October 2023

CROSS CUSTOMERS SMART NETWORK INVENTORY PLANNER (SNIP) AND OPTIMIZATIONS USING DEEP REINFORCEMENT LEARNING

Pengfei Sun

Qihong Shao

Gurvinder Singh

CROSS CUSTOMERS SMART NETWORK INVENTORY PLANNER (SNIP) AND OPTIMIZATIONS USING DEEP REINFORCEMENT LEARNING

AUTHORS:
Pengfei Sun
Qihong Shao
Gurvinder Singh

ABSTRACT

Optimal inventory upgrade planning is one of the most challenging tasks in managing network assets. A Smart Network Inventory Planning (SNIP) architecture or framework is presented herein that leverages a deep reinforcement learning (DRL) framework to enable network inventory upgrade planning for different scenarios. As a foundation for the DRL framework, techniques herein provide for establishing a network inventory environment through which interaction with a supply chain can be used to allow the SNIP architecture to incrementally optimize upgrading sequences for multiple customers. To further optimize inventory upgrades via the DRL framework, the SNIP architecture may employ a multi-objective reward function. Additionally, a transformer can be utilized as a policy network to capture long-term correlations in the inventory upgrading sequence. By incorporating weighting coefficients into both the reward function and a multi-agent actor network, the SNIP architecture can provide customized inventory task scheduling within an optimal framework.

DETAILED DESCRIPTION

One of the most challenging tasks in network assets management is performing optimal inventory upgrade planning. Such planning often involves allocating resources to thousands of hardware and software components, orchestrating a supply chain, and addressing urgent end-of-life timelines. Further, an inventory planner often has to navigate multiple considerations within the confines of budgetary constraints and a limited upgrade time window. Such considerations can include prioritizing end-user needs, accounting for the role of devices in a network topology, addressing security urgency, finding a balance between cost and efficiency, and/or ensuring the viability of the supply chain, such as

through lead times. In essence, an inventory planner tackles a combinatorial optimization problem, similar to a job-shop scheduling problem.

When dealing with partner scenarios, inventory planning can become significantly more complex, as partners are often required to manage and plan upgrades for multiple diverse customers simultaneously, with each customer potentially possessing a substantial number of assets. To streamline the upgrade and renewal processes across all customers, an inventory planner often has to consider additional factors such as customer prioritization, emergent upgrading requests, batch-order discounts, and redistribution cost.

In addressing these specific requirements, existing solutions for inventory planning encounter several challenges:

- Uncertainty in lead time/supply chain outages: In the general job-shop model-based planner, manufacturing time is typically treated as a known distribution on each task, as the processes for different jobs are considered routine procedures. On the other hand, network inventory renewal is directly associated with the supply chain, such that considerations involving lead time are particularly susceptible to supply chain variability, which poses significant challenges for conventional optimization methods that struggle to deal with uncertainties regarding lead times.
- Non-deterministic lifecycle sequential optimization: Conventional planner systems typically rely on linear programming-based optimizations in order to handle relatively small-sized and fixed task lists. However, in network inventory upgrade/renewal scenarios, the large number of assets with diverse functionalities poses a major challenge for traditional optimization techniques that can struggle to process agile and non-deterministic sequence planning within a lifecycle maintenance framework. Furthermore, a network inventory planner can face improvised task insertion, cancelation, re-prioritization, and the like such that the upgrade window often suffers from multiple non-deterministic factors, not limited to customer-wise delay.
- Non-centralized customer-isolated inventory management: In existing network inventory maintenance, partners typically schedule renewals on a customer-isolated, case-by-case basis, which can hinder the optimal allocation of

resources. In contrast, inventory updating involves the coordination of ordering, delivering, and storing network assets, often without centralized planning across multiple customers, which can result in significant resource and time wastage.

In order to address such issues, a Smart Network Inventory Planner (SNIP) architecture is provided, as shown below in Figure 1, that may operate based on a deep reinforcement learning (DRL) framework in order to provide network inventory upgrade planning solutions for different scenarios.

