On the Scaling of Reliable Interplanetary Networks with Deep Reinforcement Learning

(Invited Paper)

Xiaojian Tian, Xiaoliang Chen, Xixuan Zhou and Zuqing Zhu†
School of Information Science and Technology, University of Science and Technology of China, Hefei, China
†Email: {zqzhu}@ieee.org

***Abstract*—The rapid development of deep space exploration missions has led to the continuous expansion of interplanetary networks (IPNs) for enhanced data transfer capacities and reliability. In this paper, we propose a novel deep reinforcement learning (DRL) framework for optimizing the scaling of IPN topologies (*i.e.*, the placement of relay satellites), such that the routing and data scheduling of interplanetary data transfer (IP-DT) in the scaled IPN achieves maximized performance gain. Our proposal leverages graph neural networks (GNNs) to extract topological correlations within an IPN and hereby learns progressive local rewriting policies for approaching the optimal solution. Extensive simulations verify the effectiveness and robustness of our proposal, showing that the learned relay satellite placements facilitate higher reliability (in terms of data delivery ratios) and lower end-to-end latency for different IPN scenarios, when compared with the existing benchmarks.**

***Index Terms*—Interplanetary network, Delay tolerant network, Topology scaling, Deep reinforcement learning.**

I. Introduction

With the continuous increase of deep space (DS) exploration missions, interplanetary networks (IPNs) have been developed rapidly in recent years and gained much attention globally [1]. IPNs provide communication and data relay services among planetary bodies, satellites and spacecrafts [2], differentiating them broadly from the cases in terrestrial networks [3–14]. In particular, interplanetary data transfer (IP-DT) faces unique challenges in dynamic and unstable topologies caused by the movement and obstruction of network nodes and limited bandwidth and extremely long delays due to vast distances in DS communications. These challenges can be partially overcome by delay tolerant networking (DTN) [15, 16], which improves the reliability and efficiency of IP-DT over unreliable and ultra-long-latency links using store-carry-forward (SCF).

Current IPNs typically employ sparse network topologies or even point-to-point architectures [17]. However, the vigorous development of DS exploration missions [18–20] will demand universal and multi-hierarchy IPN architectures to support enlarged network dimensions and traffic volumes. Previous studies have reported inspiring progresses on optimizing the routing, data scheduling and rate control for IP-DT, but all assumed fixed topology configurations [21–24]. Note that, as IPNs scale up, bandwidth bottlenecking exacerbates and presents a major obstacle to improving the performance (capacity, latency, *etc*.) of IP-DT. This issue can hardly be worked out directly given the ultra-long link lengths and complex electromagnetic environment in the universe. More importantly, the lossy and disruptive links of IPNs make reliability a critical concern under fixed topology configurations. In this context, IPN topology scaling that leverages relay satellites to augment link capacities and reachability while reducing communication latencies (as shown in Fig. 1) emerges as a promising solution. Nevertheless, the dynamic nature of IPN necessitates joint optimization of the planning of relay satellites and the routing and data scheduling for IP-DT, taking into account the motions and relative positions of all the nodes. This makes IPN topology scaling very challenging.

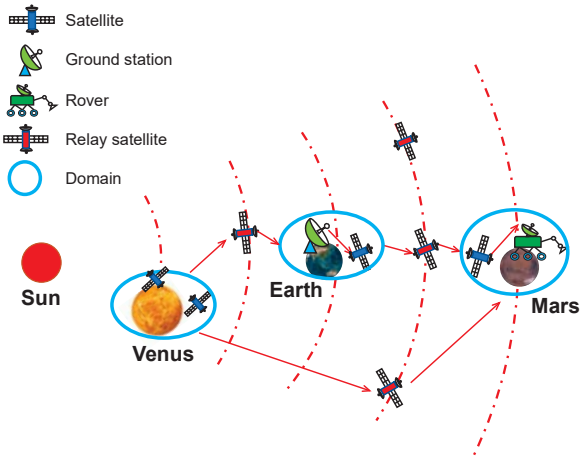


Fig. 1. Example on IPN topology scaling.

Existing studies on IPN topology scaling [25, 26] primarily highlighted the benefits of multi-hop DS communications while treating the planning of relay satellites as a static geometric optimization problem. They overlooked the interplay between relay satellite placement and routing and scheduling of IP-DT. Recently, the authors of [27] formulated a joint optimization of the planning of relay satellites and the routing and data scheduling for IP-DT, but their proposal relied on simple yet fixed heuristic policies to solve the complex nonlinear optimization, and thus could hardly find effective solutions in large-scale IPN settings. On the other hand, deep reinforcement learning (DRL) has been shown to achieve beyond human-level performance in many dynamic control tasks [28]. DRL takes advantage of the powerful representation ability of deep neural networks (DNNs) to learn successful policies through repeated trials and errors, hereby eliminating the need for deriving explicit mathematical expressions for tar-

get optimizations. This makes DRL also a promising technique for solving complex combinatorial optimizations [29–31].

In this paper, we develop a DRL framework for solving the optimization of IPN topology scaling in a progressive manner. Our DRL framework leverages graph neural networks (GNNs) to extract meaningful representations of IPN topologies from carefully-structured graph inputs. Aided by the GNNs, the proposed DRL agents learn the local rewriting (orbital parameter adjustment) policies that facilitate efficient search of the optimization space, and consequently, gradually approach the optimal placement of relay satellites such that the routing and data scheduling of IP-DT in the scaled IPN achieves maximized performance gain. More specifically, the major contributions of this work include:

- We propose, for the first time to the best of our knowledge, a DRL framework that can effectively solve the joint optimization of the planning of relay satellites and the routing and data scheduling for IP-DT.
- We model the state of a dynamic IPN as graph-structured data and utilize GNNs to learn the local rewriting policies that guide progressive approach to the optimal solutions.
- Extensive simulations under different settings of the original IPN topology and the budget for new relay satellites verify that our proposal can obtain the effective IPN scaling solutions, which consistently outperform existing benchmarks in terms of the delivery ratio and average end-to-end (E2E) latency of bundles.

