



Vol. 13/ No. 49/Autumn 2023

Research Article

# Placement of Parallelized Service Function Chain by Reusing Virtual Network Functions in Fog-Cloud Computing-Based Networks

Fatemeh Zahedi, PhD Student <sup>1</sup>  | Mohammadreza Mollahoseini Ardakani, Assistant Professor <sup>2\*</sup>  | Ahmad Heidary-Sharifabad, Assistant Professor <sup>3</sup> 

<sup>1</sup>PhD Student, Department of Computer Engineering, Maybod Branch, Islamic Azad University, Maybod, Iran, zahediifa@gmail.com

<sup>2</sup>Assistant Professor, Department of Computer Engineering, Maybod Branch, Islamic Azad University, Maybod, Iran, mr.mollahoseini@iau.ac.ir

<sup>3</sup>Assistant Professor, Department of Computer Engineering, Maybod Branch, Islamic Azad University, Maybod, Iran, ahmad.heidary@iau.ac.ir

**Correspondence**

Mohammadreza Mollahoseini Ardakani, Assistant Professor of Computer Engineering, Maybod Branch, Islamic Azad University, Maybod, Iran, mr.mollahoseini@iau.ac.ir

**Received:** 16 December 2022

**Revised:** 10 January 2023

**Accepted:** 21 February 2023

## Abstract

Network function virtualization technology transforms hardware middleboxes into sets of software-based Virtual Network Function (VNF) that can host the growing demand for latency-sensitive services at the fog-cloud computing-based networks. Dynamic placement of service functions chain by reusing (VNF) instances can improve resource utilization and save time. To address this problem, we propose a parallelized service function chain placement method by reusing (VNF) in fog-cloud computing-based networks. Here, the placement problem is configured using deep reinforcement learning approaches with the aim of maximizing long-term cumulative reward. By sharing (VNF) s in parallel, this method can achieve computational acceleration in providing online services. In addition, the proposed method increases the ability to accept future requests by extracting the distribution of the initialized (VNF) s. The simulation results show the superiority of the proposed method, where considering the monetary cost criterion of more than 7%, it performs better than the best existing methods.

**Keywords:** Service function chain, virtual network function, fog-cloud computing-based networks.

## Highlights

- Increasing the productivity of fog computing resources based on optimizing the location of service function chains.
- Responding to the growing demand for delay-sensitive cloud computing services.
- Minimizing the amount of processor consumption in cloud-fog data centers in order to reduce energy consumption.

**Citation:** F. Zahedi, M. Mollahoseini Ardakani, and A. Heidary-Sharifabad, "Placement of Parallelized Service Function Chain by Reusing Virtual Network Functions in Fog-Cloud Computing-based Networks," *Journal of Southern Communication Engineering*, vol. 13, no. 49, pp. 49–64, 2023, doi: 10.30495/jce.2023.1975209.1181, (in Persian).

## مقاله پژوهشی

# مکان‌یابی زنجیره توابع سرویس موازی شده با استفاده مجدد از توابع شبکه مجازی در شبکه‌های مبتنی بر محاسبات ابر-مه

فاطمه زاهدی<sup>۱</sup> | محمدرضا ملاحسینی اردکانی<sup>۲\*</sup> | احمد حیدری شریف‌آبادی<sup>۳</sup> 

## چکیده:

مجازی‌سازی تابع شبکه، جعبه‌های میانی سخت‌افزاری را به مجموعه‌ای از تابع شبکه مجازی مبتنی بر نرم‌افزار تبدیل می‌کند که می‌تواند میزبان تقاضای رو به رشد سرویس‌های حساس به تأخیر در شبکه‌های مبتنی بر محاسبات ابر-مه باشد. مکان‌یابی پویا زنجیره توابع سرویس با استفاده مجدد از نمونه‌های توابع شبکه می‌تواند منجر به بهبود استفاده از منابع و صرفه‌جویی در زمان شود. برای رسیدگی به این مسئله، ما یک روش مکان‌یابی زنجیره توابع سرویس موازی شده با استفاده مجدد از تابع شبکه در شبکه‌های مبتنی بر محاسبات ابر-مه را پیشنهاد می‌کنیم. در اینجا، مسئله مکان‌یابی با استفاده از رویکردهای یادگیری تقویتی عمیق باهدف حداکثر سازی پاداش جمعی بلندمدت پیکربندی شده است. این روش با اشتراک گذاشته توابع شبکه به صورت موازی می‌تواند شتاب محاسباتی را در ارائه سرویس‌های آنلاین محقق کند. علاوه بر این، روش پیشنهادی با استخراج توزیع توابع شبکه مقداردهی شده توانایی پذیرش درخواست‌های آینده را افزایش می‌دهد. نتایج شبیه‌سازی برتری روش پیشنهادی نشان می‌دهد، جایی که با در نظر گرفتن معیار هزینه پولی بیش از ۷٪ نسبت به بهترین روش‌های موجود عملکرد بهتری دارد.

**کلید واژه‌ها:** زنجیره تابع سرویس، تابع شبکه مجازی، شبکه‌های مبتنی بر محاسبات ابر-مه.

<sup>۱</sup> دانشجوی دکتری گروه کامپیوتر، واحد میبد، دانشگاه آزاد اسلامی، میبد، ایران، zahediifa@gmail.com

<sup>۲</sup> استادیار گروه کامپیوتر، واحد میبد، دانشگاه آزاد اسلامی، میبد، ایران، mr.mollahoseini@iau.ac.ir

<sup>۳</sup> استادیار گروه کامپیوتر، واحد میبد، دانشگاه آزاد اسلامی، میبد، ایران، ahmad.heidary@iau.ac.ir

نویسنده مسئول

\* محمدرضا ملاحسینی اردکانی، استادیار گروه کامپیوتر، واحد میبد، دانشگاه آزاد اسلامی، میبد، ایران، mr.mollahoseini@iau.ac.ir

تاریخ دریافت: ۲۵ آذر ۱۴۰۱

تاریخ بازنگری: ۲۰ دی ۱۴۰۱

تاریخ پذیرش: ۲ اسفند ۱۴۰۱

<https://doi.org/10.30495/jce.2023.1975209.1181>

## ۱-مقدمه

در سال‌های اخیر، تقاضا برای سرویس‌های محاسبات فشرده به دلیل ظهور فناوری‌هایی نظیر اینترنت اشیا به شدت رشد داشته است. شبکه‌های مبتنی بر محاسبات ابر-مه می‌توانند بستر مناسبی برای فراهم‌سازی منابع موردنیاز این تقاضاها باشند [۱]. زیرساخت شبکه‌های مبتنی بر محاسبات ابر-مه بر اساس سه لایه ابر، مه و اینترنت اشیا پیکربندی شده است، جایی که ابر در بالاترین سطح قرار دارد و پس‌از آن لایه مه است و در نهایت دستگاه‌های اینترنت اشیا در پایین‌ترین سطح هستند. شبکه‌های مبتنی بر محاسبات ابر-مه می‌تواند انعطاف‌پذیری و چابکی شبکه را با استقرار گره‌های مه در لبه شبکه بهبود بخشد [۱، ۲]. فراهم‌سازی سرویس برای هر تابع شبکه (نظیر، دروازه، مسیریاب، پروکسی، فایروال، بازرسی عمیق بسته<sup>۱</sup> (DPI)، مترجم آدرس شبکه<sup>۲</sup> (NAT) و سیستم تشخیص نفوذ<sup>۳</sup> (IDS)) در زیرساخت فعلی شبکه‌های مبتنی بر محاسبات ابر-مه از طریق زنجیره‌بندی منظم جعبه‌های میانی اختصاصی انجام می‌شود [۳]. با این وجود، این جعبه‌های میانی گران‌قیمت هستند و فراهم‌کنندگان

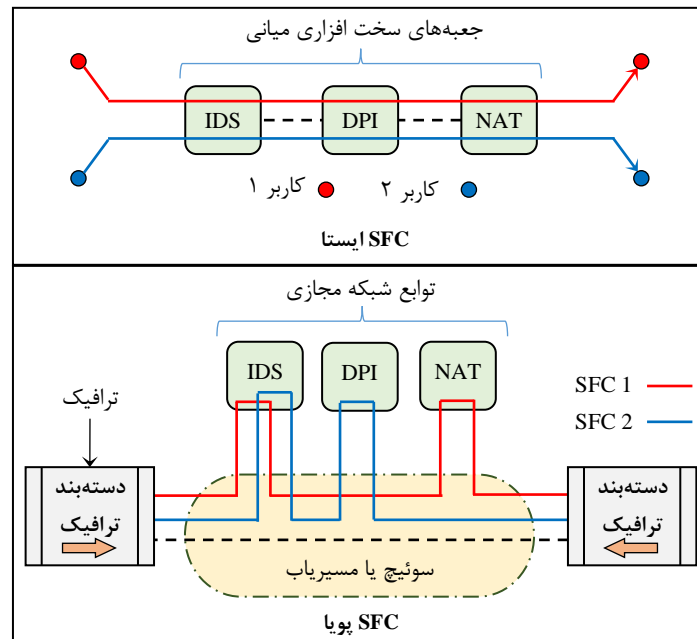
<sup>۱</sup> Deep Packet Inspection

<sup>۲</sup> Network Address Translation

<sup>۳</sup> Intrusion Detection System

سرویس برای کاهش هزینه‌های عملیاتی، هزینه سرمایه‌ای و ارائه خدمات جدید در رقابت هستند [۴]. برای مقابله با این چالش، فناوری نوظهور مجازی‌سازی تابع شبکه<sup>۱</sup> (NFV) یکپارچه‌سازی توابع شبکه را از طریق تبدیل جعبه‌های میانی به برنامه‌های کاربردی مبتنی بر نرم‌افزار فراهم می‌کند [۱].

به‌طور کلی، NFV می‌تواند توابع شبکه مجازی را برای فراهم‌سازی یک تابع سرویس به نام زنجیره تابع سرویس<sup>۲</sup> (SFC) به هم متصل کند [۴، ۵]. زنجیره تابع سرویس‌ها با تشکیل زنجیره‌های سرویس مجازی توسط توابع شبکه مجازی، باعث بهبود مقیاس‌پذیری و انعطاف‌پذیری شبکه در راه‌اندازی انواع جریان‌های ترافیک می‌شوند. روش NFV با استقرار توابع شبکه مجازی‌ها به‌عنوان کانتینرها یا ماشین‌های مجازی روی گره‌های فیزیکی در شبکه‌های مبتنی بر محاسبات ابر-مه پردازش موازی چندین عملیات را برای پشتیبانی کردن برنامه‌های بلادرنگ محقق می‌سازد؛ بنابراین، فراهم‌کنندگان سرویس می‌توانند از طریق NFV تا حد زیادی زمان عرضه سرویس و هزینه‌ها را کاهش دهند [۶]. یک درخواست زنجیره تابع سرویس از یک کاربر به‌عنوان مبدأ به یک سرور به‌عنوان مقصد ارسال می‌شود که شامل مجموعه‌ای مرتبی از توابع شبکه مجازی‌ها است. بسته ورودی در زنجیره تابع سرویس ایستا از همه توابع شبکه موجود عبور می‌کند، اما زنجیره تابع سرویس پویا عبور بسته را تنها از توابع شبکه مجازی‌های درخواست شده هدایت می‌کند. در اینجا، کنترل‌کننده در شبکه‌های مبتنی بر محاسبات ابر-مه می‌تواند بسته را به‌صورت هوشمند توسط دسته‌بند ترافیک به یک تابع شبکه خاص ارسال کند. شکل ۱ مثالی از زنجیره تابع سرویس‌های ایستا و پویا را نشان می‌دهد.