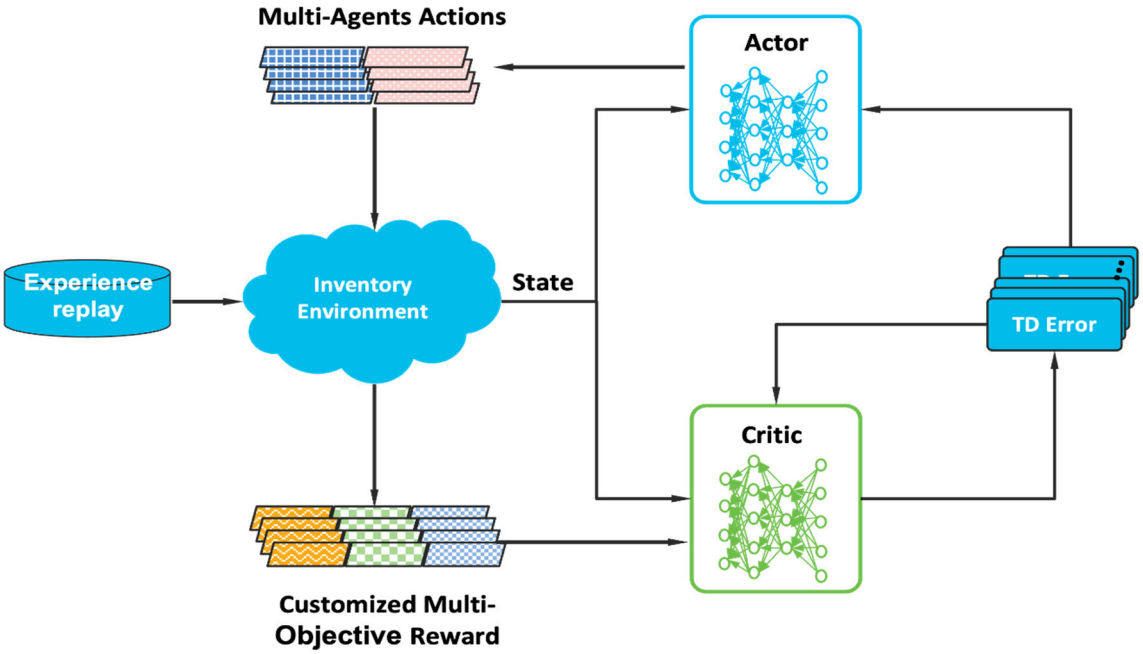

Figure 1: Smart Network Inventory Planning (SNIP) Architecture

The SNIP architecture as shown in the Figure 1 includes three specific features that contribute to the SNIP operational framework: a flexible inventory environment simulator (FIES); customized multi-objective rewarding; and a centralized multi-agents orchestration module. In the FIES, several factors (e.g., supply chain related lead time, end-of-life timeline, capacity related human resources and logistic limitation, and priority relating to customer's service level, critical level (immediate requests), critical level related priority, etc.) are employed to establish a comprehensive inventory environment. As a result, the rewards generated by FIES upon inventory renewal actions can most tailor the policy agents learning. For the customized multi-objective rewarding, a multi-objective

rewarding framework is leveraged to enable the policy agents navigate multiple considerations, for instance, reduced operational cost and optimal upgrading latency. Specifically, customer-depended weightings can be utilized to further address end-users' personalized requirements. Regarding the centralized multi-agent orchestration module, once a Critic network receives the customized multi-objective rewards and the updated state of inventory environment, the generated temporal difference errors can be further fed into the actor network to train the multi-agents policy. By learning the values generated from Critic network, an Actor network orchestrates the multi-agents' reactions in a centralized and ensemble manner.

Broadly, during operation of the SNIP framework, the FIES comprehensively mimics the relations among assets, lead times, operational costs, product discount, EOX (End of Life, End of Maintenance, End of Support, etc.), etc. Thus, the FIES associates the inventory upgrade/renewal pipeline with the supply chain. When the agents (referring to the policy network) take actions, the inventory environment can update the state and generate multi-objective rewards that can then be utilized to train the actor-critic learning system. Notably, the customized multi-objective rewarding and the centralized multi-agents' modules may incorporate customer-specific weightings to discern and address diverse customer requirements.