II. Related Work

Routing and data scheduling for IP-DT is a fundamental problems in IPNs, which can be addressed either separately or jointly. Most representatively, the National Aeronautics and Space Administration (NASA) proposed and standardized the contact graph routing (CGR) algorithm [32] that calculates an SCF route for each bundle over the time-varying topology of an IPN based on scheduled communication contacts. The data scheduling problem for IP-DT was first studied in [21]. Aiming at reducing the queuing delay, the authors designed an algorithm that adjusts the transmission orders of queued bundles considering multiple attributes of them (*e.g.*, size and priority). In [22, 24], Tian *et al.* conducted an in-depth study on the joint optimization of distributed routing and data scheduling for IP-DT, and achieved noticeable performance improvement in terms of E2E latency and throughput under high traffic loads, while ensuring good scalability.

Existing studies on routing and data scheduling for IP-DT mostly assume fixed IPN topologies. However, the rapid development of DS missions, especially with the introduction of relay satellites, has led to expanding IPN topologies continuously [25]. Consequently, considering the IPN topology scaling that introduces multi-hop DS communications facilitated by new relay satellite deployment becomes imperative. The merits of IPN topology scaling have been embodied by several previous studies, such as commutating rings, minimal Earth rings, and elliptical transfer between planetary orbits [25]. A linear-circular commutating chain topology was proposed in [33] to improve the throughput of Earth-Mars communications. Two-hop relay schemes based on Sun-Earth L4/L5 Lagrange points were studied in [34, 35]. Later in [26], Wan *et al.* presented a Solar System satellite relay constellation network topology designed to boost the bandwidth between Earth and Mars. Nevertheless, all these studies unanimously treated the deployment of relay satellites as a static geometric problem, which simply optimizes the link lengths in expanded IPNs. Therefore, they failed to address the joint optimization of the relay satellite orbit parameters and the routing and data scheduling for IP-DT, which could plausibly compromise their performance in dynamic DS missions.

The recent study in [27] filled the aforementioned gap and for the first time, investigated the joint optimization of the planning of relay satellites and the routing and data scheduling for IP-DT, exploring comprehensive and dynamic topology evolution strategies to better adapt to the evolving demands of DS missions. However, the proposed mixed integer nonlinear programming model and heuristic algorithm either suffer from scalability issues or may produce suboptimal solutions when tackling large-scale problems. In this work, we opt for a DRL-based approach to overcome such limitations, as DRL has exhibited superior performance in many network planning and optimization tasks (*e.g.*, routing [36], planning [31], and reconfiguration [37, 38]). The application of DRL in IPN routing and scheduling has also been explored by [23]. Nonetheless, solving the joint optimization of relay satellite placement, orbital parameters, routing and data scheduling with DRL is never a trivial task, which necessitates careful remodeling of neural network structure, action space and reward function.

III. Problem Statement

We symbolize the time-varying topology of an IPN by $G^t(V, E^t)$, where V and E^t represent the sets of nodes and temporal links at time t, respectively. Each directed temporal link $e^t(u, v, t^s, t^e, r, \tau) \in E^t$ connects node u to node v, with $[t^s, t^e]$ being the contact duration, r being the data-rate, and τ being the transmission latency. An IPN topology is further divided into a set of domains $\mathcal{H} = \{\mathcal{H}_j\}$ based on celestial bodies (see Fig. 1). We augment inter-domain links by deploying relay satellites using circular orbits centered on the Sun. For economic considerations, the number of relay satellites is restricted to be K. Then, we can sketch an IPN topology by its backbone representation $\tilde{G}^t(\tilde{V}, V_R, \tilde{E}^t)$, where each domain is abstracted as a virtual node $\tilde{v} \in \tilde{V}$ and the edges (in $\tilde{E}^t$) signify the communication links between domains or between domains and relay satellites in V_R. We use a polar coordinate system with the Sun being the pole and the Earth-Sun ray at $t = 0$ being the polar axis. Thus, the location of a node v at time t can be expressed as $P(v, t) = (\rho(v, t), \theta(v, t))$, where $\rho(v, t)$ is its radius in astronomical units (AU, $\sim 1.496 \times 10^8$ kilometers) and $\theta(v, t)$ is the phase. We model the operation of an IPN as a discrete-time system, where each node configures its IP-DT scheme at the beginning of every time slot (TS) Δt. Ultimately, the topology scaling problem is stated as: *optimizing the placement of V_R so that the IPN performance*

in a macro level, i.e., the cumulative performance gains over a long trajectory of TS's (say, a few years), is maximized.

Variables:

- $P(v,0)=(\rho(v,0),\theta(v,0)),\forall v\in V_R$: initial positions of relay satellites.
- $x_{e^t}^{\tilde{u},\tilde{v}},\forall\tilde{u},\tilde{v}\in\tilde{V},e^t\in\tilde{E}^t$: Boolean variable for inter-domain routing, where $x_{e^t}^{\tilde{u},\tilde{v}}$ equals 1 if e^t is traversed by the routing path for $\tilde{u}\to\tilde{v}$ at TS t, and 0 otherwise.

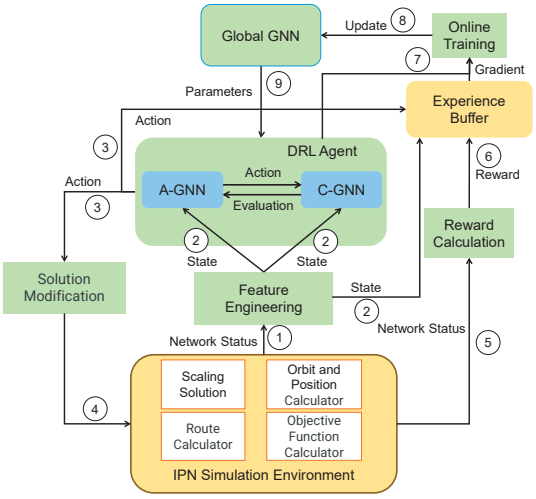


Fig. 2. Framework and process of our proposal.