شکل ۱: مثالی از SFC ایستا و پویا [۲]

Figure 1. An example of static SFC and dynamic SFC [2]

همان‌طور که نشان داده‌شده است، شبکه شامل سه میان‌افزار (یعنی، IDS، DPI و NAT) است که بسته‌های مرتبط با کاربران ۱ و ۲ طبق شکل ۱ در مدل ایستا از همه زنجیره‌ها عبور می‌کند. با این حال، در مدل پویا، توابع شبکه مجازی‌ها جایگزین جعبه‌های میانی شده و بسته SFC1 را از IDS و NAT و بسته SFC2 را از IDS و DPI عبور می‌دهد. مطابق الزامات کیفیت خدمات، مکان‌یابی زنجیره تابع سرویس‌ها با (۱) استقرار توابع شبکه مجازی‌های مناسب روی گره‌ها، (۲) تعبیه پیوند بین گره‌ها بر اساس ترتیب توابع شبکه مجازی‌ها فراهم می‌شوند [۷]. در مکان‌یابی زنجیره تابع سرویس می‌توان از نمونه‌های توابع شبکه مجازی مقداره‌ی شده برای تشکیل زنجیره تابع سرویس‌های جدید استفاده کرد، جایی که این در اغلب مطالعات نادیده گرفته شده است. استفاده مجدد از توابع شبکه مجازی‌ها می‌تواند منجر به بهبود استفاده از منابع و کاهش زمان برنامه‌ریزی شود. به‌طور سنتی، زنجیره تابع سرویس‌ها به‌صورت سریالی مونتاژ می‌شوند و در آن ترافیک از همه توابع شبکه مجازی‌های موجود

<sup>1</sup> Network Function Virtualization

<sup>2</sup> Service Function Chains

در شبکه عبور می‌کند [۸]. روش SFC سریالی، تأخیر را افزایش می‌دهد که این امر با شبکه‌های نسل پنجم همخوانی ندارد. این مسئله توسط زنجیره تابع سرویس موازی شده رسیدگی می‌شود که در آن شتاب محاسباتی با اشتراک‌گذاری برخی توابع شبکه مجازی‌ها در زنجیره تابع سرویس فراهم می‌شود. اساس، هر جفت توابع شبکه مجازی بدون تضاد روی ترافیک می‌توانند بسته/جریان را به صورت هم‌زمان پردازش کنند [۹]. زنجیره تابع سرویس موازی شده می‌تواند با کاهش طول زنجیره تابع سرویس منجر به کاهش تأخیر شود. طیف وسیعی از مطالعات از یادگیری تقویتی عمیق برای حل مسئله مکان‌یابی زنجیره تابع سرویس استفاده می‌کنند [۱۰]. یادگیری تقویتی عمیق با محیط پویا سازگاری است و قابلیت یافتن مکان‌های مناسب برای استقرار توابع شبکه مجازی‌ها را به صورت آنلاین دارد. بر این اساس ما یک مدل برنامه‌ریزی پویا با استفاده از الگوریتم عملگر-نقاد مزیت ناهمگن به عنوان یک تکنیک یادگیری تقویتی عمیق جدید برای بهبود مکان‌یابی زنجیره تابع سرویس‌ها استفاده می‌کنیم. مشارکت اصلی این تحقیق به شرح زیر است:

- مکان‌یابی زنجیره تابع سرویس پویا از طریق استفاده مجدد از نمونه‌های توابع شبکه مجازی مقداردهی شده
  - تبدیل زنجیره تابع سرویس‌های سریال به زنجیره تابع سرویس‌های موازی شده بر اساس گراف سرویس برای بهبود شتاب محاسباتی
  - بهبود استقرار توابع شبکه مجازی‌ها روی گره‌ها بر اساس استخراج توزیع توابع شبکه مجازی‌های مستقر شده
- توسعه یک الگوریتم عملگر-نقاد مزیت ناهمگن، برای مکان‌یابی زنجیره تابع سرویس‌های موازی شده باهدف حداکثر سازی پاداش تجمعی بلندمدت ادامه این مقاله به شرح زیر سازمان‌دهی شده است: بخش ۲ ادبیات تحقیق را شرح می‌دهد، بخش ۳ مسئله را فرموله سازی می‌کند، بخش ۴ روش پیشنهادی را برای مکان‌یابی زنجیره تابع سرویس ارائه می‌دهد، بخش ۵ مرتبط با ارزیابی عملکرد روش پیشنهادی است و در نهایت، بخش ۶ مقاله را به پایان می‌رساند.

## ۲- پیشینه تحقیق

بسیاری از مطالعات برای مکان‌یابی زنجیره تابع سرویس به هزینه‌های عملیاتی و هزینه سرمایه‌ای پرداخته‌اند [۱۱، ۱۲]. باین حال، تأخیر، استفاده بهینه از منابع، مصرف انرژی و درآمد از دیگر نگرانی‌های اپراتورهای در مکان‌یابی زنجیره تابع سرویس است [۱۳]. اساساً، کارهای مرتبط در این حوزه به سه دسته مختلف تقسیم می‌شوند: زنجیره تابع سرویس سبز، مسیریابی زنجیره تابع سرویس و مکان‌یابی زنجیره تابع سرویس [۱۴]. این مقاله روی مکان‌یابی زنجیره تابع سرویس متمرکز است و ما آثار مرتبط در این دسته را شرح می‌دهیم.

یانگ و همکاران [۱۵] یک الگوریتم مسیریابی تأخیر-آگاه برای مکان‌یابی زنجیره تابع سرویس‌ها در شبکه‌های مبتنی بر محاسبات ابر-مه پیشنهاد دادند. این الگوریتم به عنوان مکان‌یابی و مسیریابی توابع شبکه مجازی گرد کردن تصادفی بر اساس برنامه‌ریزی خطی عدد صحیح توسعه یافته است. گرد کردن تصادفی یک الگوریتم تقریب زمان چندجمله‌ای باهدف برآورد تأخیر و کاهش حداکثر بار پیوندهای شبکه است.

پی و همکاران [۱۲] مسئله تعبیه زنجیره تابع سرویس را برای محیط‌های ابری توزیع شده با در نظر گرفتن بار شبکه پویا توسعه دادند. نویسندگان دو الگوریتم شامل نقشه زنجیره تابع سرویس و شبکه عریض بر پایه توابع مجازی مختلف را به ترتیب برای مکان‌یابی زنجیره تابع سرویس‌ها و بهینه‌سازی تعداد توابع شبکه مجازی‌ها پیشنهاد دادند. نقشه زنجیره تابع سرویس مکان‌یابی را با استفاده از یک الگوریتم کوتاه‌ترین مسیر مبتنی بر گراف چندلایه با در نظر گرفتن هزینه توابع شبکه مجازی و هزینه منابع انجام می‌دهد و شبکه عریض بر پایه توابع مجازی مختلف باهدف کاهش زمان اجرا تعداد نمونه‌های استفاده شده را حداقل می‌کند.

سوبرامانیا و همکاران [۱۶] از شبکه‌های عصبی برای تعیین تعداد توابع شبکه مجازی‌های مورد نیاز استفاده کردند و مسئله مکان‌یابی زنجیره تابع سرویس‌ها را با در نظر گرفتن تأخیر سرویس‌ها برای شبکه‌های نسل پنجم حل کردند. همچنین، چن و همکاران [۱۷] مکان‌یابی زنجیره تابع سرویس را باهدف حداقل‌سازی تأخیر و هزینه سرویس به عنوان یک مسئله برنامه‌ریزی خطی عدد صحیح فرموله کردند. نویسندگان از مدل مارکوف پنهان برای حل مسئله در محیط‌های ابری استفاده کردند.

باماره و همکاران [۱۸] مسئله مکان‌یابی زنجیره تابع سرویس را باهدف کاهش زمان پاسخ به‌عنوان یک مسئله بهینه‌سازی ترکیبی فرموله کردند. نویسندگان از ترکیب شبیه‌سازی تبرید و الگوریتم شاخه و حد برای مکان‌یابی زنجیره تابع سرویس‌ها در محیط‌های چند ابری استفاده کردند. در تحقیقی دیگر، زی و همکاران [۱۹] دو الگوریتم شامل، مکان‌یابی توابع شبکه مجازی پویا DAFT<sup>۱</sup> و مکان‌یابی توابع شبکه مجازی پویا امکان‌پذیر FDAFT<sup>۲</sup> را برای مکان‌یابی پویا زنجیره تابع سرویس‌ها باهدف حداکثرسازی درآمد پیشنهاد دادند. مکان‌یابی توابع شبکه مجازی پویا یک رویکرد اولیه-دوگانه و مکان‌یابی توابع شبکه مجازی پویا امکان‌پذیر یک گراف امدادی مبتنی بر ظرفیت است.

کواح و همکاران [۲۰] مسئله مکان‌یابی پویا زنجیره تابع سرویس‌ها را به‌عنوان یک برنامه عدد صحیح مختلط برای محیط‌های ابری فرموله کردند. نویسندگان مسئله را باهدف حداقل‌سازی مصرف انرژی با استفاده از الگوریتم ژنتیک حل کردند و یک تکنیک حل‌کننده مشترک برای کاهش پیچیدگی زمانی ارائه دادند.

یادگیری تقویتی عمیق یک رویکرد جدید برای حل مسائل در محیط‌های پویا است که عامل را باهدف حداکثرسازی پاداش جمعی بلندمدت برای تعامل با محیط آموزش می‌دهد [۲۱]. در مطالعات زیادی توانایی یادگیری تقویتی عمیق برای حل مسئله مکان‌یابی پویا زنجیره تابع سرویس اثبات‌شده است [۲۱، ۲۲]. گو و همکاران [۲۲] از یادگیری تقویتی عمیق برای استقرار آنلاین زنجیره تابع سرویس در محیط مه استفاده کردند. همچنین، لی و همکاران [۲۳] یک الگوریتم یادگیری Q تطبیقی مبتنی بر یادگیری تقویتی عمیق را برای مکان‌یابی زنجیره تابع سرویس‌ها با پیچیدگی کم پیشنهاد دادند. هدف این الگوریتم حداقل‌سازی مصرف پردازنده و پهنای باند در شبکه‌های نسل پنجم است که به‌صورت برنامه‌ریزی خطی عدد صحیح فرموله شده است.