Consider additional details regarding the FIES, as shown below in Figure 2.

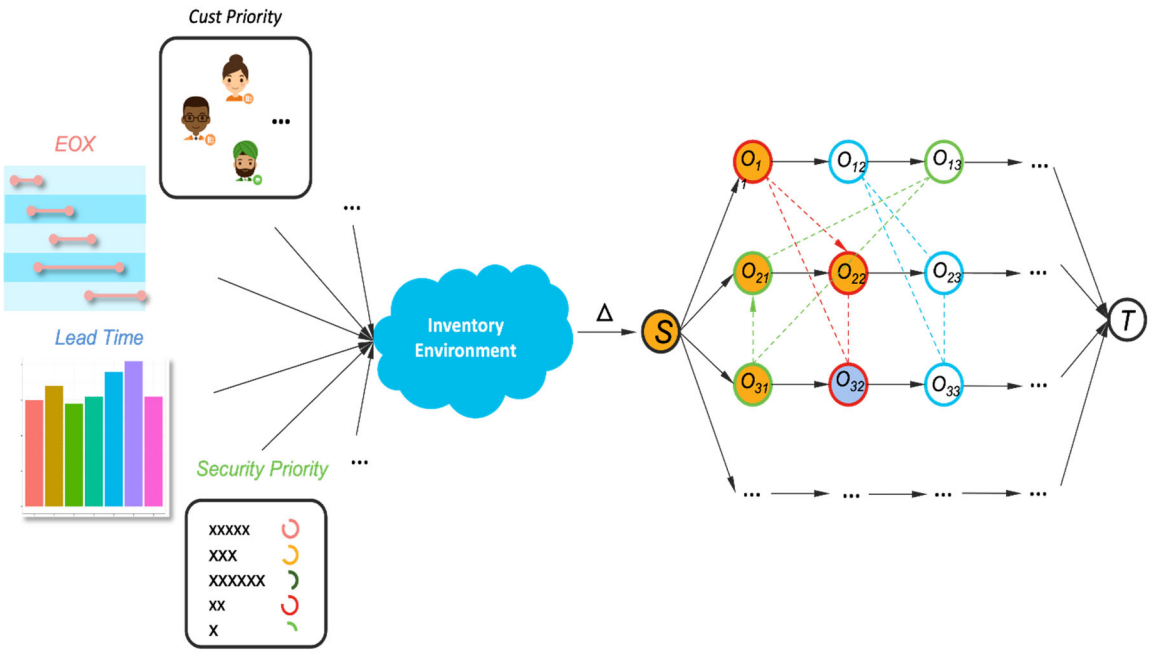

Figure 2: Flexible Inventory Environment Simulator (FIES) Framework

With reference to Figure 2, the FIES can be leveraged to comprehensively capture correlations and adhere to updating and renewing constraints within the inventory management framework. Thus, the FIES is to effectively interact with these changes and accurately estimate the rewards to reflect the impact of agents' actions, such as device upgrades, cancellation of renewals, and re-prioritizing device replacements.

As shown in Figure 2, multiple factors can be employed to build the FIES: 1) the EOXs of assets (e.g., the EOXs in the FIES can help to avoid upgrading past due); 2) the critical level of security (e.g., devices/software with urgent security issues should be upgraded first); 3) customer service level (e.g., end-users with higher priority should receive quicker upgrading relatively); and 4) the lead time in the supply chain (e.g., the lead time fundamentally affects the time window to process the order and ship; those replacements with longer lead time in general should be prioritized in the upgrading queue).

By considering such factors in the FIES, the inventory environment can return well-balanced reward vectors in responses to upgrading/replacement actions. Here, the reward vectors refer to the gain when the agent's action are evaluated based on the FIES such that the gain is considered to be positive when the FISE encourages such action, and vice versa.

In more detail, the inventory environment includes the sample space, where each sample represents correlations or constraints among the assets. As depicted in Figure 2, the interconnected graph that represents the upgrading sequences can be seen as samples from the FIES. To optimize the agents' actions in the real-world application scenarios, the planner relies on the FIES to provide a full simulation within the inventory environment.