Objective:

The optimization objective is to minimize the time-averaged total IP-DT latency among all the domains, *i.e.*,

$$\text{Minimize}\quad \mathcal{J}=\frac{1}{|\mathcal{T}|}\sum_{\{\tilde{u},\tilde{v}\in\tilde{V},\tilde{u}\neq\tilde{v}\}}\sum_{t\in\mathcal{T}}\mathcal{L}_t^{\tilde{u},\tilde{v}}, \tag{1}$$

where $\mathcal{T}=\{0,\Delta t,2\Delta t,\cdots,N\Delta t\}$ is the evaluation period and $\mathcal{L}_t^{\tilde{u},\tilde{v}}$ is the IP-DT latency between domains $\tilde{u}$ and $\tilde{v}$ at TS t, which is determined by the routing scheme (*i.e.*, $x_{e^t}^{\tilde{u},\tilde{v}}$) and the data rates of inter-domain links [27].

Specifically, $\mathcal{L}_t^{\tilde{u},\tilde{v}}$ consists of propagation delay and queuing delay. The propagation delay can be easily got by calculating the total path distance, according to the real-time positions of planets and relay satellites derived from their orbital parameters and the laws of mechanics. To obtain the queuing delay, we model the processing of bundles in an outgoing queue as a birth-death Markov chain, which can be characterized by an M/G/1 queuing system. As such, the average delay a bundle experiences in an outgoing queue is expressed as,

$$T_{\text{M/G/1}}=\left(\frac{\tau_l^2+\sigma_l^2}{2\cdot\tau_l}\right)\cdot\left(\frac{1}{r_t^{\tilde{u},\tilde{v}}-\lambda\cdot\tau_l}\right), \tag{2}$$

where τ_l and σ_l represent the mean and standard deviation of bundle sizes, respectively, and $r_t^{\tilde{u},\tilde{v}}$ is the data rate (*i.e.*, processing capacity) of the queue. $r_t^{\tilde{u},\tilde{v}}$ is determined by the data rate bottleneck of all the links along the path, *i.e.*,

$$r_t^{\tilde{u},\tilde{v}}\le x_{e^t}^{\tilde{u},\tilde{v}}\cdot r^{e^t},\ \forall e^t\in\tilde{E}^t. \tag{3}$$

Therein, r^{e^t} is the data rate of link e^t. It can be derived under the assumption of optimal channel coding in an additive white Gaussian noise (AWGN) channel model:

$$r^{e^t}=\alpha\cdot C^{e^t}=B\cdot\log_2\left[1+\epsilon\cdot\frac{1}{\left(D^{e^t}\right)^2}\right],\ \alpha\in(0,1), \tag{4}$$

where C^{e^t} is the channel capacity, B is the channel bandwidth, and ϵ denotes the ratio between the signal-to-noise ratio (SNR) and the reciprocal of the squared link distance, *i.e.*, $\frac{1}{\left(D^{e^t}\right)^2}$. SNR is related to the link's physical parameters, such as antenna gain and power. Finally, we obtain $\mathcal{L}_t^{\tilde{u},\tilde{v}}$ as,

$$\mathcal{L}_t^{\tilde{u},\tilde{v}}=T_{\text{M/G/1}}+\frac{D_t^{\tilde{u},\tilde{v}}}{c}, \tag{5}$$

where $D_t^{\tilde{u},\tilde{v}}$ is the total path length and c is the speed of light.

Overall, our formulation for IPN topology scaling harmonizes the orbital parameters (orbital radius and initial phase) of each relay satellite to improve the long-term IP-DT performance in terms of E2E latency, which implicitly translates into the optimization of data rate, bundle delivery ratio, and latency of inter-domain links in each TS t.

IV. Deep Reinforcement Learning Design

In this section, we elaborate on the DRL design for solving the aforementioned optimization.

A. Model Overview

Fig. 2 shows the schematic of the proposed design, featuring the major components of DRL agents and their interactions (marked by solid arrows with step numbers) with an IPN simulation environment for progressive learning of the optimal solutions. The environment implements the IPN network model described in the previous section and runs discrete-time simulations to evaluate IPN performance metrics based on the topology scaling solutions generated by the DRL agents. Note that, the solutions provided by the DRL agents decide only the initial positions of the relay satellites, while the simulator assesses the long-term performance over $\mathcal{T}$ by calculating time-varying node positions and routing schemes. The learning process starts with the DRL agents rewriting a random relay satellite placement $P(v,0)$. In particular, each agent reads IPN state data and outputs an action that deviates the current solution locally by ΔP (*Steps* 1-3). The simulation environment reevaluates the objective function with the updated solution (*i.e.*, $P(v,0)+\Delta P$), which is then translated into a numerical reward for the agent (*Steps* 4-6). The tuple of the state, action and reward is pushed into the experience buffer. *Steps* 1-6 are repeated for iterations until the termination condition is satisfied, allowing the agents to search the solution space adequately. Meanwhile, we perform periodical training with samples from the experience buffer to update the policies (parameterized by DNNs) of the agents (*Steps* 7-9).

B. DRL Agents

We design DRL agents based on the asynchronous advantage actor-critic (A3C) framework, which employs multiple agents interacting with independent environments in parallel for searching the solution space effectively. Fig. 3 shows the structure of an agent, which makes use of an actor network for policy generation and a critic network for evaluation purposes. The two networks adopt the same architecture for feature extraction but use different readout modules. Details about the DNN architectures will be provided later.

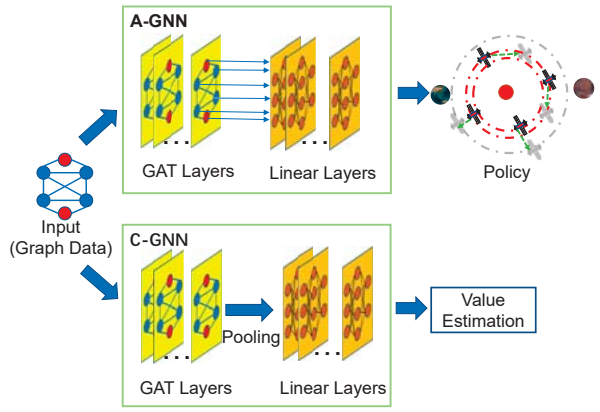


Fig. 3. Structure of a DRL agent.