سولوزابال و همکاران [۲۴] از سیاست توجه به‌دانا<sup>۳</sup> مبتنی بر یادگیری تقویتی عمیق با در نظر گرفتن یک چارچوب رمزگذار-رمزگشا برای ضبط توزیع درخواست‌ها و مکان‌یابی زنجیره تابع سرویس‌ها استفاده کردند. ژنگ و همکاران [۲۵] مسئله مکان‌یابی زنجیره تابع سرویس را با یک روش ابتکاری به‌صورت آفلاین و همچنین یک روش یادگیری تقویتی عمیق به‌صورت آنلاین حل کردند. سان و همکاران [۲۶] از یک الگوریتم یادگیری Q مبتنی بر یادگیری تقویتی عمیق برای آموزش یک عامل در تعامل با محیط مه استفاده کردند و آن را برای حل مسئله مکان‌یابی زنجیره تابع سرویس پیشنهاد دادند.

وانگ و همکاران [۲۷] مسئله مکان‌یابی زنجیره تابع سرویس تطبیقی را بر اساس یادگیری تقویتی عمیق حل کردند و آن را یادگیری تقویتی عمیق کوتاهترین مسیر نامیدند. روش یادگیری تقویتی عمیق کوتاهترین مسیر شامل ترکیب مدل دنباله-به-دنباله و شبکه کانولوشن گراف است که به ترتیب برای استخراج ویژگی‌های شبکه و جمع‌آوری داده‌های درخواست‌های زنجیره تابع سرویس اعمال می‌شود. هدف این الگوریتم حداکثرسازی میانگین پاداش جمعی بلندمدت با مشاهده کارایی تصمیمان استقرار گذشته است. در تحقیقی دیگر، وانگ و همکاران [۲۸] یک روش یادگیری عمیق دوگانه برای مکان‌یابی زنجیره تابع سرویس با نام DDQP<sup>۴</sup> را پیشنهاد دادند. این الگوریتم از شبکه‌های عصبی عمیق برای حل چالش فضای حالت بزرگ در محیط پویا استفاده می‌کند. علاوه بر این، نویسندگان مکانیسمی برای تحمل خطا و ارائه سرویس‌های قابل اطمینان در این الگوریتم تعبیه کردند.

### ۳- مدل سیستم

درخواست‌های زنجیره تابع سرویس توسط واحد کنترل پذیرش در نزدیک‌ترین دامنه مه پردازش می‌شود. برای سهولت ما دامنه مه جاری را به‌عنوان گراف  $G=(N,E)$  در نظر می‌گیریم که  $N$  مجموعه گره‌ها است و  $E$  مجموعه پیوندها است. اجازه دهید  $n_i \in N$  گره  $i$ -ام و  $e_{ij} \in E$  پیوند ارتباطی بین  $n_i$  و  $n_j$  باشد. همچنین،  $|*|$  تعداد عناصر در یک مجموعه را نشان می‌دهد، به‌عنوان مثال،  $|N|$  تعداد گره‌های مه است.

<sup>1</sup> Dynamic VNF placement

<sup>2</sup> Feasible Dynamic VNF placement

<sup>3</sup> Bahdanau

<sup>4</sup> Dynamic Differential Queue Protocol

پارامتر  $V$  مجموعه انواع توابع شبکه مجازی‌های موجود در  $G$  را نشان می‌دهد، جائیکه  $v \in V$  به‌عنوان توابع شبکه مجازی نوع  $v$  است. همچنین،  $U$  مجموعه درخواست‌ها زنجیره تابع سرویس دریافت شده در یک دوره زمانی می‌باشد، جائیکه  $u \in U$  درخواست  $u$ -ام است. درخواست  $u$ -ام به‌عنوان  $(V_u, S_u, \alpha_u, \beta_u, T_u, \Delta_u)$  تعریف می‌شود. در اینجا،  $V_u = \{v_u^1, v_u^2, \dots, v_u^{|V_u|}\}$  مجموعه مرتب‌شده توابع شبکه مجازی‌ها است،  $S_u = \{s_u^1, s_u^2, \dots, s_u^{|V_u|}\}$  منابع محاسباتی انواع توابع شبکه مجازی‌های موردنیاز است،  $\alpha_u$  ترمینال مبدأ (یعنی،  $v_u^0$ ) است،  $\beta_u$  ترمینال مقصد ( $v_u^{|V_u|+1}$ ) است،  $T_u$  ترافیک است و  $\Delta_u$  زمان زنده ماندن درخواست  $u$ -ام است. بر این اساس،  $s_u^v \in S_u$  منابع محاسباتی موردنیاز درخواست  $u$  از توابع شبکه مجازی نوع  $v$  است، همان‌طور که در رابطه ۱، نشان داده شده است.

$$s_u^v = \begin{cases} > 0, & v \in V \{V_u\} \\ = 0, & v \notin V \{V_u\} \end{cases} \quad (1)$$

جائیکه،  $V \{V_u\}$  مجموعه انواع توابع شبکه مجازی‌های موجود در  $V_u$  است.

پارامتر  $a_u$  یک متغیر باینری برای تعیین پذیرش یا رد درخواست  $u$  در مکان‌یابی می‌باشد، جائیکه شرط پذیرش  $a_u$ ،  $\lambda = |V_u|$  است و  $\lambda$  بر اساس رابطه ۲، تعریف می‌شود. در واقع،  $\lambda < |V_u|$  به مفهوم عدم استقرار موفقیت‌آمیز حداقل یک توابع شبکه مجازی از  $u$  است. بنابراین، تمام توابع شبکه مجازی متعلق به درخواست  $u$  با  $a_u = 0$  باید رد شده و منابع اشغال شده آن‌ها آزاد شود.

$$\lambda = \sum_{v_u^i \in V_u} \sum_{n_j \in N} b_u^{i,j} \quad (2)$$

جائیکه،  $b_u^{i,j}$  یک متغیر باینری برای تعیین وضعیت استقرار  $v_u^i$  روی  $n_j$  است.

مسئله مکان‌یابی زنجیره تابع سرویس شامل چندین محدودیت است. برای مثال، منبع ذخیره سازی در  $n_j$  نباید بیش از حد رزرو شود، همان‌طور که در رابطه ۳، تعریف شده است. همچنین، ترافیک روی  $n_j$  نباید از پهنای باند خروجی آن بیشتر باشد، همان‌طور که در رابطه ۴، تعریف شده است. علاوه بر این، توابع شبکه مجازی‌های مقداردهی شده با منابع محاسباتی کافی در  $n_j$  می‌توانند برای سایر درخواست‌ها به اشتراک گذاشته شود، همان‌طور که در رابطه ۵، تعریف شده است.

$$\sum_{v \in V} p_v^{n_j} m_v \leq n_j^S \quad (3)$$

$$\sum_{u \in U} \sum_{i=0}^{|V_u|} b_u^{i,j} (b_u^{i,j} - b_u^{i+1,j}) T_u k_u^t a_u \leq n_j^B \quad (4)$$

$$\sum_{u \in U} s_u^v k_u^t a_u \leq c_v p_v^{n_j} \quad (5)$$

جائیکه،  $p_v^{n_j}$  تعداد توابع شبکه مجازی‌های نوع  $v$  مستقر شده روی  $n_j$  است،  $m_v$  هزینه ذخیره‌سازی توابع شبکه مجازی نوع  $v$  است،  $c_v$  هزینه محاسباتی توابع شبکه مجازی نوع  $v$  است،  $n_j^S$  ظرفیت ذخیره‌سازی  $n_j$  است،  $n_j^B$  پهنای باند خروجی  $n_j$  است و  $k_u^t$  یک متغیر باینری برای تعیین وضعیت درخواست  $u$  (اجرا یا عدم اجرا) در زمان  $t$  است. در اینجا،  $b_u^{i,j} - b_u^{i+1,j}$  بدین مفهوم است که پهنای باند تنها با استقرار دو توابع شبکه مجازی متوالی روی دو گره متفاوت مصرف می‌شود.

اهداف مختلفی در مسئله مکان‌یابی زنجیره تابع سرویس وجود دارد که اغلب متناقض هستند و باید از طریق یک مصالحه بین آن‌ها بهینه‌سازی شود. در اینجا، ما مصالحه بین سه هدف شامل هزینه تأخیر، هزینه مصرف منابع و هزینه توان عملیاتی را بر اساس رابطه ۶، در نظر می‌گیریم.

$$\text{Min } \eta_L C_L + \eta_R C_R - \eta_T C_T \quad (6)$$

جائیکه،  $C_L$ ،  $C_R$  و  $C_T$  به ترتیب هزینه تأخیر، هزینه مصرف منابع و هزینه توان عملیاتی است که بر اساس روابط ۷ تا ۹ محاسبه می‌شوند. همچنین،  $\eta_L$ ،  $\eta_R$  و  $\eta_T$  به ترتیب ضرایب تأثیر این اهداف هستند.

$$C_L = \sum_{u \in U} \sum_{k=0}^{|V_u|} \sum_{n_i \in N} \sum_{n_j \in N} b_u^{k,i} b_u^{i+1,j} l_{i,j} a_u \quad (7)$$

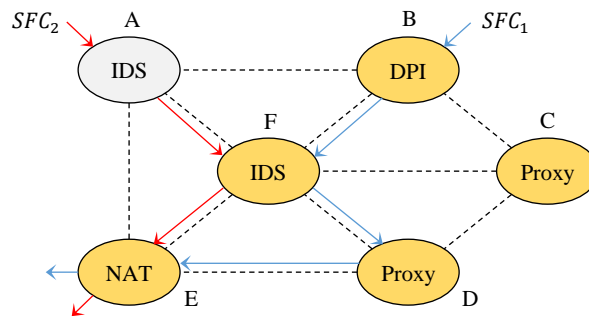
$$C_R = \zeta_N \sum_{u \in U} \sum_{i=0}^{|V_u|} \sum_{n_j \in N} b_u^{i,j} (b_u^{i,j} - b_u^{i+1,j}) T_u a_u + \zeta_L \sum_{u \in U} \sum_{v \in V} g_u^v m_v \quad (8)$$

$$C_T = \sum_{u \in U} \zeta_T [\Delta_u | V_u | T_u] a_u \quad (9)$$

جائیکه،  $I_{ij}$  تأخیر انتقال بین  $\pi_i$  و  $\pi_j$  است. متغیر  $g_u^v$  یک متغیر باینری است که وضعیت مقداردهی اولیه توابع شبکه مجازی نوع  $v$  را برای مکان‌یابی درخواست  $u$  نشان می‌دهد. همچنین،  $k_A$ ،  $k_B$  و  $k_C$  به ترتیب ضریب هزینه برای گره‌ها، پیوندها و توان عملیاتی هستند.