The SNIP further employs a multi-objective rewarding system to incentivize or penalize agent actions. Traditional reinforcement learning frameworks typically use a single objective to evaluate agent actions, such as the distance to a target in a robot arm environment. However, in inventory planning scenarios, end users often desire both lower upgrading latency and minimal operational cost. To address this, a multi-objective reward function can be employed to explore optimal solutions.

Nevertheless, applying the multi-objective reward function uniformly across all agents can overlook the distinct upgrading needs of end-users. For instance, end-user #1

may prioritize switch upgrading, while end-user #2 may prioritize security product upgrading. A global criterion may not adequately reflect the diverse interests among different users. Therefore, to ensure that SNIP caters to customized inventory upgrading/renew schedule, a weighting system will be incorporated into the inventory environment. Figure 3, below, illustrates the inclusion of customer-specific weightings into the reward function.

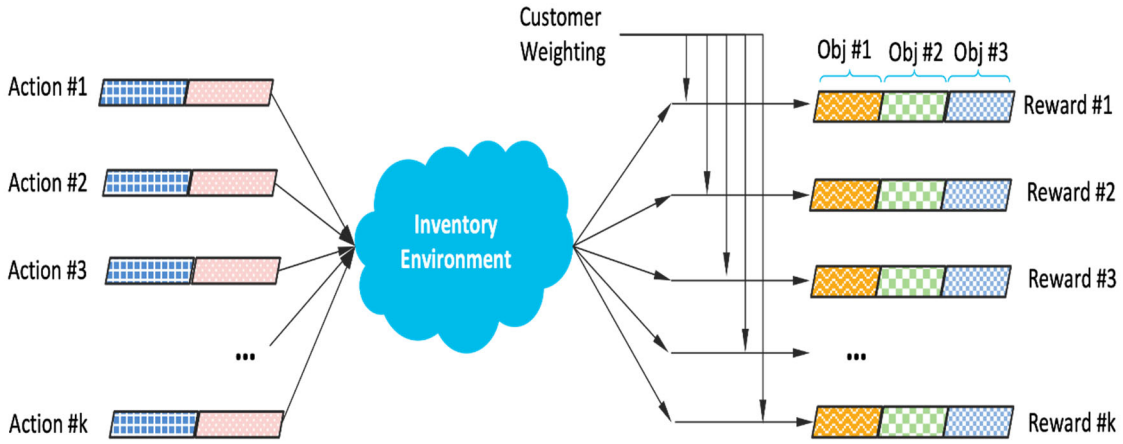

Figure 3: Multi-Objective Reward Diagram

As illustrated in Figure 3, when agent #k takes action #k, the FIES evaluates the action and generates a multi-objective reward vector #k, considering the customer #k's weighting for different objective. Thus, the general reward function is formulated to jointly account for the objectives and incorporate the customer-specific weightings. The SNIP can assign each end-user (e.g., group/unit) an agent to simulate a cooperative environment, within which individual agents can emphasize different aspects with respect to an end-user's own interests/priorities.

Centralized multi-agent orchestration can also be provided by the SNIP. For example, in the SNIP, each agent can perform actions such as upgrading devices and learns from the rewards generated by these actions within the environment. The goal for each agent is to maximize the rewards and optimize its ability to select the appropriate action based on the FISE.

However, it is important to note that the optimal action for an individual agent may not necessarily lead to a global optimal action across multi-agents. Considering factors such as storage cost, batch buying discounts, and other considerations, a centralized

inventory maintenance approach would be more efficient and cost-effective. Consequentially, centralized inventory management requires agents to collaborate with each other to achieve global optimal rewards. To strike a balance among multi-agents, the SNIP employs an approach as shown below in Figure 4 in order to orchestrate the optimization of agent policies.