State: the IPN node positions and link data rates are key information for the decision making of DRL agents. We encode these information as graph-structured data, where each graph instance $s_n \in \mathcal{S}$ is composed of nodes representing domains or relay satellites and edges corresponding to the communication links at the n-th iteration of optimization. Each node $v \in \tilde{V} \cup V_R$ is conveyed by a three-dimensional feature vector $[\rho(v,0), \theta(v,0), \kappa]$, indicating the node's initial orbital parameters and whether the node corresponds a domain ($\kappa = 0$) or a relay satellite ($\kappa = 1$).

Action: the agents take an action $a_n(v)$ to modify the orbital parameters of each relay satellite v. We reduce the action space by making the agents select one from five local rewriting strategies, namely, (*i*) increasing or (*ii*) decreasing $\rho(v,0)$ by fixed step size $\Delta\rho$, (*iii*) increasing or (*iv*) decreasing $\theta(v,0)$ by fixed step size $\Delta\theta$, and (*v*) staying unchanged.

Reward: the reward is defined as the performance gain (latency reduction) brought by the local rewriting action, *i.e.*,

$$r_n = \eta \cdot (\mathcal{J}_{n-1} - \mathcal{J}_n) + \zeta \cdot (M - \mathcal{J}_n), \tag{6}$$

where $\mathcal{J}_n$ denotes the latency at step n, M is a constant represents the average latency determined through historical data and priori analysis, and η and ζ are weight coefficients. Then, by maximizing the long-term cumulative reward, we guide the agents to move progressively to the optimal solution.

DNN Architectures: we parameterize the actor and critic networks by GNNs owing to their powerful capabilities in processing graph-structured data and generalizing arbitrary topologies. As shown by Fig. 3, the actor GNN (A-GNN) and critic GNN (C-GNN) both use several GNN layers to learn latent representations from graph inputs, which are then pooled (averaging) and fed to the linear layers for outputting a rewriting policy $\pi(s_n)$ for each node or a value estimation. $\pi(s_n)$ indicates a probability distribution over the strategies that guide action selection while the value estimation helps an agent evaluate the quality of action taken.

The GNNs are built on a message-passing mechanism, where each node aggregates the states of its neighbor nodes to calculate a more comprehensive node representation. This process is repeated for iterations (or layers), allowing the nodes to extend their perceptive fields and learn graph-level representations. The operations of a GNN layer include:

Algorithm 1: Training Procedure

Input: IPN topology G^t, and set of relay satellites V_R;
construct initial topology scaling solution $P(V_R, 0) = \{(\rho(v,0), \theta(v,0)) : \forall v \in V_R\}$;
set episode count $\mathcal{C} = 0$ and experience buffer $\Phi = \emptyset$;
while $\mathcal{C} < \mathcal{C}_{max}$ **do**
  generate graph instance s_n with G^t and $P(V_R, 0)$;
  calculate $\pi(s_n)$ with A-GNN;
  sample local rewriting actions $a_n(v)$ with $\pi(s_n)$;
  $P(v,0) \leftarrow P(v,0) + a_n(v), \forall v \in V_R$;
  compute r_n with Equation (6);
  $\Phi \leftarrow \{\Phi, \{s_n, a_n(v), r_n\}\}$;
  if $|\Phi| \geq 2N - 1$ **then**
    compute the losses for A-GNN and C-GNN with Equations (10) and (12) using $\{\Phi_i : i < N\}$;
    calculate gradients of the loss functions;
    apply gradients to the global GNNs;
    load the global GNN parameters to update A-GNN and C-GNN;
    $\Phi \leftarrow \{\Phi_i : i \geq N\}$, $\mathcal{C} \leftarrow \mathcal{C} + 1$;
  end
end

- *Message Passing*: each node disseminates its state (node representation) to its adjacent nodes.
- *Aggregation*: each node aggregates incoming messages.
- *Update*: each node updates its state based on the aggregated information.

The above operations can be mathematically expressed as,

$$\mathbf{h}_v^{(k+1)} = \sigma\left(\mathbf{W} \cdot \text{AGG}\left(\{\mathbf{h}_u^{(k)} : u \in \mathcal{N}(v)\}\right)\right), \tag{7}$$

where $\mathbf{h}_v^{(k)}$ is the state of node v in the k-th layer, $\mathcal{N}(v)$ contains the neighbors of node v, AGG denotes an aggregation function (*e.g.*, sum, mean, or max), $\mathbf{W}$ is a learnable weight matrix, and σ is a non-linear activation function (*e.g.*, ReLU) to introduce non-linearity and enhance model expressiveness.

To facilitate more effective state aggregation, we bring in the graph attention mechanism, *i.e.*, graph attention networks (GATs) [39]. GATs allow nodes to weight the importance of their neighbors and focus on more relevant information, *e.g.*, the states of closer neighbors, making them promising for handling combinatorial optimizations. In particular, GATs assign an attention coefficient $\alpha_{u,v}$ to each edge (u,v) as,

$$\alpha_{u,v} = \frac{\exp\left(\text{LeakyReLU}(\boldsymbol{\omega}^T[\mathbf{W}\mathbf{h}_u \| \mathbf{W}\mathbf{h}_v])\right)}{\sum_{v' \in \mathcal{N}(v)} \exp\left(\text{LeakyReLU}(\boldsymbol{\omega}^T[\mathbf{W}\mathbf{h}_v \| \mathbf{W}\mathbf{h}_{v'}])\right)}, \tag{8}$$

where $\|$ denotes the concatenation operation, $\boldsymbol{\omega}^T$ is the transpose of a learnable weight vector, and LeakyReLU is the Leaky ReLU activation function. Then, message passing with attention can be concretized as an activation function acting on the weighted summation of the states of a node's neighbors:

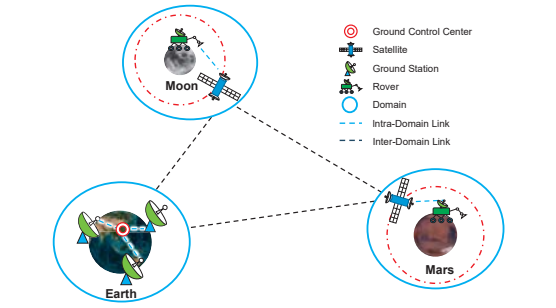
$$\mathbf{h}_v^{(k+1)} = \sigma\left(\sum_{u \in \mathcal{N}(v)} \alpha_{uv} \mathbf{W}\mathbf{h}_u^{(k)}\right). \tag{9}$$

By using the powerful representation capabilities of GNNs and attention mechanism of GATs, the model can potentially cap-

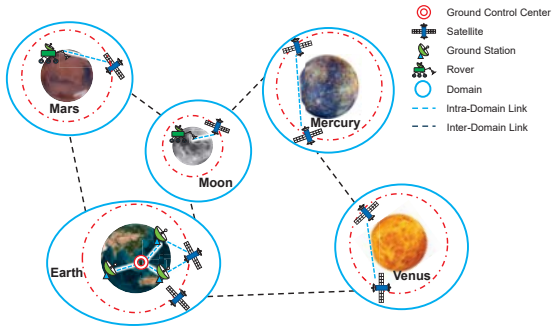
ture complex topological dependencies and dynamic changes in IPNs, and learn successful topology scaling policies.

C. Training Procedure

Algorithm 1 summarizes the training procedure of a DRL agent under the A3C framework, where multiple agents maintain a pair of global GNNs while each explores an independent environment in parallel. *Lines 2-3* are for initialization, where we construct an initial topology scaling solution by randomly placing a set of relay satellites V_R for each agent, and reset the episode count and experience buffer. Then, the while-loop covering *Lines 4-18* performs repeated trial and error of $\mathcal{C}_{\max}$ episodes. Specifically, *Lines 5-6* generate a graph instance s_n based on the IPN topology and current relay satellite placement and call A-GNN to output a local rewriting policy $\pi(s_n)$. Afterward, we sample a rewriting action $a_n(v)$ for each relay satellite v with $\pi(s_n)$ using the roulette method (*Line 7*) and update the topology scaling solution accordingly (*Line 8*). *Lines 9-10* calculate the instant reward r_n, which, together with the state observation and action taken, are stored in the experience buffer as a training sample. Once $2N-1$ samples are collected, we performs training with *Lines 12-14*.



(a) 3-domain topology (IPN-1)



(b) 5-domain topology (IPN-2)

Fig. 4. IPN topologies used in simulations.

In training, we set the goal of each agent at step n to be maximizing the total discounted reward from the next N continuous operations. Hence, we can calculate the target values for the first N samples in the buffer and use them as training samples. The policy loss is defined as the negative product of logarithmic action probability and temporal difference (TD) error averaged over all the relay satellites and samples, *i.e.*,

$$L_{\vartheta_a} = -\frac{1}{N \cdot |V_R|} \sum_{n=0}^{N-1} \sum_{v \in V_R} \delta_n \cdot \log\left(\pi_{\vartheta_a}(s_n, a_n(v))\right), \quad (10)$$

where $\pi_{\vartheta_a}(s_n, a_n(v))$ signifies the probability of selecting action $a_n(v)$, ϑ_a is the collection of A-GNN weights, and δ_n is the TD error (*a.k.a.* the advantage). δ_n is computed as

$$\delta_n = \sum_{i=n}^{n+N-1} \gamma^{i-n} r_i - \nu_{\vartheta_c}(s_n), \quad (11)$$

which indicates how much better an action turns out to be than expected. Here, γ is the discount factor, $\nu_{\vartheta_c}(s_n)$ gives the value prediction on s_n, and ϑ_c is the collection of C-GNN weights. Consequently, minimizing L_{ϑ_a} will reinforce actions resulting larger advantages. The value loss is the average TD error for minimizing the prediction error.

$$L_{\vartheta_c} = \frac{1}{N} \sum_{i=0}^{N-1} \delta_n^2. \quad (12)$$

We calculate gradients of the loss functions with respect to ϑ_a and ϑ_c and apply them to the global GNNs (*Lines 13-14*). Finally, *Lines 15-16* reload A-GNN and C-GNN with the global weights, delete the used samples, and increase the episode count by one to prepare for the next training.

The DRL agent is trained with the following configuration: the neural network employs a hidden layer dimension of 128 with LeakyReLU activation functions, optimized using the Adam algorithm at a learning rate of 10^{-4} to maintain training stability. The A3C framework is deployed with 8 parallel threads, each processing batches of 32 experiences collected from distributed actors. The discount factor is set to $\gamma = 0.9$ to compute cumulative reward.

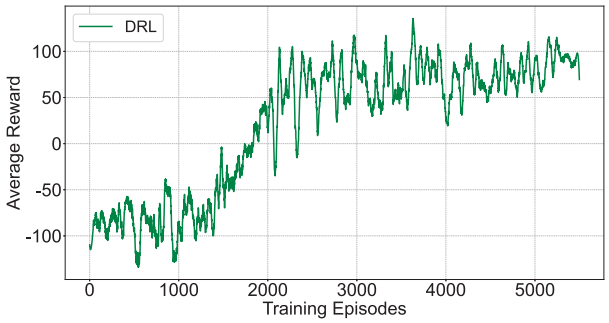
V. Performance Evaluation

We conduct simulations to compare our proposal with existing benchmarks to verify its effectiveness and robustness.