#### ۴- روش پیشنهادی

مسئله مکان‌یابی زنجیره تابع سرویس یک مسئله چند-هدفه مبتنی بر محدودیت است که الگوریتم‌های ابتکاری برای حل آن از زمان اجرای طولانی رنج می‌برند [۲۹]. این الگوریتم‌ها به دلیل محیط پویا توانایی پیش‌بینی آینده را ندارند و نمی‌توانند حداکثر پاداش تجمعی بلندمدت را فراهم کنند. رویکردهای یادگیری تقویتی عمیق می‌تواند مکان‌یابی پویا زنجیره تابع سرویس‌ها را تضمین کنند زیرا خاصیت مارکوف مرتبه اول را در شبکه‌های مبتنی بر محاسبات ابر-مه ارضا می‌کنند [۴]؛ بنابراین، این مسئله را می‌توان به عنوان فرآیند تصمیم‌گیری مارکوف در نظر گرفت. روش پیشنهادی برای حل تصمیم‌گیری مارکوف از رویکردهای مبتنی بر یادگیری تقویتی عمیق استفاده می‌کند که برای مصالحه بین تأخیر، منابع مصرف‌شده و توان عملیاتی از طریق حداکثرسازی پاداش تجمعی بلندمدت تلاش می‌کند. علاوه بر این، روش پیشنهادی پتانسیل استفاده مجدد از توابع شبکه مجازی‌ها را در مکان‌یابی زنجیره تابع سرویس برای حصول شتاب محاسباتی در نظر می‌گیرد. شکل ۲ مثالی از تکنیک استفاده مجدد از توابع شبکه مجازی‌ها را نشان می‌دهد.



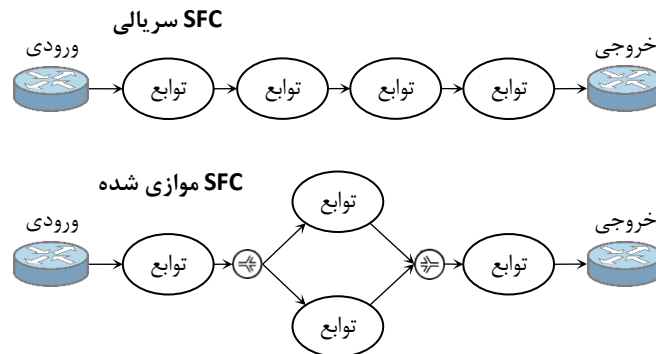
شکل ۲: تکنیک استفاده مجدد از توابع شبکه مجازی‌ها [۵]

Figure 2. The technique of reusing virtual network functions [5]

در اینجا،  $SFC_1$  شامل چهار توابع شبکه مجازی (یعنی، DPI، IDS، Proxy، NAT) و  $SFC_2$  شامل دو توابع شبکه مجازی (یعنی، NAT و IDS) است. تکنیک  $SFC_1$  بر اساس  $Path_1 = B \rightarrow F \rightarrow D \rightarrow E$  و  $SFC_2$  بر اساس  $Path_2 = A \rightarrow F \rightarrow E$  مکان‌یابی می‌شود. تکنیک  $SFC_2$  می‌تواند از طریق کوتاه‌ترین مسیر (یعنی،  $A \rightarrow E$ ) مکان‌یابی شود اما توابع شبکه مجازی IDS در A را مقداردهی جدید می‌کند. با این حال، F دارای منابع محاسباتی موردنیاز  $SFC_2$  برای توابع شبکه مجازی IDS- است و می‌تواند استفاده مجدد شود. اگرچه  $Path_2$  از مصرف ذخیره‌سازی ناشی از مقداردهی توابع شبکه مجازی جدید اجتناب می‌کند اما دارای مسیر طولانی‌تری است و منجر به افزایش تأخیر و پهنای باند می‌شود. بنابراین، استفاده مجدد از توابع شبکه مجازی‌ها یک مسئله چالش‌برانگیز است و برای همه زنجیره تابع سرویس‌ها مقرون به صرفه نیستند.

استراتژی موازی‌سازی توابع شبکه مجازی‌ها در یک زنجیره تابع سرویس یک موضوع داغ برای شتاب خدمات است [۷]. در این استراتژی برخی از توابع شبکه مجازی‌ها در یک زنجیره تابع سرویس به طور هم‌زمان پیکربندی می‌شوند. به طور معمول، هر جفت توابع شبکه مجازی که تضادی روی بسته عامل ایجاد نمی‌کنند می‌توانند به صورت موازی پیکربندی شوند. استراتژی موازی‌سازی توابع شبکه مجازی‌ها می‌تواند طول زنجیره تابع سرویس و در نتیجه تأخیر را کاهش دهد. مثالی از زنجیره تابع سرویس سریالی

در مقایسه با زنجیره تابع سرویس موازی شده در شکل ۳ نشان داده شده است. در اینجا، فرض شده است که توابع شبکه مجازی ۲ و توابع شبکه مجازی ۳ هیچ تضادی روی بسته‌ها ایجاد نمی‌کنند و می‌توانند به صورت موازی پیکربندی شوند.



شکل ۳: مثالی از SFC سریالی در مقایسه با SFC موازی شده [۷]  
Figure 3. An example of serial SFC compared to parallel SFC [7]

موازی‌سازی توابع شبکه مجازی‌ها تنها در صورت عدم وجود وابستگی بین آن‌ها امکان‌پذیر است. الگوریتم ۱ در مرجع [۳۰] می‌تواند صحت پردازش موازی بسته را در یک زنجیره تابع سرویس تضمین کند. این الگوریتم یک گراف سرویس تولید می‌کند که امکان پردازش موازی توابع شبکه مجازی‌ها را بر مبنای برخی اقدامات عملیاتی توابع شبکه مجازی‌ها نظیر خواندن/نوشتن بسته و ارسال/رها کردن بسته تعیین می‌کند. گراف سرویس خروجی در میان انواع مختلف گراف‌های سرویس که تعداد یکسان توابع شبکه مجازی‌ها را شامل می‌شوند، دارای کمترین تأخیر است. رابطه ۱۰، یک نشانگر است که وابستگی بین دو توابع شبکه مجازی متوالی در یک زنجیره تابع سرویس را نشان می‌دهد.

$$P(j, j+1) = \begin{cases} 1 & \text{if } VNF_j \text{ \& } VNF_{j+1} \text{ can run in parallel} \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

یادگیری در روش پیشنهادی مبتنی بر الگوریتم عملگر-نقاد مزیت ناهمگن است که از یک شبکه صف زنجیره تابع سرویس برای استخراج توزیع کارآمد توابع شبکه مجازی‌ها در طول زمان استفاده می‌کند. الگوریتم عملگر-نقاد مزیت ناهمگن یک الگوریتم پیشرفته در یادگیری تقویتی عمیق است که یادگیری را از طریق شبکه‌های عصبی بازیگر و منتقد با در نظر گرفتن چندین عامل مستقل انجام می‌دهد [۳۱]. هر مرحله از الگوریتم عملگر-نقاد مزیت ناهمگن شامل استقرار یک توابع شبکه مجازی از درخواست زنجیره تابع سرویس روی یک گره است. اگر توابع شبکه مجازی جاری با موفقیت مکان‌یابی شد، پاداش عامل محاسبه شده و آموزش برای مکان‌یابی توابع شبکه مجازی بعدی تکرار می‌شود. از طرف دیگر، عامل یک تنبیه هنگام مکان‌یابی ناموفق دریافت کرده و یک مکانیسم بازیابی برای آزاد کردن منابع اشغال شده توسط درخواست زنجیره تابع سرویس جاری اعمال می‌گردد. اجزای اصلی الگوریتم عملگر-نقاد مزیت ناهمگن شامل حالت، عمل، پاداش/تنبیه و سیاست است. عامل در هر دوره از یادگیری ( $\tau$ ) بر اساس سیاست و مشاهده حالت  $S_\tau$  در مورد انجام عمل  $A_\tau$  تصمیم‌گیری می‌کند. بر اساس  $S_\tau$  و  $A_\tau$ ، عامل پاداش  $R_\tau(S_\tau, A_\tau)$  را دریافت کرده و محیط به حالت بعدی  $S_{\tau+1}$  بروز می‌شود. در نهایت، عامل برای حداکثرسازی پاداش بلندمدت سیاست را بروز می‌کند.

حالت: مشخصات درخواست جاری  $u$  و جزئیات منابع شبکه یک حالت را ایجاد می‌کند. فاکتورهای  $\Delta_u$ ،  $T_u$ ،  $\beta_u$ ،  $S_u$ ،  $V_u$  و  $\mu_u$  و  $b_{ij}^u$  از درخواست ورودی و همچنین  $\bar{c}_j^S$ ،  $\bar{n}_j^S$  و  $\bar{n}_j^B$  از شبکه مه برای ایجاد یک حالت در نظر گرفته می‌شوند. در اینجا،  $\mu_u$  توان عملیاتی است که مطابق رابطه ۱۱ محاسبه می‌شود. پارامتر  $\bar{c}_j^S$  هزینه محاسباتی باقیمانده  $n_j$  در توابع شبکه مجازی نوع  $v$  است که در رابطه ۱۲ نشان داده شده است. بعلاوه،  $\bar{n}_j^S$  و  $\bar{n}_j^B$  به ترتیب منبع ذخیره‌سازی باقی‌مانده و پهنای باند خروجی باقی‌مانده  $n_j$  هستند، همان‌طور که در روابط ۱۳ و ۱۴ نشان داده شده است.

$$\mu_u = \zeta_\tau \Delta_u | V_u | T_u + TP_{lb} \quad (11)$$



$$\overline{c}_j^v = c_v p_v^{n_j} - \sum_{u \in U} s_u^v k_u^t a_u \quad (12)$$

$$\overline{n}_j^S = n_j^S - \sum_{v \in V} p_v^{n_j} m_v \quad (13)$$

$$\overline{n}_j^B = n_j^B - \sum_{u \in U} \sum_{i=0}^{|V_u|} b_u^{i,j} (b_u^{i,j} - b_u^{i+1,j}) T_u k_u^t \quad (14)$$

عمل: عامل در هر مرحله  $v_u^k \in V_u$  را روی  $n_j \in N$  مکان یابی می کند. بنابراین،  $A_\tau$  شامل مجموعه گره ها در کلونی مه جاری است، یعنی  $A_\tau \in N$ .

