DRL-SFCP: Adaptive Service Function Chains Placement with Deep Reinforcement Learning

Tianfu Wang, Qilin Fan*, Xiuhua Li, Xu Zhang, Qingyu Xiong, Shu Fu and Min Gao

Presenter

Chongqing University

Nanjing University
Introduction

Traditional Network Architecture

User -> App -> ISP

Difficulties:
- Management
- Deployment
- Consumption
- Programmability
Introduction

**Network Function Virtualization**

- **User** ➔ **App** ➔ **ISP**

**Efficiency**
- Low-cost
- High-security
- Resource utilization

**Flexibility**
- Schedulable
- Programmability
- Management difficulty
Introduction

NFV Market

BY REGION (USD, BILLION)

Major Factors

the growing need for advanced network management systems to handle the increasing network traffic and complexities.

*Image from MarketsandMarkets Research
Introduction

Service Function Chain Request

A series of ordered VNFs (Virtual Network Functions)

VNF1 10 20 5 → VNF2 15 20 30 → VNF3 20 30 10 → VNF4 10 20 20

- CPU resource
- RAM resource
- storage resource
- bandwidth resource
Introduction

Service Function Chain Request

VNF1

VNF2 20

VNF3

VNF4

Host1

Host3

Host5

Host4

Physical Network

SFC Placement Problem

NP-hard

Constricted CO

CPU resource

RAM resource

storage resource

bandwidth resource
Introduction

Existing Solutions

A Mathematical optimization-based

- Integer Linear Programming
- Binary Integer Programming

Require the prior knowledge of SFCs

B (Meta) Heuristic-based

- Global Resource Control
- Node Rank based on degree
- Constructive Particle Swarm

Fall into the local optimum and static scene

C Reinforcement learning-based

- Q learning-based
- Dynamic programming-based
- Policy gradient-based

Large search space & manually selected features
Formulation

System

- Physical network: A weighted undirected graph $G' = (N', L')$
- SFC request: A weighted directed graph $G^i = (N^i, L^i)$

Constraints

- Node resources constraint:
  \[ \sum_{n'} \phi_{n', n^i, k}^{n^i} \leq R_{n', k}^{r}, \forall n' \in N', \forall k \in K \]
  \[ \sum_{n'} \phi_{n', n^i}^{n^i} \leq 1, \forall n^i \in N^i \] (Deployment constraint)

- Link bandwidth constraint:
  \[ \sum_{l'} \phi_{l', n^i}^{l^i} b_{l'} \leq B_{l'}^{r}, \forall l' \in L' \]
  \[ \sum_{l'' \in I(n')} \phi_{l''}^{l''} - \sum_{l' \in O(n')} \phi_{l'}^{l'} = \phi_{n', l'}^{n^i} - \phi_{n^i, l'}^{n^i}, \forall l', l'' \in L' \] (Path constraint)

Objective

Maximize the long-term average revenue

\[ \text{Max} \quad R(\pi) = \lim_{\tau \to \infty} \frac{1}{\tau} \sum_{i \in I_\tau} \text{rev}(i) \]

Single SFC request revenue

\[ \text{rev}(i) = \begin{cases} 
\mu_k \sum_{n^i} r_{n^i, k} + \eta \sum_{l^i} b_{l^i}, & \text{if } i \text{ is accepted,} \\
0, & \text{otherwise,} 
\end{cases} \]
Basic RL

**Agent** make placement decisions with Neural Network

**Env.** Physical Network
Serialized SFC requests

**State** Current situation of PN
Demand of underway SFC

**Action** One of physical nodes to accommodate the VNF

**Reward** Returned to award or punish the agent's behavior
Env

Virtual Network

Physical Network

State

Max Node Resource
Available Node Resource
Max Sum(Bandwidth)
Available Sum(Bandwidth)

Physical Network

Virtual Network

Agent

Neural Network

Action Probs

Node Request
Bandwidth Request
Number of Pending Nodes
Agent (Neural Network Architecture)

Current Physical Network State

\[
\begin{align*}
\mathbf{s}_t^p & \xrightarrow{\text{GCN}} \mathbf{Z}_t \\
\end{align*}
\]

\[
\begin{align*}
\alpha_t & \rightarrow \mathbf{c}_t \\
\end{align*}
\]

Alignment Vector

Context Vector

\[
\begin{align*}
\mathbf{e}_1 & \rightarrow \mathbf{e}_t \\
\mathbf{e}_t & \rightarrow \mathbf{e}_T \\
\end{align*}
\]

\[
\begin{align*}
\mathbf{s}_1^r & \rightarrow \mathbf{s}_t^r \\
\mathbf{s}_t^r & \rightarrow \mathbf{s}_T^r \\
\end{align*}
\]

\[
\begin{align*}
\mathbf{a}_1 & \rightarrow \mathbf{a}_t \\
\mathbf{a}_t & \rightarrow \mathbf{a}_T \\
\end{align*}
\]