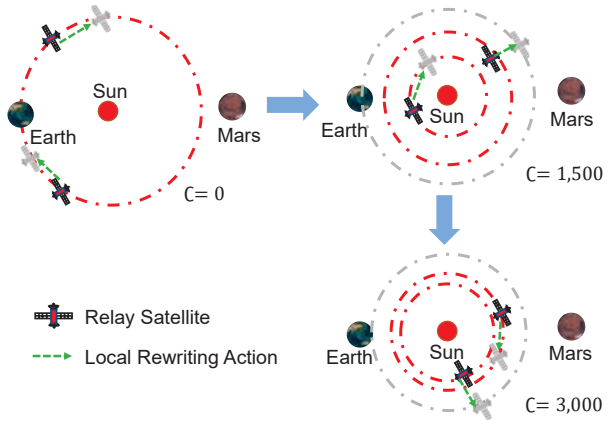
A. Simulation Setup

Fig. 4 displays the 3-domain and 5-domain IPN topologies (denoted as IPN-1 and IPN-2, respectively) used in simulations. IPN-1 spans 3 domains (Earth, Moon, and Mars) and consists of 8 nodes: a ground control center, 3 ground stations, 2 rovers, and 2 satellites. IPN-2 expands IPN-1 by including also the Mercury and Venus systems (14 nodes in total).

We generate IPN scaling solutions for IPN-1 and IPN-2 with the proposed DRL-based approach and three existing benchmarks, namely, "Lagrange" (deploying relay satellites at the two Lagrange points L4 and L5 of Earth), and "LOTS" and "MINLP" in [27]. For fair comparisons, all the approaches use the same numbers of relay satellites. The obtained solutions are assessed by bundle-level fine-grained IP-DT simulations using the routing and data scheduling scheme developed in [24]. Specifically, considering the planetary motion cycles, we extract several time slices from a period of thousands of days and conduct 24-hour simulations in each time slice. The motions of IPN nodes are emulated using the Satellite Tool Kit (STK) [40]. In a simulation, each IPN node generates bundles dynamically according to a Poisson process. Bundles are then assigned different priorities according to their sizes, *i.e.*, copper ($[128, 1024]$ KBytes), silver ($[16, 64]$ KBytes), and gold ($[1, 8]$ KBytes), following a ratio of 18:1:1. The average



(a) Reward



(b) Rewriting actions sampled.

Fig. 5. Performance evolution during training.

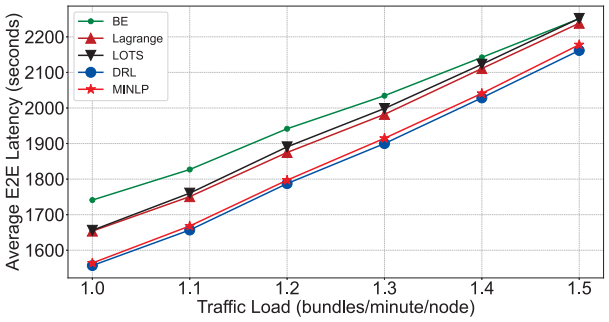
lifetime of a bundle is set as $10,000$ and $20,000$ seconds for IPN-1 and IPN-2, respectively. To ensure statistical reliability, each data point shown later is obtained by averaging the results from 5 independent runs in the simulations.

B. Solving IPN Topology Scaling with DRL Training

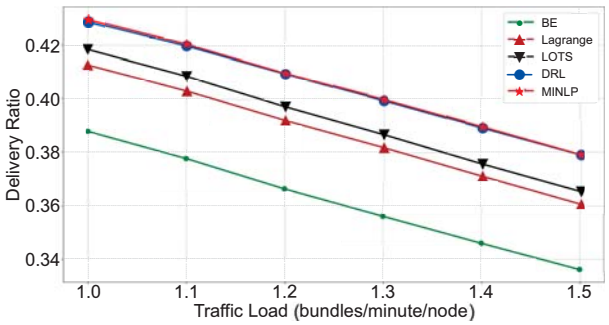
Fig. 5(a) shows the evolution of cumulative reward received by the DRL agents during the training under IPN-1 with a budget of 2 relay satellites. The results indicate that the performance of the learned solution improves steadily and converges after training of $\sim 2,000$ episodes. To provide more insights on how the agents approximate progressively toward the optimal solutions, we sample and visualize the agents' decision making at different training steps in Fig. 5(b). In can be seen that the agents start with random movements of relay satellites from the Earth's orbit (centered on the Sun) and gradually learn to place the two satellites in lower orbits with phases closer to that of Mars (bottom right in the figure).

C. Comparisons with Benchmarks

Next, we compare our proposal with the benchmarks applied to scaling IPN-1 with two relay satellites. Fig. 6 shows the IP-DT performance of the yielded IPN expansions. Here, "BE" refers to the original IPN-1 without adding new relay satellites. Noticeably, scaling the IPN brings significant performance gains in average E2E latency and bundle delivery ratio over the original IPN configuration, and thereby, effectively enhances the reliability of the IPN. The "Lagrange" approach provides



(a) Average E2E latency



(b) Average delivery ratio

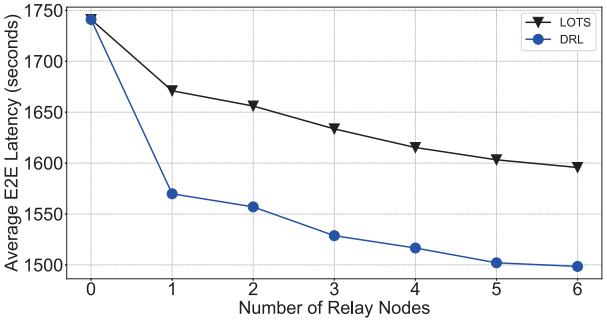
Fig. 6. Results on scaling IPN-1 with two relay satellites.

the worst IPN scaling solutions, attributing to the fixed locations that it chooses to deploy relay satellites. Conversely, by allowing the DRL agents to search the optimization space intelligently, our proposal achieves the best performance. Although "MINLP" solves the planning of relay satellites exactly, its performance degenerates as a consequence of compromise to model tractability, *i.e.*, its operates on a pre-processed discrete solution space to ensure time efficiency.