پاداش: عامل با تخصیص توابع شبکه مجازی جاری به یک گره پاداش دریافت می کند، یعنی مکان یابی موفقیت آمیز است. مکان یابی موفق به مفهوم ارضا منابع مورد نیاز توابع شبکه مجازی توسط گره انتخابی است؛ بنابراین، محدودیت های منبع ذخیره سازی، هزینه محاسباتی و پهنای باند خروجی باید بعد از هر  $A_\tau$  چک شود. در محدودیت منبع ذخیره سازی باید گره انتخابی (e.g.,  $n_j$ ) منبع ذخیره سازی مورد نیاز برای میزبانی از توابع شبکه مجازی جاری (e.g.,  $v_u^k$ ) را داشته باشد، یعنی، رابطه ۱۵ به طور مشابه در محدودیت هزینه محاسباتی باید رابطه ۱۶ نیز چک شود. علاوه بر این، ارضا محدودیت پهنای باند خروجی مطابق رابطه ۱۷ بررسی می شود.

$$\overline{n}_j^S \geq m_{v_u^k} \quad (15)$$

$$\overline{c}_j^{v_u^k} \geq s_u^{v_u^k} \quad (16)$$

$$\overline{n}_j^B \geq b_u^{k-1,j} (b_u^{k-1,j} - b_u^{k,j}) T_u \quad (17)$$

عدم ارضا این محدودیت ها به عنوان مکان یابی ناموفق در نظر گرفته می شود. برای سهولت محاسبه پاداش و تنبیه، سابقه همه پاداش ها برای درخواست جاری از طریق حافظه  $R_u^{mem}$  در دسترس قرار دارد. عامل با انتخاب یک گره برای توابع شبکه مجازی جاری از طریق ارضا همه محدودیت ها یک پاداش دریافت می کند، همان طور که در رابطه ۱۸، نشان داده شده است. در اینجا، پاداش بر اساس تأخیر، منابع مصرف شده و توان عملیاتی مطابق رابطه ۶، محاسبه شده است.

$$R_\tau = \eta_T C_\tau^T - \eta_L C_\tau^L - \eta_R C_\tau^R \quad (18)$$

جائیکه،  $C_\tau^L$ ،  $C_\tau^R$  و  $C_\tau^T$  به ترتیب هزینه تأخیر، هزینه منابع مصرف شده و هزینه توان عملیاتی هستند و از طریق روابط ۱۹ تا ۲۱ محاسبه می شوند.

$$C_\tau^L = \sum_{n_j \in N} b_u^{i,j} b_u^{i+1,j} l_{i,j} \quad (19)$$

$$C_\tau^R = \zeta_N \sum_{n_j \in N} b_u^{i-1,j} (b_u^{i-1,j} - b_u^{i,j}) T_u + \zeta_L g_u^{v_u^k} m_{v_u^k} \quad (20)$$

$$C_\tau^T = \zeta_T \Delta_u T_u \quad (21)$$

جائیکه،  $\hat{J}$  گره انتخاب شده در  $A_\tau$  است.

از طرف دیگر، ممکن است همه محدودیت های تعریف شده با انتخاب یک گره برای توابع شبکه مجازی جاری ارضا نشود. در اینجا، یک مقدار منفی به عنوان  $R_\tau$  اعمال می گردد تا انجام  $A_\tau$  توسط عامل با تنبیه مواجه شود. ما تنبیه را بر اساس سابقه پاداش های فوری چندگانه محاسبه می کنیم، همان طور که در رابطه ۲۲، نشان داده شده است.

$$R_\tau = -\zeta_T [\Delta_u | V_u | T_u] - \sum_{R_i \in R_u^{mem}} R_i \quad (22)$$

سیاست: روش پیشنهادی یک سیاست بر اساس احتمال توزیع توابع شبکه مجازی ها در طول زمان را برای انتخاب  $A_\tau$  در  $S_\tau$  فراهم می کند، همان طور که در رابطه ۲۳، نشان داده شده است. این سیاست می تواند پاداش بلندمدت را بر اساس به روزرسانی همه اطلاعات تاریخی حداکثر کند. با این حال، روش پیشنهادی به دنبال حداکثر سازی پاداش تخفیف تجمعی است، همان طور که در رابطه ۲۴، نشان داده شده است.

$$\psi_\tau(S_\tau, A_\tau) \in (0, 1], \quad \sum_{A'_\tau \in V_\tau} \psi_\tau(S_\tau, A'_\tau) = 1 \quad (23)$$

$$\psi_\tau^* = \arg \max_{\psi_\tau} \mathbb{E}_{\psi_\tau} \left[ \sum_{\tau=0}^{\infty} \omega R_\tau(S_\tau, A_\tau) \right] \quad (24)$$

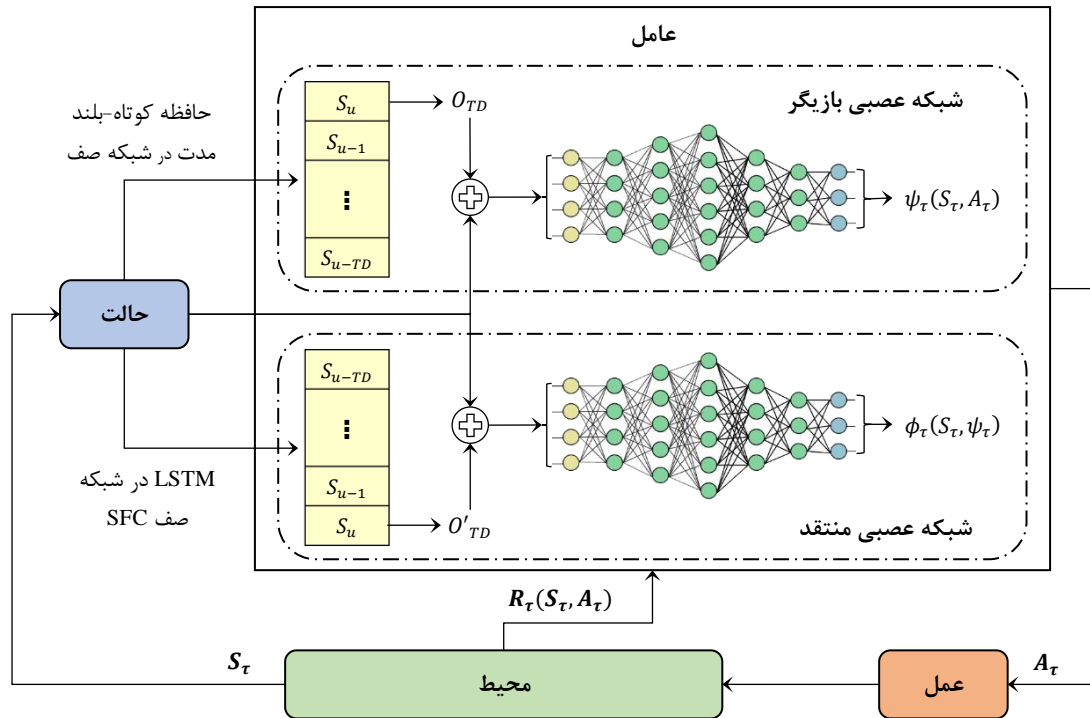
جائیکه،  $\psi_\tau(S_\tau, A_\tau)$  احتمال انجام  $A_\tau$  در  $S_\tau$  است و  $\omega$  یک فاکتور میرایی برای پاداش آینده است. به‌طور کلی، تخصیص توابع شبکه مجازی‌ها بر اساس توزیع آن‌ها در طول زمان منجر به توزیع مصرف منابع و در نتیجه افزایش تعداد درخواست‌های پذیرفته خواهد شد. از این‌رو، توزیع توابع شبکه مجازی‌ها در هر مرحله از یادگیری بر اساس شبکه صف زنجیره تابع سرویس استخراج می‌شود. شبکه صف زنجیره تابع سرویس، منابع محاسباتی مصرف‌شده توسط هر درخواست را در طول زمان ذخیره می‌کند (یعنی،  $S_{II}$ ). سپس، توزیع توابع شبکه مجازی‌های گذشته از طریق شبکه عصبی حافظه کوتاه-بلند مدت استخراج می‌شود. خروجی حافظه کوتاه-بلند مدت به عامل در انتخاب عمل  $A_\tau$  کمک می‌کند. روش پیشنهادی مکان‌یابی هر  $v_{II}^i \in V_{II}$  را یک-به-یک انجام می‌دهد، زیرا ترتیب توابع شبکه مجازی‌ها باید در نظر گرفته شود. در هر مرحله از یادگیری، هدف شبکه عصبی بازیگر تعیین سیاست انتخاب  $A_\tau$  توسط عامل است که بر اساس شبکه صف زنجیره تابع سرویس بروز می‌شود. همچنین، شبکه عصبی منتقد بهینه‌سازی  $R_\tau(S_\tau, A_\tau)$  را برای حصول بهترین تجربه یادگیری سریع و حداکثرسازی پاداش جمعی بر اساس  $A_\tau$  انتخاب شده انجام می‌دهد.

چارچوب یادگیری در روش پیشنهادی بر اساس الگوریتم عملگر-نقاد مزیت ناهمگن در شکل ۴ نشان داده شده است. در هر مرحله از یادگیری عامل  $S_\tau$  را با مشاهده محیط ایجاد می‌کند. سپس، شبکه صف زنجیره تابع سرویس بر اساس  $S_{II}$  بروز می‌شود. خروجی حافظه کوتاه-بلند مدت در شبکه صف زنجیره تابع سرویس حاوی توزیع توابع شبکه مجازی‌های موردنیاز (یعنی،  $O^{TD}$  و  $O^{TD}$ ) است که به‌همراه  $S_{II}$  به‌عنوان ورودی‌های بازیگر و منتقد در نظر گرفته می‌شود. طول حافظه شبکه صف زنجیره تابع سرویس به‌عنوان عمق زمانی توسط پارامتر یادگیری تفاوت زمانی  $TD^1$  تعریف می‌شود. در نهایت،  $\psi_\tau(S_\tau, A_\tau)$  و  $\phi_\tau(S_\tau, \psi_\tau)$  به ترتیب توسط شبکه‌های عصبی بازیگر و منتقد محاسبه می‌شوند.  $\psi_\tau$  بهترین  $A_\tau$  برای عامل است و  $\phi_\tau$  خروجی منتقد است که به فرایند یادگیری در بازیگر کمک می‌کند.

به‌طور کلی، نوآوری اصلی این تحقیق حل مسئله مکان‌یابی زنجیره تابع سرویس با استفاده از یک مکان‌یابی زنجیره تابع سرویس پویا مبتنی بر یادگیری تقویتی عمیق با توابع شبکه مجازی‌های موازی شده است. نوآوری دیگر این تحقیق استفاده مجدد از نمونه‌های توابع شبکه مجازی مقداردهی شده است تا شتاب محاسباتی در هنگام مکان‌یابی تضمین شود. علاوه بر این، ما برای بهبود استقرار توابع شبکه مجازی‌ها روی گره‌ها از یک تکنیک استخراج توزیع توابع شبکه مجازی‌های مستقر شده استفاده می‌کنیم.

