FlagVNE: A Flexible and Generalizable Reinforcement Learning Framework for Network Resource Allocation

IJCAI
JEJU 2024

AI for Networking

Tianfu Wang, Qilin Fan, Chao Wang, Long Yang, Leilei Ding, Nicholas Jing Yuan, Hui Xiong

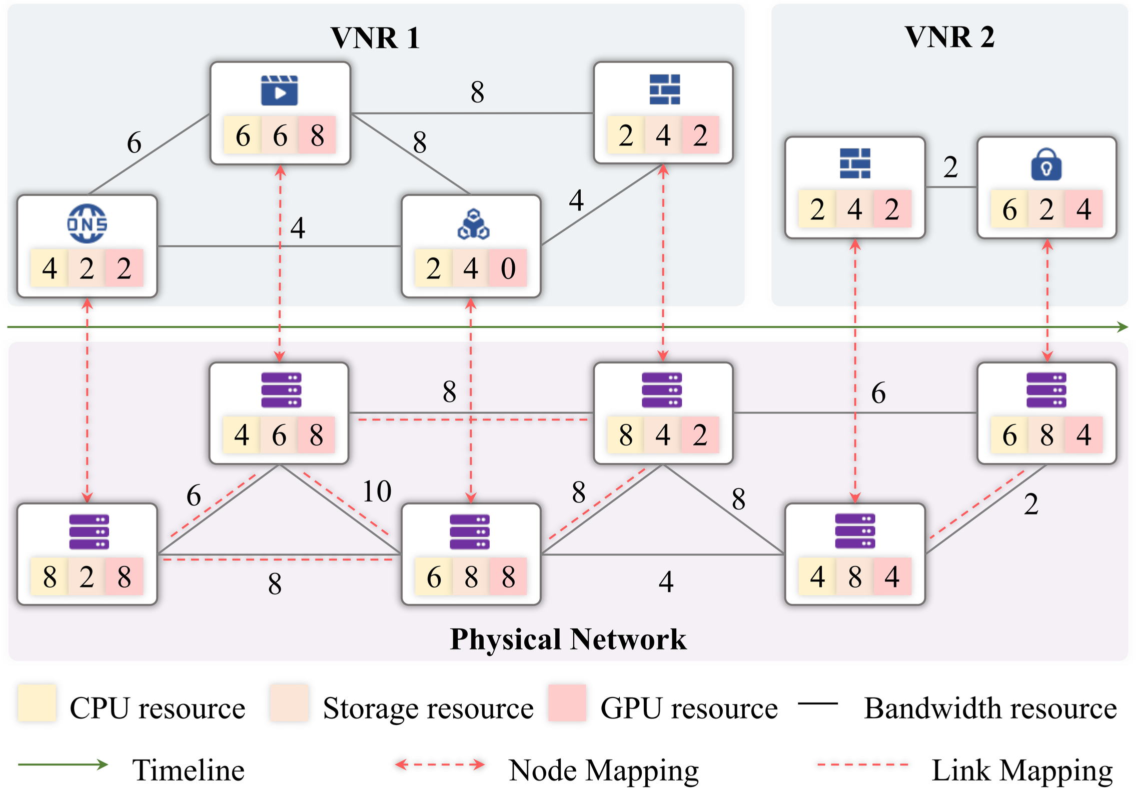
University of Science and Technology of China | Chongqing University | Peking University | Microsoft Inc. | The Hong Kong University of Science and Technology (Guangzhou)

Virtual Network Embedding Problem

A critical resource allocation task in network virtualization

- User network service → Virtual network requests (VNR)
- Underlaying infrastructure → Physical network

Maps VNRs to physical network while satisfying QoS constraints



Combinatorial Explosion	NP-hard Combinatorial Optimization Problem The solution space of VNE is extensive
Differentiated Demands	Specific requirements of user service are diverse Dynamic VNR topologies and dynamic demands

Motivations & Challenges Inspired by Preliminary Study

VNE Algorithms	Exact methods	Heuristics	RL-based Methods
	Expensive time consumption	Heavily rely on manual designs	Automatically build efficient solving policies