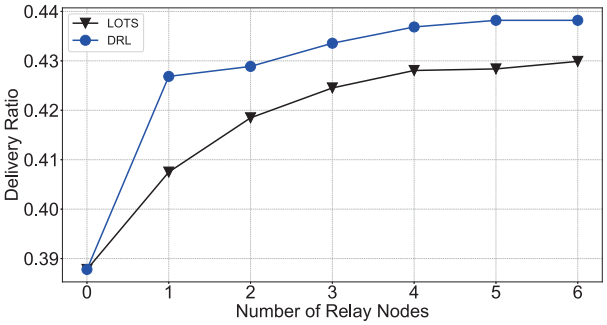
D. Evaluations on Model Robustness

To further verify the robustness of our proposal, we assess it under different numbers of relay satellites and IPN topologies. This time, we exclude "Lagrange" and "MINLP" considering their inapplicability to larger-scale problem instances due to either the intrinsic limitation ("Lagrange" can only place two relay satellites at fixed locations) or high time complexity ("MINLP"). In all the simulations, the traffic load is fixed as one bundle/minute/node. Fig. 7 shows the results of E2E latency and delivery ratio as a function of the number of relay satellites using IPN-1. As expected, increasing the number of relay satellites boosts IP-DT performance and the acceleration of this gain slows down as bandwidth bottlenecks diminish. Our proposal consistently outperforms the benchmark, proving its robustness against different relay satellite configurations.

We also evaluate our proposal and "LOTS" using IPN-2 with different numbers of relay satellites. As the target values and optimal policies with respect to different IPN topologies vary, we get the solutions for INP-2 by fine tuning the agents trained under IPN-1 in the new environment. The results are shown in Fig. 8, which coincide with those in Fig. 7 and further verify the universality of our DRL-based approach.

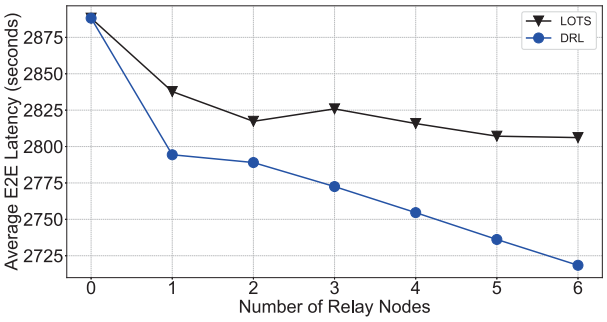


(a) Average E2E latency

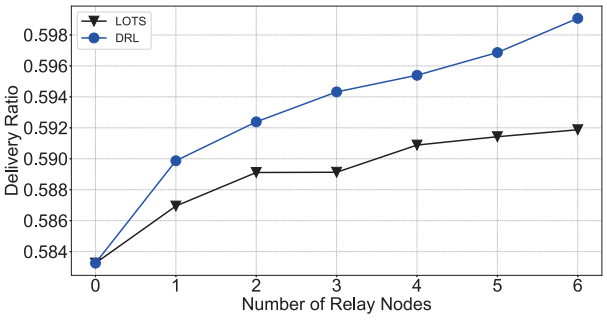


(b) Average delivery ratio

Fig. 7. Scaling IPN-1 with different relay satellites.



(a) Average E2E latency



(b) Average delivery ratio

Fig. 8. Scaling IPN-2 with different relay satellites.

VI. Conclusion

In this paper, we proposed a DRL framework for solving the complex optimization of IPN topology scaling, which needs to jointly consider the planning of relay satellites and the routing and data scheduling for IP-DT. Our proposal modeled the states of an IPN as graph-structured data and utilized GNNs to extract meaningful graph-level representations. Aided by the GNNs, the DRL agents could learn effective local rewriting policies that allow for progressive approximation to the optimal solutions in which the routing and data scheduling of IP-DT achieves maximized performance gain in data transfer reliability and latency. Extensive simulations using different IPN topologies and budgets of relay satellites confirmed that our proposal consistently obtains better scaled IPNs, which deliver superior IP-DT performance over the existing benchmarks. Future research directions include reframing the DRL design to secure improved asymptotic performance with less training steps and better generalization ability, and to explicitly optimize network reliability and fault tolerance.

Acknowledgments

This work was supported by the NSFC project 62371432.

References

[1] I. De Moortel *et al.*, "Roadmap for Solar System research 2022," *White Paper, the Science and Technology Facilities Council (STFC)*, Feb. 2023. [Online]. Available: https://www.ukri.org/wp-content/uploads/2022/12/STFC-050123-RoadmapSolarSystemResearch.pdf.

[2] A. Alhilal, T. Braud, and P. Hui, "The sky is NOT the limit anymore: Future architecture of the interplanetary Internet," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, pp. 22–32, Aug. 2019.

[3] Z. Pan *et al.*, "Advanced optical-label routing system supporting multicast, optical TTL, and multimedia applications," *J. Lightw. Technol.*, vol. 23, pp. 3270–3281, Oct. 2005.

[4] Z. Zhu *et al.*, "Energy-efficient translucent optical transport networks with mixed regenerator placement," *J. Lightw. Technol.*, vol. 30, pp. 3147–3156, Oct. 2012.

[5] Z. Zhu, W. Lu, L. Zhang, and N. Ansari, "Dynamic service provisioning in elastic optical networks with hybrid single-/multi-path routing," *J. Lightw. Technol.*, vol. 31, pp. 15–22, Jan. 2013.

[6] L. Gong *et al.*, "Efficient resource allocation for all-optical multicasting over spectrum-sliced elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, pp. 836–847, Aug. 2013.

[7] Y. Yin *et al.*, "Spectral and spatial 2D fragmentation-aware routing and spectrum assignment algorithms in elastic optical networks," *J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A100–A106, Oct. 2013.

[8] L. Gong and Z. Zhu, "Virtual optical network embedding (VONE) over elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 450–460, Feb. 2014.

[9] C. Chen *et al.*, "Demonstrations of efficient online spectrum defragmentation in software-defined elastic optical networks," *J. Lightw. Technol.*, vol. 32, pp. 4701–4711, Dec. 2014.