## ۵- نتایج آزمایش‌ها

در این بخش روش پیشنهادی در مقایسه با برخی از الگوریتم‌های مکان‌یابی زنجیره تابع سرویس موجود مقایسه می‌شود. این الگوریتم‌ها شامل گرد کردن تصادفی [۱۵]، یادگیری تقویتی عمیق کوتاهترین مسیر [۲۷] و DDQP [۲۸] هستند. علاوه بر این، روش پیشنهادی با الگوریتم پایه SPFF [۳۱] مقایسه شده است. روش SPFF یک الگوریتم ابتکاری است که ابتدا کوتاه‌ترین مسیر را شناسایی می‌کند و سپس توابع شبکه مجازی‌ها را روی مسیرها از طریق اولین گره مناسب مستقر می‌کند. ما از متریک‌های کارایی نظیر نرخ استقرار، تأخیر، توان عملیاتی، هزینه مصرف منابع، نقض الزامات کیفیت سرویس و طول زنجیره تابع سرویس برای ارزیابی الگوریتم‌های مختلف استفاده می‌کنیم.



شکل ۴: چارچوب یادگیری در روش پیشنهادی بر اساس الگوریتم عملگر-نقاد مزیت ناهمگن

Figure 4. Learning framework in the proposed method based on the heterogeneous advantage operator-critic algorithm

### ۵-۱- راه‌اندازی شبیه‌سازی

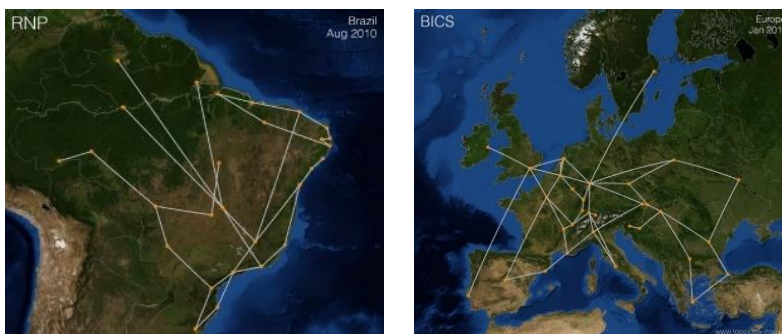
همه آزمایش‌ها توسط یک شبیه‌ساز طراحی شده با نرم‌افزار متلب ۲۰۱۹ انجام شده است. در اینجا، روش و سایر روش‌های مورد مقایسه با استفاده از این شبیه‌ساز راه‌اندازی شده است. ما شبیه‌سازی‌ها را برای ۴۰۰۰ درخواست زنجیره تابع سرویس انجام می‌دهیم. در همین حال، گره‌های مبدأ و مقصد از هر درخواست زنجیره تابع سرویس به صورت تصادفی از بین گره‌های شبکه انتخاب می‌شوند. هر درخواست در مدل ترافیک شبکه یک شکاف زمانی برای پردازش دارد که این باعث تشکیل یک صف زنجیره تابع سرویس می‌شود. هر الگوریتم یک درخواست زنجیره تابع سرویس را از صف برمی‌دارد و برای مونتاژ آن تلاش می‌کند. پس از پایان پردازش درخواست جاری، الگوریتم شروع به پردازش درخواست بعدی می‌کند؛ بنابراین، همه درخواست‌ها از اولویت یکسانی برخوردار هستند.

جدول ۱: پارامترهای مدل ترافیک شبکه

Table 1. Network traffic model parameters

مقدار	پارامتر
۱۰-۵ گیگا بیت بر ثانیه	پهنای باند لینک
۳۰-۲۵ میلی ثانیه	تأخیر پیوند
۸	تعداد انواع توابع شبکه مجازی
۸-۳ واحد	تقاضای منابع هر توابع شبکه مجازی
۲۵۰-۱۰۰ واحد	ظرفیت منابع هر گره
۱۰ واحد	تعداد منابع در هر گره
۶۰-۳۰ مگابیت بر ثانیه	ترافیک به ازای هر SFC
۸-۳ توابع شبکه مجازی	تعداد توابع شبکه مجازی در هر SFC
۲۰-۱۰ میلی ثانیه	زمان پردازش توابع شبکه مجازی
۱۰-۵ دلار	هزینه هر منبع

تأخیر انتقال در مدل ترافیک شبکه توسط پیوند نقطه-به-نقطه تخمین زده می‌شود. سایر پارامترهای مربوط به مدل ترافیک شبکه و راه‌اندازی شبیه‌سازی در جدول ۱ خلاصه شده است. در اینجا، مقادیر برخی پارامترها در یک محدوده تعریف شده است که برای هر درخواست به صورت تصادفی مشخص می‌شود. برای ارزیابی الگوریتم‌های مکان‌یابی زنجیره تابع سرویس مختلف از دو توپولوژی شبکه استفاده شده است که توپولوژی زوو<sup>۱</sup> در دسترس است. این توپولوژی‌ها شامل حلقوی و خطی هستند، همان‌طور که در شکل ۵ نشان داده شده است.



شکل ۵: توپولوژی‌های شبکه استفاده شده  
Figure 5. Network topologies used

## ۵-۲- تجزیه و تحلیل عملکرد

این بخش مرتبط با تجزیه و تحلیل عملکرد روش پیشنهادی در مقایسه با روش‌های پایه و پیشرفته در زمینه مکان‌یابی SFC است. مکان‌یابی زنجیره تابع سرویس در روش پیشنهادی با استفاده از الگوریتم عملگر-نقاد مزیت ناهمگن انجام می‌شود. الگوریتم عملگر-نقاد مزیت ناهمگن یک رویکرد جدید مبتنی بر یادگیری تقویتی عمیق است. در یک مقایسه آماری ما استفاده از الگوریتم عملگر-نقاد مزیت ناهمگن را توجیه می‌کنیم. جدول ۲ نشان می‌دهد که الگوریتم عملگر-نقاد مزیت ناهمگن در مقایسه با سایر رویکردهای یادگیری تقویتی عمیق نظیر بازیگر-منتقد و شبکه صف عمیق عملکرد بهتری در مکان‌یابی زنجیره تابع سرویس دارد. این برتری به وضوح روی هر دو توپولوژی شبکه حلقوی و خطی قابل مشاهده است. در این مقایسه برای هر متریک کارایی نتایج بر اساس میانگین و انحراف استاندارد روی ۴۰۰۰ درخواست ارائه شده است.

در همه توپولوژی‌ها الگوریتم عملگر-نقاد مزیت ناهمگن موفق به ارائه نرخ استقرار، تأخیر، توان عملیاتی، هزینه مصرف منابع، نقض الزامات کیفیت سرویس و طول زنجیره تابع سرویس بهتری در مقایسه با الگوریتم بازیگر-منتقد شده است. با این حال، الگوریتم عملگر-نقاد مزیت ناهمگن و شبکه صف عمیق نتایج رقابتی را در اغلب مقایسه‌ها ارائه می‌دهند. بر اساس انحراف استاندارد می‌توان برخی متریک‌های کارایی را در الگوریتم عملگر-نقاد مزیت ناهمگن و شبکه صف عمیق از نظر آماری مشابه در نظر گرفت. دلیل برتری نسبی الگوریتم عملگر-نقاد مزیت ناهمگن در مقایسه با شبکه صف عمیق این است که شبکه صف عمیق تنها از یک عامل واحد مبتنی بر شبکه‌های هدف برای تعامل با محیط استفاده می‌کند، اما الگوریتم عملگر-نقاد مزیت ناهمگن یادگیری را از طریق سیاست انجام می‌دهد که نسبت به مقدار Q پایدارتر است. بر این اساس، الگوریتم عملگر-نقاد مزیت ناهمگن وابسته به شبکه‌های هدف نیست و قابلیت کنش فضاهای پیوسته بهتری دارد. در مجموع، الگوریتم عملگر-نقاد مزیت ناهمگن با در نظر گرفتن همه متریک‌های کارایی در مقایسه با AC، A2C و شبکه صف عمیق به ترتیب ۱۲/۱٪، ۹/۶٪ و ۴/۳٪ روی توپولوژی حلقوی برتری دارد. این برتری برای توپولوژی خطی به ترتیب ۱۱/۵٪، ۱۰/۳٪ و ۳/۸٪ گزارش شده است.

همان‌طور که نشان داده شد، روش پیشنهادی با استفاده از الگوریتم عملگر-نقاد مزیت ناهمگن کارایی بهتری برای مکان‌یابی زنجیره تابع سرویس ارائه می‌دهد. در آزمایش دیگری روش پیشنهادی در مقایسه با برخی الگوریتم‌های موجود شامل SPFF<sup>۲</sup>

<sup>۱</sup> <http://www.topology-zoo.org>

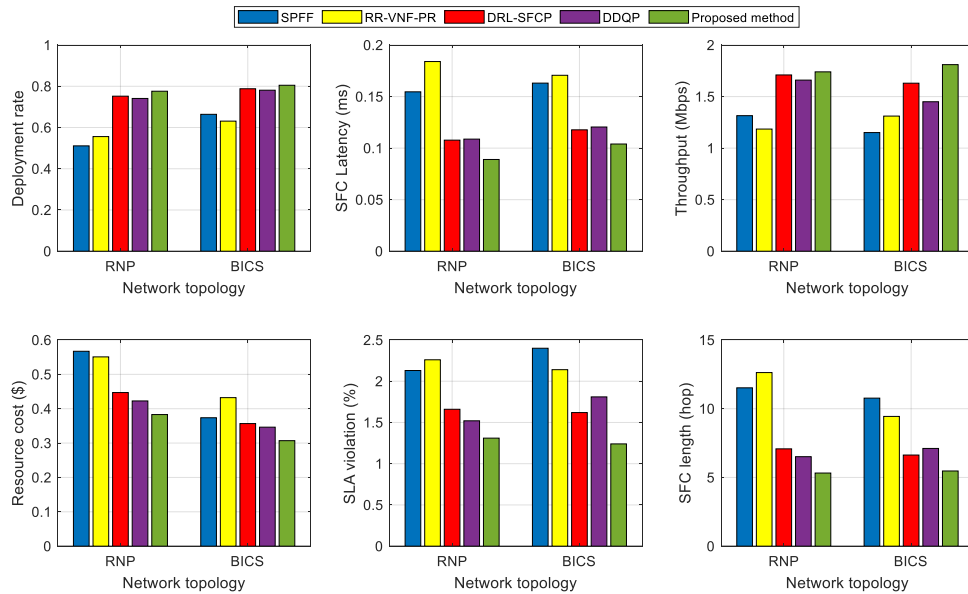
<sup>۲</sup> - Shortest Path First Fit

توابع شبکه مجازی،  $DDQP$  و  $DRL-SFCP^1$  ارزیابی می‌شود. نتایج این ارزیابی در شکل ۶ بر اساس نرخ استقرار، تأخیر زنجیره تابع سرویس، توان عملیاتی، هزینه مصرف منابع، نقض الزامات کیفیت سرویس و طول زنجیره تابع سرویس نشان داده شده است.