A. Flexibility of Action Space

Existing Methods	They employ a unidirectional action design, i.e., assuming that decision sequence of virtual nodes is predetermined
Preliminary Study	Figure 4 reveals that varying the decision sequences of virtual nodes significantly impacts performance
Intuitive Direction	Achieve a joint selection of both physical and virtual nodes to enhance the flexibility of exploration and exploitation
Latent Challenges	The difficulty of variable action prob distribution generation The training efficiency issue caused by large action space

B. Generalization of Solving Policy

Existing Methods	They typically use a one-size-fits-all policy to tackle VNRs of varying sizes, leading to generalization issues
Preliminary Study	Figure 5 reveals that some size-specific policies are superior or to single-policy, while some are inferior.
Intuitive Direction	Train a set of sub-policies directly to handle VNRs of different sizes from scratch
Latent Challenges	Specific policies trained from scratch encounter local optima How to quickly adapt to handle previously unseen VNR sizes

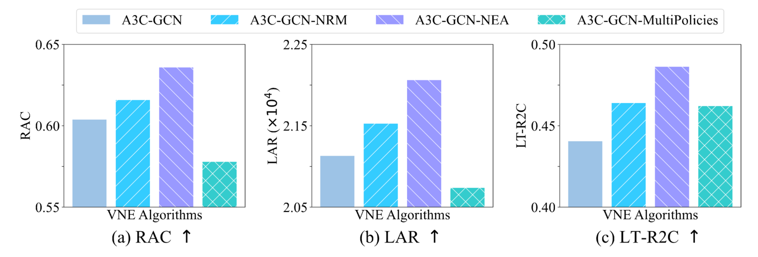


Figure 4: Comparative performance of A3C-GCN variants on three metrics: Impact of decision sequence and size-specific policies on VNE. (We conduct experiments using WX100 as the physical network, with a VNR arrival rate of 0.18. All other settings remained consistent with those described in Section 5.)

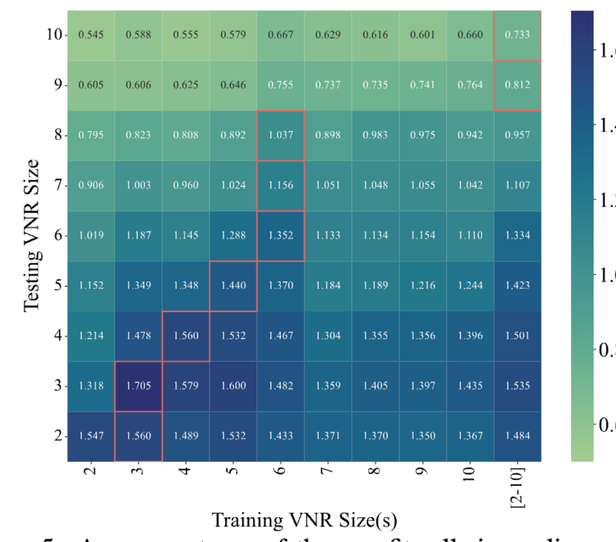


Figure 5: Average returns of the one-fits-all-size policy and each size-specific policy on all testing VNR sizes. The red boxes indicate the best performance results for test sizes. In the horizontal axis, [2-10] indicates a well-trained A3C-GCN policy while a single number represents a size-specific policy derived from well-trained A3C-GCN-MultiPolicy. (We use WX100 as the physical network and all training settings are the same as those mentioned in Section 5. For testing data of each VNR size, to exclude network system dynamics for a fairer comparison, we randomly generated 1000 static instances, including VNR and physical networks, as the benchmark. The performance metric is defined as the average episode return over 1000 instances.)

FlagVNE | A FLexible And Generalizable Reinforcement Learning Framework for Solving VNE Problem

A. Bidirectional Action-based MDP

Joint selection of virtual & physical nodes

Enhance the flexibility of agent exploration and exploitation

B. Hierarchical Policy Architecture

High-level ordering policy

Select the appropriate virtual node for the low-level placement.

Low-level placement policy

Identify a suitable physical node for placing to-be-placed virtual node

Distribution Size: $|N^v| \times |N^p| \rightarrow |N^v| + |N^p|$

Adaptively generate action prob dists and ensure high training efficiency.

C. Generalizable Training Method

Meta-RL for VNE

Formulate varying VNRs as distinct MDPs based their size.