[10] P. Lu *et al.*, "Highly-efficient data migration and backup for Big Data applications in elastic optical inter-datacenter networks," *IEEE Netw.*, vol. 29, pp. 36–42, Sept./Oct. 2015.

[11] N. Xue *et al.*, "Demonstration of OpenFlow-controlled network orchestration for adaptive SVC video manycast," *IEEE Trans. Multimedia*, vol. 17, pp. 1617–1629, Sept. 2015.

[12] W. Lu, Z. Zhu, and B. Mukherjee, "On hybrid IR and AR service provisioning in elastic optical networks," *J. Lightw. Technol.*, vol. 33, pp. 4659–4669, Nov. 2015.

[13] J. Liu *et al.*, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Serv. Manag.*, vol. 14, pp. 543–553, Sept. 2017.

[14] P. Lu and Z. Zhu, "Data-oriented task scheduling in fixed- and flexible-grid multilayer inter-DC optical networks: A comparison study," *J. Lightw. Technol.*, vol. 35, pp. 5335–5346, Dec. 2017.

[15] S. Burleigh *et al.*, "Delay-tolerant networking: an approach to interplanetary Internet," *IEEE Commun. Mag.*, vol. 41, pp. 128–136, Jun. 2003.

[16] S. Burleigh, "Interplanetary overlay network: An implementation of the DTN bundle protocol," in *Proc. of CCNC*, pp. 222–226, Jan. 2007.

[17] C. Zhou *et al.*, "Scientific objectives and payloads of the lunar sample return mission–chang'e-5," *Adv. Space Res.*, vol. 69, pp. 823–836, Jan. 2022.

[18] M. Smith *et al.*, "The artemis program: An overview of NASA's activities to return humans to the Moon," in *Proc. of AEROCONF*, pp. 1–10, Aug. 2020.
[19] E. Gibney, "Asteroids, Hubble rival and Moon base: China sets out space agenda," *Nature*, vol. 603, p. 19, Feb. 2022.
[20] M. Wall, "SpaceX's Elon Musk unveils interplanetary spaceship to colonize Mars," 2016. [Online]. Available: https://www.space.com/34210-elon-musk-unveils-spacex-mars-colony-ship.html.
[21] S. El Alaoui and B. Ramamurthy, "MARS: A multi-attribute routing and scheduling algorithm for DTN interplanetary networks," *IEEE/ACM Trans. Netw.*, vol. 28, pp. 2065–2076, Oct. 2020.
[22] X. Tian and Z. Zhu, "On the distributed routing and data scheduling in interplanetary networks," in *Proc. of ICC*, pp. 1109–1114, May 2022.
[23] X. Zhou, X. Tian, and Z. Zhu, "Multi-agent DRL for distributed routing and data scheduling in interplanetary networks," in *Proc. of GLOBECOM*, pp. 4877–4882, Feb. 2024.
[24] X. Tian and Z. Zhu, "On the fine-grained distributed routing and data scheduling for interplanetary data transfers," *IEEE Trans. Netw. Serv. Manag.*, vol. 21, pp. 451–462, Feb. 2024.
[25] J. Breidenthal, "The merits of multi-hop communication in deep space," in *Proc. of AESS 2000*, pp. 211–222, Mar. 2000.
[26] P. Wan and Y. Zhan, "A structured Solar System satellite relay constellation network topology design for Earth-Mars deep space communications," *Int. J. Satell. Commun. New.*, vol. 37, pp. 292–313, Oct. 2019.
[27] X. Tian and Z. Zhu, "On the topology scaling of interplanetary networks," in *Proc. of NaNA*, pp. 274–280, Oct. 2023.
[28] R. Gu, Z. Yang, and Y. Ji, "Machine learning for intelligent optical networks: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 157, pp. 1–22, May 2020.
[29] X. Chen and Y. Tian, "Learning to perform local rewriting for combinatorial optimization," in *Proc. of NeurIPS*, vol. 32, 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/131f383b434fdf48079bff1e44e2d9a5-Paper.pdf
[30] M. Gasse *et al.*, "Exact combinatorial optimization with graph convolutional neural networks," in *Proc. of NeurIPS*, vol. 32, 2019. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/d14c2267d848abeb81fd590f371d39bd-Paper.pdf
[31] H. Zhu *et al.*, "Network planning with deep reinforcement learning," in *Proc. of SIGCOMM*, pp. 258–271, Aug. 2021.
[32] G. Araniti *et al.*, "Contact graph routing in DTN space networks: overview, enhancements and performance," *IEEE Commun. Mag.*, vol. 53, pp. 38–46, Mar. 2015.
[33] S. Haque, "A broadband multi-hop network for Earth-Mars communication using multi-purpose interplanetary relay satellites and linear-circular commutating chain topology," in *Proc. of AIAA*, pp. 1–28, Jan. 2011.
[34] E. Butte, L. Chu, and J. Miller, "An enhanced architecture for the next generation NASA SCaN study," in *Proc. of AIAA*, pp. 1–28, Oct. 2016.
[35] B. Du, F. Gao, and J. Xu, "The analysis of topology based on Lagrange points L4/L5 of Sun-Earth system for relaying in Earth and Mars communication," in *Proc. of ICCSN*, pp. 533–537, May 2017.
[36] W. Kool, H. Van Hoof, and M. Welling, "Attention, learn to solve routing problems!" in *Proc. of ICLR*, Feb. 2019.
[37] B. Niu *et al.*, "Visualize your IP-over-optical network in realtime: A P4-based flexible multilayer in-band network telemetry (ML-INT) system," *IEEE Access*, vol. 7, pp. 82 413–82 423, Aug. 2019.
[38] X. Tian *et al.*, "Reconfiguring multicast sessions in elastic optical networks adaptively with graph-aware deep reinforcement learning," *J. Opt. Commun. Netw.*, vol. 13, no. 11, pp. 253–265, Jul. 2021.
[39] P. Velickovic *et al.*, "Graph attention networks," in *Proc. of ICLR*, vol. 1050, no. 20, pp. 10–48 550, 2018.
[40] Satellite tool kit. [Online]. Available: http://www.agi.com/products/stk/.