جدول ۲: عملکرد الگوریتم عملگر-نقاد مزیت ناهمگن در مقایسه با AC، A2C و شبکه صف عمیق

Table 2. Performance of heterogeneous advantage operator-critic algorithm compared to AC, A2C and DQN

توپولوژی	الگوریتم	نرخ استقرار	تأخیر SFC (ms)	توان عملیاتی (Mbps)	هزینه منابع (\$)	نقض الزامات کیفیت سرویس (%)	طول SFC (پرش)
RNP	AC	۰/۵۸۴ (۰/۰۱۲)	۰/۱۱۸ (۰/۰۰۱)	۱/۲۳ (۰/۰۱۱)	۰/۴۱۸ (۰/۱۲۱)	۰/۷۷۶ (۰/۰۱۸)	۶/۱۱ (۰/۰۲۴)
	A2C	۰/۶۴۵ (۰/۰۱۱)	۰/۱۲۴ (۰/۰۰۱)	۱/۴۴ (۰/۰۲۱)	۰/۴۱۸ (۰/۰۲۱)	۰/۷۷۶ (۰/۰۱۱)	۶/۰۸ (۰/۰۱۶)
	DQN	۰/۷۴۱ (۰/۱۱)	۰/۰۹۱ (۰/۰۰۱)	۱/۷۱ (۰/۰۱۱)	۰/۴۰۷ (۰/۰۱۲)	۰/۱۴۲ (۰/۰۳۱)	۵/۸۸ (۰/۰۰۴)
	الگوریتم عملگر-نقاد مزیت ناهمگن	۰/۷۷۶ (۰/۰۰۱)	۰/۰۸۹ (۰/۰۰۴)	۱/۷۴ (۰/۰۱۱)	۰/۳۸۳ (۰/۰۲۵)	۱/۳۱ (۰/۰۰۱)	۵/۳۲ (۰/۰۰۱)
BICS	AC	۰/۷۶۹ (۰/۰۴۱)	۰/۱۱۴ (۰/۰۴۴)	۱/۶۵ (۰/۰۵۳)	۰/۳۲۱ (۰/۰۱۱)	۱/۴۱ (۰/۰۱۴)	۶/۰۲ (۰/۰۰۱)
	A2C	۰/۷۷ (۰/۰۱۱)	۰/۱۱۱ (۰/۰۱۱)	۱/۷۱ (۰/۰۱۸)	۰/۳۲۴ (۰/۰۴۱)	۱/۳۶ (۰/۰۱۳)	۵/۶۶ (۰/۰۰۶)
	DQN	۰/۷۷۷ (۰/۰۱۲)	۰/۱۱ (۰/۰۰۱)	۱/۷۷ (۰/۰۰۲)	۰/۳۱۸ (۰/۰۱۷)	۱/۳۲ (۰/۰۰۳)	۵/۵۱ (۰/۰۰۷)
	الگوریتم عملگر-نقاد مزیت ناهمگن	۰/۸۰۵ (۰/۰۲۱)	۰/۱۰۴ (۰/۱۰۶)	۱/۸۱ (۰/۰۴۶)	۰/۳۰۷ (۰/۰۱۲)	۱/۲۴ (۰/۰۰۱)	۵/۴۷ (۰/۰۰۴)



شکل ۶: عملکرد روش پیشنهادی در مقایسه با الگوریتم‌های موجود

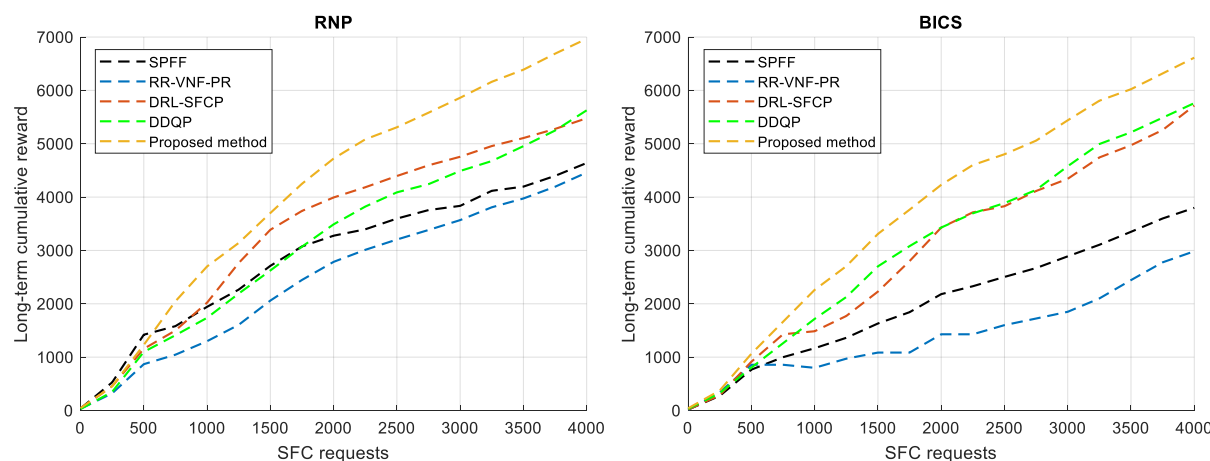
Figure 6. Performance of the proposed method compared to existing algorithms

همان‌طور که نشان داده شده است، روش پیشنهادی در مقایسه با SPFF و گرد کردن تصادفی برتری قابل توجهی دارد. روش SPFF و گرد کردن تصادفی فرایند مکان‌یابی زنجیره تابع سرویس را به صورت سریال انجام می‌دهند. اگرچه هر دو باهدف ارضا تأخیر ترافیک را هدایت می‌کنند، اما زنجیره تابع سرویس را با تعداد پرش بیشتری مونتاز می‌کنند. افزایش طول زنجیره تابع سرویس منجر به افزایش تأخیر و کاهش توان عملیاتی در SPFF و گرد کردن تصادفی شده است. اگرچه الگوریتم‌های یادگیری تقویتی عمیق کوتاهترین مسیر و DDQP موازی‌سازی زنجیره تابع سرویس را برای کاهش تأخیر و پهنای باند در نظر می‌گیرند، اما توابع شبکه مجازی‌های جدید بیشتری در فرایند مکان‌یابی مستقر می‌کنند و مصرف منابع بیشتری دارند. در همین حال،

روش پیشنهادی علاوه بر تبدیل زنجیره تابع سرویس‌های سریال به زنجیره تابع سرویس‌های موازی قابلیت استفاده مجدد از نمونه‌های توابع شبکه مجازی مقداردهی شده را دارد. از این رو، روش پیشنهادی هزینه مصرف منابع کمتری در مقایسه با یادگیری تقویتی عمیق کوتاهترین مسیر و DDQP فراهم می‌کند.

هنگامی که مصرف منابع برای مونتاژ زنجیره تابع سرویس‌ها کاهش یابد، درخواست‌های بیشتری در آینده می‌توانند پردازش شوند. به همین دلیل است که روش پیشنهادی نرخ استقرار بهتری در مقایسه با یادگیری تقویتی عمیق کوتاهترین مسیر و DDQP دارد. الگوریتم‌های یادگیری تقویتی عمیق کوتاهترین مسیر و DDQP از نرخ استقرار کمتری برخوردار هستند، زیرا برای درخواست‌های آینده منابع کمتری حفظ می‌کنند. نرخ استقرار کمتر به کاهش کیفیت و در نتیجه نقض الزامات کیفیت سرویس منجر می‌شود. بر اساس نتایج، نقض الزامات کیفیت سرویس در روش پیشنهادی در مقایسه با یادگیری تقویتی عمیق کوتاهترین مسیر و DDQP کمتر است. علاوه بر این، نقض الزامات کیفیت سرویس به توپولوژی شبکه وابسته نیست، زیرا تعداد نقض الزامات کیفیت سرویس در هر دو توپولوژی تقریباً یکسان است. اگرچه هدف اصلی روش پیشنهادی کاهش تأخیر زنجیره تابع سرویس است، اما الگوریتم ما با اجتناب از افزایش هزینه منابع و حداکثر استفاده از منابع موفق به کاهش نقض الزامات کیفیت سرویس در طول فرایند مکان‌یابی شده است.

مسئله مکان‌یابی زنجیره تابع سرویس یک مسئله تصمیم‌گیری است که هر تصمیم در مورد استقرار منجر به توزیع متفاوت توابع شبکه مجازی‌ها برای درخواست‌های آینده می‌شود؛ بنابراین، تجزیه و تحلیل پاداش تجمعی بلندمدت در رویکردهای مبتنی بر یادگیری تقویتی عمیق اهمیت دارد. در آزمایش دیگری، پاداش تجمعی بلندمدت در الگوریتم‌های مختلف برای مقایسه کارایی فرایند یادگیری ارزیابی می‌شود. نتایج این مقایسه برای ۴۰۰۰ درخواست زنجیره تابع سرویس متوالی در شکل ۷ نشان داده شده است. بر اساس نتایج، روش پیشنهادی بهترین عملکرد را در حداکثرسازی پاداش تجمعی بلندمدت دارد و یادگیری تقویتی عمیق کوتاهترین مسیر دومین بهترین الگوریتم است. بر اساس نتایج به دست آمده از توپولوژی حلقوی، روش پیشنهادی پاداش ارائه‌دهندگان سرویس را به طور متوسط ۲۳/۸٪ نسبت به یادگیری تقویتی عمیق کوتاهترین مسیر بهبود می‌دهد. این بهبود برای توپولوژی خطی به طور متوسط ۱۴/۷٪ است.



شکل ۷: ارزیابی الگوریتم‌های مختلف بر اساس حداکثرسازی پاداش تجمعی بلندمدت  
Figure 7. Evaluation of different algorithms based on maximizing the long-term cumulative reward

در همین حال، گرد کردن تصادفی کمترین پاداش تجمعی بلندمدت را در مقایسه با سایر الگوریتم‌ها روی هر دو توپولوژی فراهم می‌کند. پاداش تجمعی بلندمدت به دست آمده از گرد کردن تصادفی در توپولوژی‌های حلقوی و خطی به ترتیب ۴۴۶۱ و ۲۹۸۴ است. از طرف دیگر، عملکرد SPFF نیز مشابه گرد کردن تصادفی است و برای توپولوژی‌های RNP و BICS به ترتیب ۴۶۳۹ و ۳۷۹۸ گزارش شده است. اساساً، حداکثرسازی پاداش تجمعی بلندمدت به عنوان هدف اصلی توسط SPFF و گرد کردن تصادفی در نظر گرفته نمی‌شود و در نتیجه این الگوریتم‌ها پاداش تجمعی بلندمدت کمتری در مقایسه با روش پیشنهادی ارائه می‌دهند.

علاوه بر این، روش پیشنهادی حدود ۲/۲۷٪ از DDQP ارزش پاداش تجمعی بلندمدت بیشتری روی توپولوژی RNP ارائه می‌کند. این برتری برای توپولوژی BICS برابر ۷/۱۵٪ است.