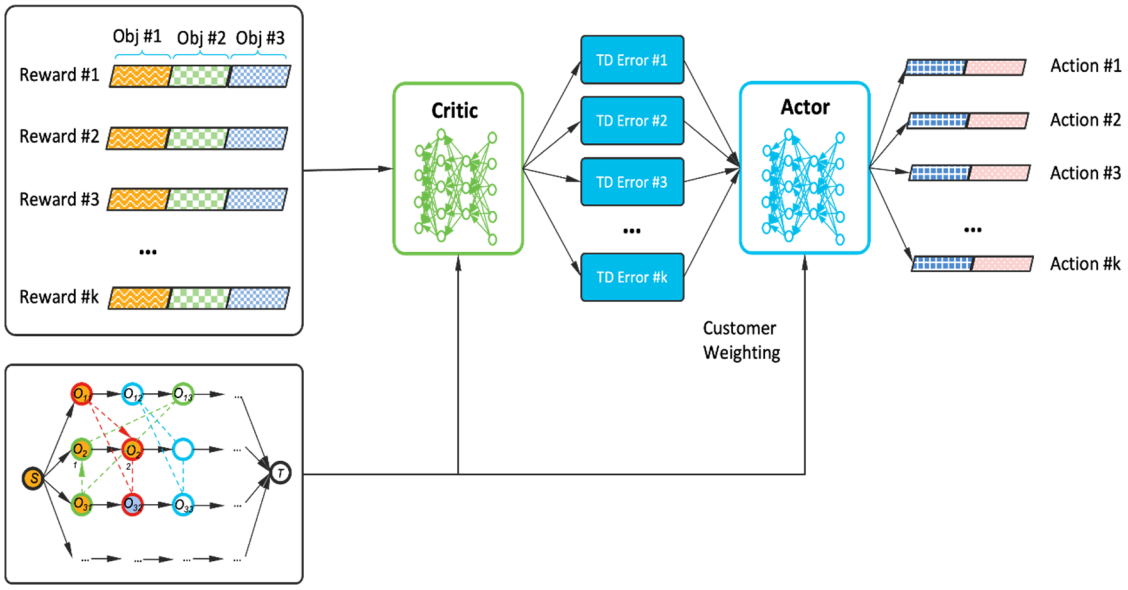

Figure 4: Multi-Agent Action Diagram

It is typically not clear how to evaluate available trade-offs between different objectives, and there is no single optimal policy. However, in accordance with techniques of this proposal, for multiple-agent reinforcement learning, different customers can be represented by a single agent. By applying separate customized rewarding into the critic network that generates independent temporal difference loss, the multi-agent actor network can orchestrate optimizations among multiple customers.

Thus, the dynamic inventory environment of the SNIP framework provides a multi-dimension feature space, in which customer dynamics, such as customers' decisions including prioritizing certain device upgrading, re-entry/withdrawal from upgrading queue, are included in the reinforced learning framework. By adding such factors as constraints, the reinforced learning model can adapt the actions according to the customer's request. Not limited to responding to prioritization or skipping specific upgrading, the dynamic inventory environment of the reinforced learning framework provided by the SNIP architecture employs more comprehensive constraints (e.g., customer's requests on

selected time window for delivery, grouping of device delivery for batch upgrading) to further drive the planner to be a highly flexible and customer-centric optimizer. In essence, the reinforced learning framework presented herein relies on building both a supply-chain and customer-dependent inventory environment.

Consider an example scenario involving brownfield managed service providers (MSPs). MSP brownfield partners typically work on multiple customers' inventory maintenance, and in conventional framework, such partners often manually process device/software upgrades based on their experiences (typically occurring bi-annually or annually). Tens of customers by thousands of devices per customer can be overwhelming to manage, however, the SNIP framework may address this problem utilizing the multi-customer inventory upgrade process, as shown in Figure 5, below.