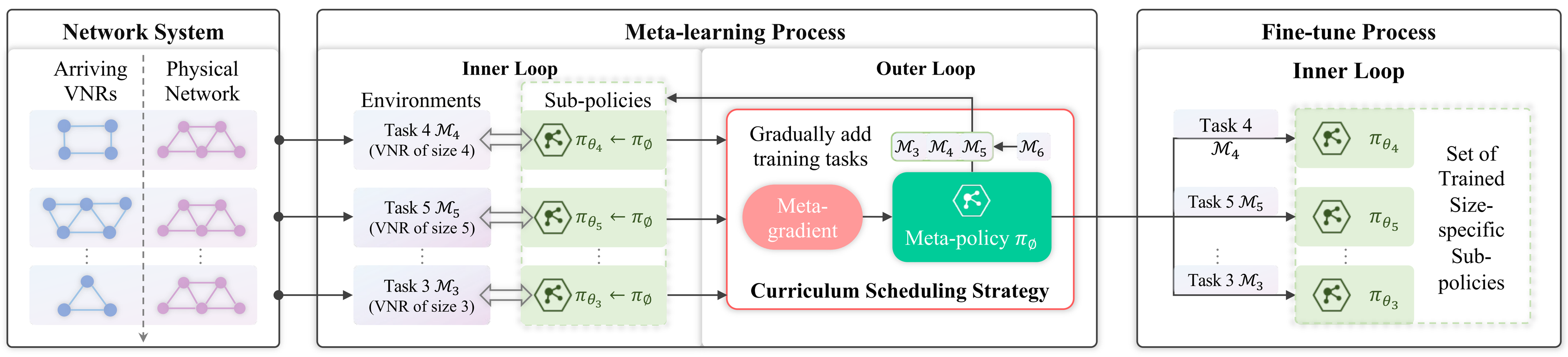
Adopt model-agnostic meta-learning (MAML) as the basic training method

Curriculum Scheduling Strategy

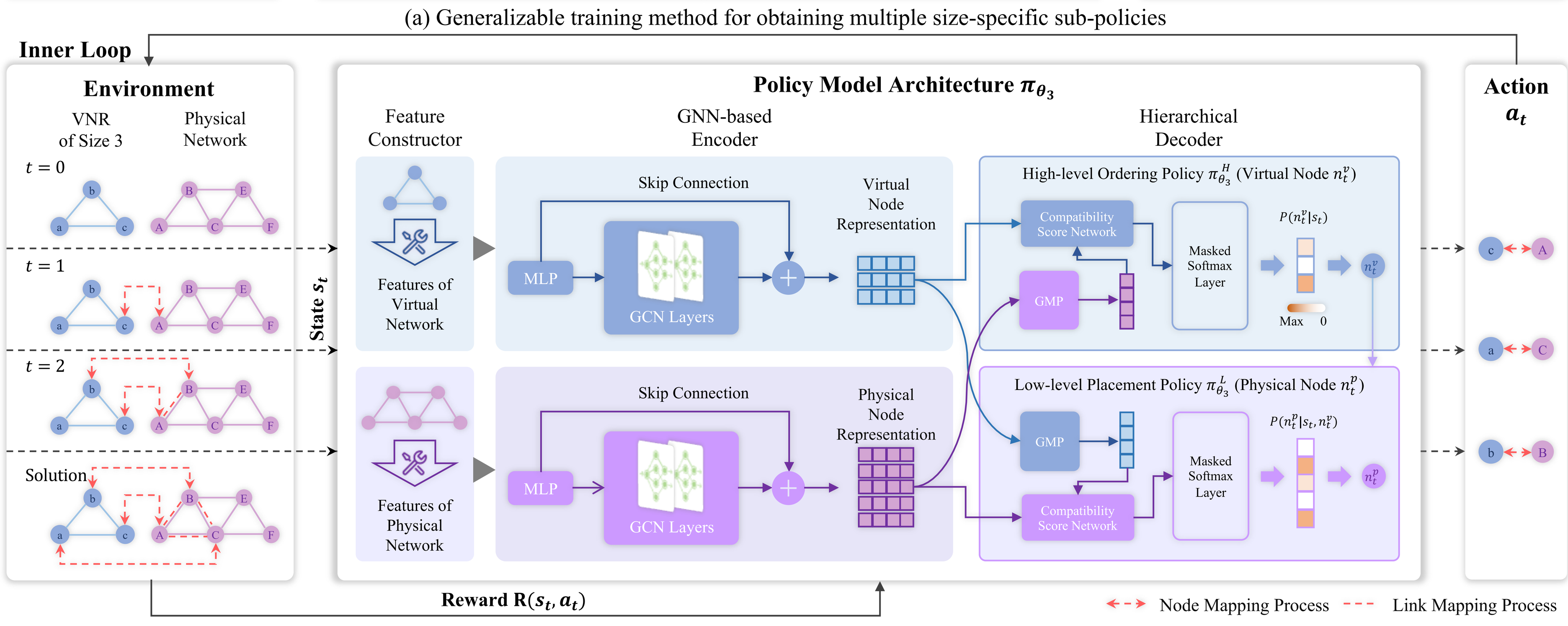
Training large-size policies from scratch adversely impacting meta-learning

