C_P^k)}{|s_2| \cdot min(C_E^k, C_P^k) + |n - s_2| \cdot min(C_E^k, C_P^k)} \\
&\leq \frac{|n| \cdot max(C_E^k, C_P^k)}{|n| \cdot min(C_E^k, C_P^k)} \leq \frac{max(C_E^k, C_P^k)}{min(C_E^k, C_P^k)}
\end{aligned} \tag{6}
$$

In this equation, $max(C_E^k)$ and $max(C_P^k)$ refers to find the maximal value of C_E^k and C_P^k respectively. $min(C_E^k)$ and $min(C_P^k)$ refers to find the minimal value of C_E^k and C_P^k respectively. $max(C_E^k, C_P^k)$ refers to find the maximal value between C_E^k and C_P^k, and $min(C_E^k, C_P^k)$ refers to find the minimal value between C_E^k and C_P^k. According to the analysis, we can see that the competitive ratio between SFC-CEB and offline optimal solution will not exceed $max(C_E^k, C_P^k)/min(C_E^k, C_P^k)$. The specific value depends on the setting of network environment parameters, which includes VNF operation cost parameter, SLA violation cost parameter, bandwidth cost parameter and so on.