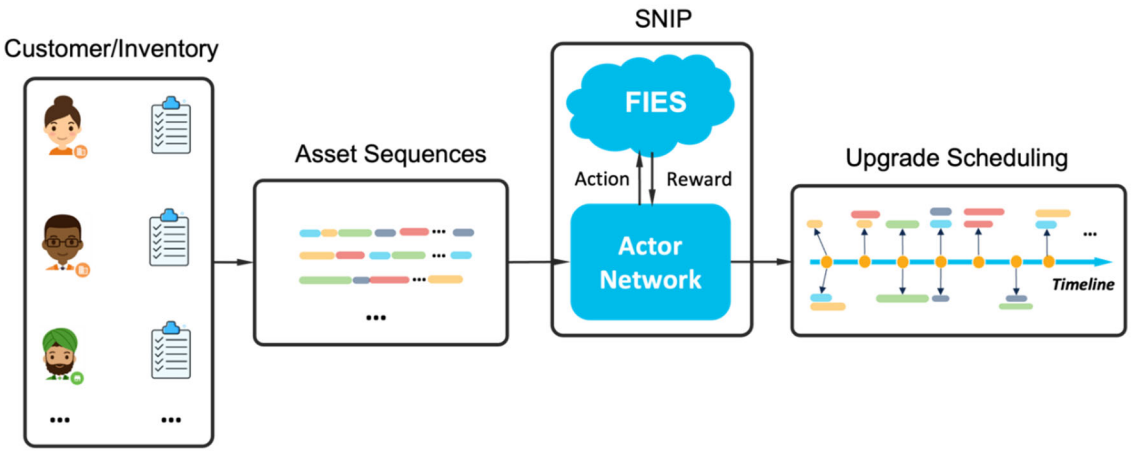

Figure 5: Multi-Customer Inventory Upgrade Planning

As generally illustrated for Figure 5, the customers/inventories are transformed into multiple asset sequences to be upgraded/replaced in which each single sequence corresponds to a specific end-user. Once the asset sequences are obtained, the pre-trained SNIP system, including the FIES and Actor network, will cooperatively consume the asset sequences piece by piece. For instance, the FIES can send rewards to the Actor network (i.e., the policy network), and Actor network can determine the next action based on the reward value and current state.

Incrementally, the SNIP system can make choices on upgrade tasks based on the input asset sequences and those choices can be driven by the objective of minimal cost and optimal latency. Explicitly, optimal upgrade strategies can be achieved through the interaction between the Actor network and the FIES. Once sub-upgrading-tasks in the asset

sequences are fed into the SNIP, a single upgrade schedule along the timeline can be obtained.

Thus, broadly, techniques presented herein may provide for the ability to associate renewal planning jointly with supply chain and the dynamic inventory environment. The SNIP framework provides for incorporating the supply chain and multiple factors, such as job queue dynamic, customer dynamics, and priorities dynamics, into the inventory environment. Through a flexible inventory environment, SNIP framework can generate comprehensive rewards to determine optimal upgrading actions.

Further, the SNIP framework incorporates customer-dependent multi-objective rewarding into the DRL framework. This customized rewarding enables the inventory environment to differentiate the responses to different end-users in order to optimize inventory upgrade scheduling with multi-objective deep reinforcement learning. Additionally, by utilizing multi-agents in the deep reinforcement learning, the SNIP framework can leverage a joint rewarding system to achieve a balanced optimization between maximizing individual-customer's interests and partner's benefits in order to facilitate multi-agent-based customer-wise and across-customer inventory planning.

Unlike conventional inventory management systems that focus on "within-organization scheduling," the leveraged SNIP framework can strike a balance between maximizing a single-customer interest and a customer-group's interests in order to facilitate centralized, multiple customer inventory upgrade optimizations. Generally, a customer-group's interest may refer to a balanced/global optimization among different customers. For example, within a given time window and partner's capacity, a service provider (partner) may seek to maximize customer A's interests, without delaying upgrades or increasing the costs for customer B. In other words, a global optimal solution with compromised single-customer interests can be chosen from the perspective of overall reward across all the customers utilizing the SNIP framework.

Moreover, since the SNIP framework uniformly leverages a multi-customer and multi-task inventory environment, the multi-objective rewarding function (i.e., cost and time) can be generalized such that different network source management goals can be realized. For example, the customer-dependent orchestration system in the SNIP

framework can be enhanced to other application scenarios, such as lifecycle network management.