Gradually increase the complexity of tasks during RL training

Effectively obtain refined solving policies for each VNR size



(a) Generalizable training method for obtaining multiple size-specific sub-policies



(b) Bidirectional action-based MDP modeling within each inner loop (sub-policy π_{θ_3} for example)

(a) Vary-sized VNRs arriving at the network system are considered as different tasks following $M_i \sim p(M)$

- First train a meta-policy π_φ in the meta-learning process with a curriculum scheduling strategy
- Then, Fine-tune it to obtain a set of size-specific sub-policies $\pi_{\theta i}$

(b) Within each inner loop, (use sub-policy $\pi_{\theta 3}$ for example)

- Formulate the solution construction process of each VNR as a bidirectional action-based MDP
- Design a hierarchical encoder with a bilevel policy to decide virtual and physical node

Theorem. *Given two MDPs with the bidirectional and unidirectional action, $\mathcal{M}^b = \langle \mathcal{S}^b, \mathcal{A}^b, P, R, \lambda \rangle$ and $\mathcal{M}^u = \langle \mathcal{S}^u, \mathcal{A}^u, P, R, \lambda \rangle$, and their optimal policy denote as $\pi^{\star,b}$ and $\pi^{\star,u}$, respectively, we have $\pi^{\star,b} \succeq \pi^{\star,u}$.*

Algorithm 1: Training Process of FlagVNE

```
Input  : Initial meta-policy φ; Policy set Θ = {φ};
         Meta learning rate β; Task learning rate α;
         Policy entropy threshold δ
Output : Trained policies set Θ;
// Meta-learning Process
Initialize the training task ID list I = {1};
while not done do
    Collect the trajectory memory D by interactions;
    Split D into {D_1, ···, D_|M|} based on VNR' size;
    Analyze the task distribution M_i ~ p(M);
    for i = 1, 2, ···, |M| do
        if i ∉ I then
            continue
        end
        θ_i ← DeepCopy(φ);
        Update θ_i with Eq. (9) and (10); // Inner loop
    end
    Update φ with Eq. (12); // Outer loop
    // Curriculum Scheduling Strategy
    Get the current most complex task ID k = max(I);
    if H(π_θk) < δ and k < |M| then
        I ← I ∪ {k + 1}
    end
end
// Fine-tuning Process
for i = 1, 2, ···, |M| do
    θ_i ← DeepCopy(φ);
end
while not done do
    Collect task trajectory memories {D_1, ···, D_|M|};
    for i = 1, 2, ···, |M| do
        Update θ_i with Eq. (9) and (10); // Inner loop
    end
end
for i = 1, 2, ···, |M| do
    Θ ← Θ ∪ {θ_i};
end
```