Action Distribution

Key Components

Embedding of physical network

Embedding of SFC request

Policy generation
Embedding of Physical Network with GCN

**Graph** = $G(X, A)$

- $X$: Nodes Feature
- $A$: Adjacency matrix

$Z = \sigma(\widetilde{D}^{-1/2} \widetilde{A} \widetilde{D}^{-1/2} X W)$

$\widetilde{A} = A + I_N$

**Input**

- **Node Features**
  - $h = \{ \rightarrow h_1, \rightarrow h_2, ..., \rightarrow h_N \}$, $h_i \in \mathbb{R}^F$

**Output**

- **Node Representation**
  - $h' = \{ \rightarrow h'_1, \rightarrow h'_2, ..., \rightarrow h'_N \}$, $h'_i \in \mathbb{R}^{F'}$

---

Thomas N. Kipf et al. “Semi-Supervised Classification with Graph Convolutional Networks”. In ICLR, 2017
A series of actions (physical nodes) to accommodate VNFs

Resource requests of a series of VNFs in a SFC

A series of actions (physical nodes) to accommodate VNFs

Ilya Sutskever et al. “Sequence to Sequence Learning with Neural Networks”. In NIPS, 2014
Model

Policy generation

Current Physical Network State

\[ s_t^p \]

\[ Z_t \]

GCN

\[ c_t \]

\[ a_1 \]

\[ a_t \]

\[ a_T \]

Action Distribution

Alignment Vector

\[ \alpha_t \]

\[ e_1 \]

\[ e_T \]

\[ d_1 \]

\[ d_T \]

Context Vector

\[ c_t \]

\[ e_t \]

\[ d_t \]

\[ d_{t-1} \]

\[ d_T \]

\[ s_1^r \]

\[ s_T^r \]

\[ a_{t-1} \]

\[ a_T \]

GRU

GRU

GRU

GRU

GRU

GRU

GRU

GRU

GRU
**Model**

### Reward Design

- **State**
- **Agent**
- **Action**
- **Reward**

**Reward** = \(\xi\) (SFC revenue) - \(\xi\) (SFC revenue) - \(\xi\) (SFC revenue)

- if \(t \neq T\) and **success** (VNF is placed)
- if \(t \neq T\) and **failure** (SFC is rejected)
- if \(t = T\) and **success** (SFC is accepted)

\(\xi\) reward coefficient
A3C (Asynchronous A2C)

- Accelerate the training rate
- Strengthen the model robustness

**Experimental Setup**

**Model Parameters**

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U$</td>
<td>4</td>
<td>the number of actor networks</td>
</tr>
<tr>
<td>$\mu_k$</td>
<td>0.001</td>
<td>the unit price of resource $k$</td>
</tr>
<tr>
<td>$\eta$</td>
<td>0.001</td>
<td>the unit price of bandwidth</td>
</tr>
<tr>
<td>$\varepsilon_\theta$</td>
<td>0.00025</td>
<td>the learning rate of actor</td>
</tr>
<tr>
<td>$\varepsilon_\omega$</td>
<td>0.0005</td>
<td>the learning rate of critic</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>0.95</td>
<td>the discount factor of TD error</td>
</tr>
<tr>
<td>$\xi$</td>
<td>0.125</td>
<td>the reward coefficient</td>
</tr>
<tr>
<td>$B$</td>
<td>64</td>
<td>the batch size</td>
</tr>
<tr>
<td>$U_{gcN}$, $U_{emN}$, $U_{enC}$, $U_{deC}$</td>
<td>64</td>
<td>the units number of GCN layer, embedding layer, encoder hidden states and decoder hidden states</td>
</tr>
</tbody>
</table>

**Physical Network**
- topology: About 500 Nodes and 200 Links
- Resources: Uniform distribution $[50, 100]$

**SFC Request**
- SFC lifetime: Exponential distribution with an average of 400
- SFC length*: Uniform distribution $[2, 15]$
- Average arriving rate*: 20 per 100 time units

**Compared Algorithms**
- GRC based on global resource capacity  
  (L. Gong et al. INFOCOM 2014)
- MCTS using Monte Carlo tree search  

* means these settings may vary in the experiments for diverse evaluation
Experiment

Acceptance Ratio & Average Revenue

Result

DRL-SFCP achieves greater effect on two indicators

Ascribe

the abundant information extracted from SFC and PN

![Graph showing acceptance ratio and average revenue over changes in number of SFC requests.](image-url)
The performance of DRL-SFCP outperforms GRC and MCTS in various arrival rate conditions. The excellent abilities of fitting and generalization of DNN are ascribed to the proposed method.
Experiment

Increase the SFC requests’ length

It demonstrates that DRL-SFCP is well-suited for Application on online scenarios
Our contribution

- Adaptive DRL framework  
  Guide online placement decision for SFC requests
- Effective NN architecture  
  Extract the sufficient information from input features
- Parallel Training Method  
  Enhance the training efficiency and model robustness

Future work

- More powerful NN architectures  
  i.e. GNN, Transformer
- More efficient DRL methods  
  i.e. Multi-agent
- More realistic modeling scenarios  
  i.e. latency, multi-flow
THANKS

DRL-SFCP: Adaptive Service Function Chains Placement with Deep Reinforcement Learning

Tianfu Wang, Qilin Fan*, Xiuhua Li, Xu Zhang, Qingyu Xiong, Shu Fu and Min Gao

wtfly2018@gmail.com

github.com/GeminiLight/drl-sfcp