## ۶- نتیجه‌گیری

پارادایم مجازی سازی توابع شبکه با استفاده از مجازی‌سازی جعبه‌های میانی سخت‌افزاری به‌عنوان نمونه‌های توابع شبکه مجازی می‌تواند استقرار سرویس‌ها را با انعطاف‌پذیری بیشتری فراهم کند. هنگام درخواست سرویس جدید، یک زنجیره تابع سرویس به‌عنوان توالی مرتب‌شده از توابع شبکه مجازی‌ها برای ارائه سرویس تعریف می‌شود. در این مقاله، مسئله مکان‌یابی پویا زنجیره تابع سرویس‌ها در شبکه‌های مبتنی بر محاسبات ابر-مه با استفاده مجدد از نمونه‌های توابع شبکه مجازی و موازی‌سازی زنجیره تابع سرویس بررسی شد. ما یک مدل برنامه‌ریزی پویا چندهدفه را برای به حداقل رساندن تأخیر و ایجاد مصالحه بین هزینه سرویس و کیفیت سرویس پیشنهاد دادیم که از یادگیری تقویتی عمیق برای حل مسئله استفاده می‌کند. روش پیشنهادی با ایجاد یک شبکه صف زنجیره تابع سرویس مبتنی بر حافظه کوتاه-بلند مدت و استفاده از یادگیری تقویتی عمیق سعی در حداکثرسازی پاداش تجمعی بلندمدت با در نظر گرفتن هزینه تأخیر، هزینه منابع و توان عملیاتی دارد، جاییکه با توزیع پویای توابع شبکه مجازی‌های موردنیاز در طول زمان موفق به صرفه‌جویی در منابع و پذیرش تعداد درخواست‌های بیشتری شده است. ما کارایی روش پیشنهادی را با استفاده از شبیه‌سازی‌های گسترده و مقایسه با الگوریتم‌های پیشرفته نظیر یادگیری تقویتی عمیق کوتاهترین مسیر و DDQP ارزیابی کردیم. از نظر نرخ استقرار، تأخیر، توان عملیاتی، هزینه مصرف منابع، نقض الزامات کیفیت سرویس و طول زنجیره تابع سرویس، روش پیشنهادی در مقایسه با سایر الگوریتم‌ها برتری قابل توجهی دارد و پاداش تجمعی بلندمدت را بهبود می‌دهد. پیشنهاد می‌شود برای کارهای آینده، ما برای مدل‌سازی مسئله به‌صورت برنامه‌نویسی عدد صحیح غیر محدب تلاش شود.

## مراجع

- [1] W. Li, H. Wang, X. Zhang, D. Li, L. Yan, Q. Fan and R. Yao, "Security Service Function Chain Based on Graph Neural Network," *Information*, vol. 13, no. 2, p. 78, 2022, doi: 10.3390/info13020078.
- [2] K. Kaur, V. Mangat and K. Kumar, "A comprehensive survey of service function chain provisioning approaches in SDN and NFV architecture," *Computer Science Review*, vol. 38, p. 100298, 2020, doi: 10.1016/j.cosrev.2020.100298.
- [3] M. Mechtri, C. Ghribi and D. Zeglache, "A Scalable Algorithm for the Placement of Service Function Chains," in *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 533-546, Sept. 2016, doi: 10.1109/TNSM.2016.2598068.
- [4] A. Tomassilli, F. Giroire, N. Huin and S. Pérennes, "Provably Efficient Algorithms for Placement of Service Function Chains with Ordering Constraints," *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, Honolulu, HI, USA, 2018*, pp. 774-782, doi: 10.1109/INFOCOM.2018.8486275.
- [5] S. Zhang, W. Jia, Z. Tang, J. Lou and W. Zhao, "Efficient instance reuse approach for service function chain placement in mobile edge computing," *Computer Networks*, vol. 211, p. 109010, 2022, doi: 10.1016/j.comnet.2022.109010.
- [6] K. Yang, H. Zhang and P. Hong, "Energy-Aware Service Function Placement for Service Function Chaining in Data Centers," *IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1-6, doi: 10.1109/GLOCOM.2016.7841805.
- [7] S. Xie, J. Ma and J. Zhao, "FlexChain: Bridging Parallelism and Placement for Service Function Chains," in *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 195-208, March 2021, doi: 10.1109/TNSM.2020.3047834.
- [8] L. Gupta, M. Samaka, R. Jain, A. Erbad, D. Bhamare and C. Metz, "COLAP: A predictive framework for

- service function chain placement in a multi-cloud environment," *IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, 2017, pp. 1-9, doi: 10.1109/CCWC.2017.7868377.
- [9] H. Ko, D. Suh, H. Baek, S. Pack and J. Kwak, "Optimal placement of service function in service function chaining," *Eighth International Conference on Ubiquitous and Future Networks (ICUFN)*, Vienna, Austria, 2016, pp. 102-105, doi: 10.1109/ICUFN.2016.7536993.
- [10] Z. Allybokus, N. Perrot, J. Leguay, L. Maggi and E. Gourdin, "Virtual function placement for service chaining with partial orders and anti-affinity rules," *Networks*, vol. 71, no. 2, pp. 97-106, 2018, doi: 10.1002/net.21768.
- [11] Y. Xiao et al., "NFVdeep: Adaptive Online Service Function Chain Deployment with Deep Reinforcement Learning," *IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*, Phoenix, AZ, USA, 2019, pp. 1-10, doi: 10.1145/3326285.3329056.
- [12] J. Pei, P. Hong, K. Xue and D. Li, "Efficiently Embedding Service Function Chains with Dynamic Virtual Network Function Placement in Geo-Distributed Cloud System," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 10, pp. 2179-2192, 1 Oct. 2019, doi: 10.1109/TPDS.2018.2880992.
- [13] R. S. Ponmagal, S. Karthick, B. Dhiyanesh, S. Balakrishnan and K. Venkatachalam, "Optimized virtual network function provisioning technique for mobile edge cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 6, pp. 5807-5815, 2021, doi: 10.1007/s12652-020-02122-8.
- [14] K. S. Ghai, S. Choudhury and A. Yassine, "Efficient algorithms to minimize the end-to-end latency of edge network function virtualization," *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 10, pp. 3963-3974, 2020, .
- [15] S. Yang, F. Li, S. Trajanovski, X. Chen, Y. Wang and X. Fu, "Delay-Aware Virtual Network Function Placement and Routing in Edge Clouds," in *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 445-459, 1 Feb. 2021, doi: 10.1109/TMC.2019.2942306.
- [16] T. Subramanya, D. Harutyunyan and R. Riggio, "Machine learning-driven service function chain placement and scaling in MEC-enabled 5G networks," *Computer Networks*, vol. 166, p. 106980, 2020, doi: 10.1016/j.comnet.2019.106980.
- [17] H. Chen, X. Wang, Y. Zhao, T. Song, Y. Wang, S. Xu and L. Li, "MOSC: A method to assign the outsourcing of service function chain across multiple clouds," *Computer Networks*, vol. 133, pp. 166-182, 2018, doi: 10.1016/j.comnet.2018.01.020.
- [18] D. Bhamare, A. Erbad, R. Jain, M. Zolanvari and M. Samaka, "Efficient virtual network function placement strategies for cloud radio access networks," *Computer Communications*, vol. 127, pp. 50-60, 2018, doi: 10.1016/j.comcom.2018.05.004.
- [19] Y. Xie, S. Wang and Y. Dai, "Revenue-maximizing virtualized network function chain placement in dynamic environment," *Future Generation Computer Systems*, vol. 108, pp. 650-661, 2020, doi: 10.1016/j.future.2020.03.011.
- [20] R. Kouah, A. Alleg, A. Laraba and T. Ahmed, "Energy-Aware Placement for IoT-Service Function Chain," *IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Barcelona, Spain, 2018, pp. 1-7, doi: 10.1109/CAMAD.2018.8515003.
- [21] Y. Liu, H. Lu, X. Li, Y. Zhang, L. Xi and D. Zhao, "Dynamic Service Function Chain Orchestration for NFV/MEC-Enabled IoT Networks: A Deep Reinforcement Learning Approach," in *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7450-7465, 1 May1, 2021, doi: 10.1109/JIOT.2020.3038793.



- [22] L. Gu, D. Zeng, W. Li, S. Guo, A. Y. Zomaya and H. Jin, "Intelligent VNF Orchestration and Flow Scheduling via Model-Assisted Deep Reinforcement Learning," in *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 279-291, Feb. 2020, doi: 10.1109/JSAC.2019.2959182.
- [23] G. Li, B. Feng, H. Zhou, Y. Zhang, K. Sood and S. Yu, "Adaptive service function chaining mappings in 5G using deep Q-learning," *Computer Communications*, vol. 152, pp. 305-315, 2020, doi: 10.1016/j.comcom.2020.01.035.
- [24] R. Solozabal, J. Ceberio, A. Sanchoyerto, L. Zabala, B. Blanco and F. Liberal, "Virtual Network Function Placement Optimization With Deep Reinforcement Learning," in *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 292-303, Feb. 2020, doi: 10.1109/JSAC.2019.2959183.
- [25] J. Zheng et al., "Optimizing NFV Chain Deployment in Software-Defined Cellular Core," in *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 248-262, Feb. 2020, doi: 10.1109/JSAC.2019.2959180.
- [26] J. Sun, G. Huang, G. Sun, H. Yu, A. K. Sangaiah and V. Chang, "A Q-learning-based approach for deploying dynamic service function chains," *Symmetry*, vol. 10, no. 11, p. 646, doi: 10.3390/sym10110646.
- [27] T. Wang et al., "DRL-SFCP: Adaptive Service Function Chains Placement with Deep Reinforcement Learning," *ICC 2021 - IEEE International Conference on Communications*, Montreal, QC, Canada, 2021, pp. 1-6, doi: 10.1109/ICC42927.2021.9500964.
- [28] L. Wang, W. Mao, J. Zhao and Y. Xu, "DDQP: A Double Deep Q-Learning Approach to Online Fault-Tolerant SFC Placement," in *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 118-132, March 2021, doi: 10.1109/TNSM.2021.3049298.
- [29] G. L. Santos, D. D. F. Bezerra, E. D. S. Rocha, L. Ferreira, A. L. C. Moreira, G. E. Gonçalves and P. T. Endo, "Service function chain placement in distributed scenarios: a systematic review," *Journal of Network and Systems Management*, vol. 30, no. 4, pp. 1-39, 2022, doi: 10.1007/s10922-021-09626-4.
- [30] J. Cai, Z. Huang, J. Luo, Y. Liu, H. Zhao and L. Liao, "Composing and deploying parallelized service function chains," *Journal of Network and Computer Applications*, vol. 163, p. 102637, 2020, doi: 10.1016/j.jnca.2020.102637.
- [31] M. Zeng, W. Fang and Z. Zhu, "Orchestrating Tree-Type VNF Forwarding Graphs in Inter-DC Elastic Optical Networks," in *Journal of Lightwave Technology*, vol. 34, no. 14, pp. 3330-3341, July 2016, doi: 10.1109/JLT.2016.2565002.

---

#### COPYRIGHTS

©2023 by the authors. Published by the Islamic Azad University Bushehr Branch. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution 4.0 International (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0>

---