Performance Evaluation

Experiment Setup

Network topologies
GEANT & WX100

VNR Size Distribution
from 2 to 10

Each simulation run
Random 1000 VNRs

Performance Metrics
RAC & LAR & LT-R2C

Additional Evaluation

Running Time Test

Adaptation and Convergence Analysis

Scalability Validation

Hyperparameter Sensitivity

Overall Performance

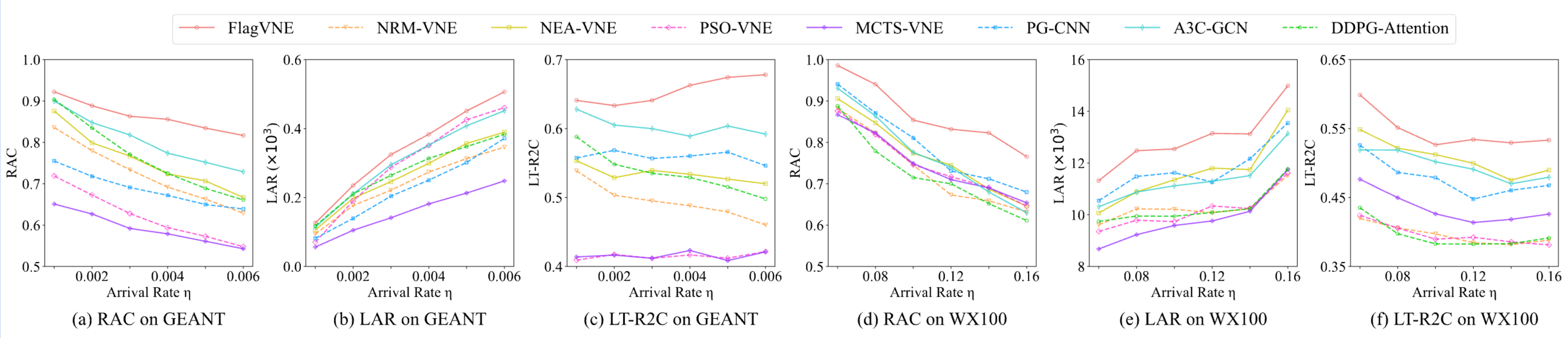


Figure 3: Experimental results in traffic throughput test.

Ablation Study

	GEANT			WX100		
	RAC ↑	LAR ↑	LT-R2C ↑	RAC ↑	LAR ↑	LT-R2C ↑
FlagVNE-UniActionNEA	0.781	475.335	0.637	0.724	14334.671	0.493
FlagVNE-MetaFree-SinglePolicy	0.758	472.455	0.614	0.712	14170.514	0.501
FlagVNE-MetaFree-MultiPolicy	0.746	435.502	0.593	0.685	14069.938	0.472
FlagVNE-MetaPolicy	0.773	478.646	0.634	0.717	14292.962	0.485
FlagVNE-NoCurriculum	0.787	485.267	0.643	0.708	14144.234	0.509
FlagVNE	**0.804**	**499.303**	**0.668**	**0.754**	**14769.080**	**0.526**

Table 1: Results on ablation study. ($\eta = 0.006$ on GEANT and $\eta = 0.18$ on WX100).

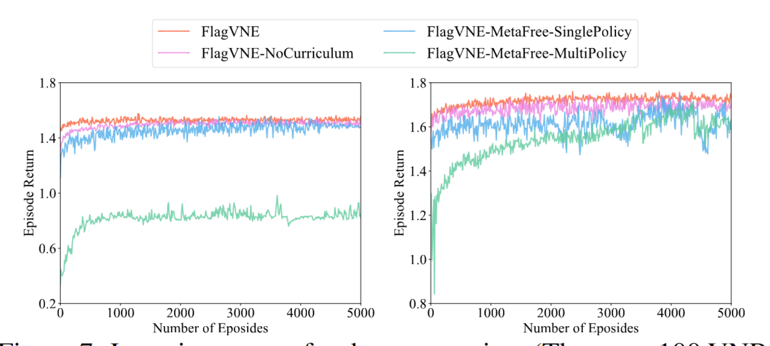


Figure 7: Learning curves for the unseen sizes. (There are 100 VNRs in one simulation and each VNR's size is set to 12).

Conclusion

Preliminary Study

Existing methods are limited by the unidirectional action design and one-size-fits-all training strategy

Result in restricted searchability and generalizability

FlagVNE Framework

A bidirectional action-based MDP model

- Jointly select of virtual and physical nodes
- Superior searchability and proven theoretically

A hierarchical decoder with a bilevel policy

- ensure adaptive action prob dist generation
- ensure high training efficiency

A meta RL-based training method

- efficient obtain multiple size-specific policies
- quick adaptation to new sizes

A curriculum scheduling strategy

- gradually incorporates larger VNRs
- alleviate suboptimal convergence

Extensive Experiments