Along with inventory upgrade/replacement, lifecycle adoption is also a very time-resource competitive task. As shown in Figure 6, below, lifecycle adoption for an example scenario can involve end-users completing different tasks with a large number of resources. For instance, with adoption tasks ranging from campus network to security endpoint, there can be more than 700 resources to consume for assisting task completion. As such, it can be quite exhausting for end-users to complete such tasks without prioritization and optimization.

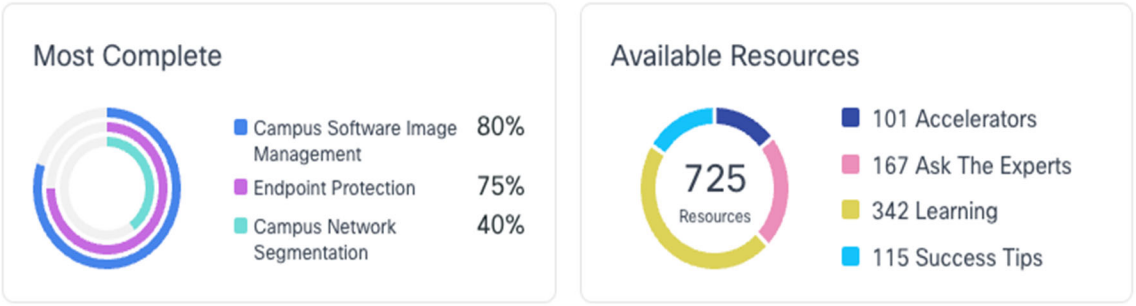

Figure 6: Example Customer Lifecycle Adoption Interface

Intrinsically, in inventory upgrade/renewal, different asset replacements can involve different times to delivery and, comparatively, can result in different in lifecycle adoption times such that the resources used to assist the different assets' adoption can involve different times. For such scenarios, the SNIP framework can be leveraged in order to optimize the adoption across multiple customers and large asset portfolios.

Figure 7, below, illustrates an example multi-customer lifecycle adoption planning pipeline that may be facilitated by the SNIP framework.

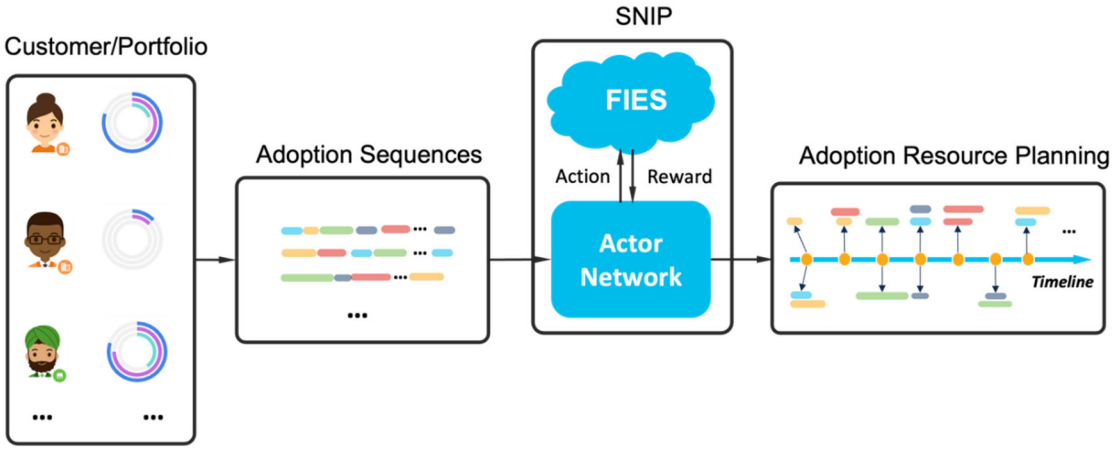

Figure 7: Multi-Customer Lifecycle Adoption Planning Pipeline

As illustrated for the lifecycle adoption planning pipeline shown in Figure 7, the adoption tasks can be transformed into sequences for each end-user; specifically, instead of using the upgrading time as the optimization objective, here, the time of consuming the resources plus the corresponding adoption procedures for the resources can be used as the optimization objective, with cost metrics removed from the rewarding function. Thus, the SNIP framework can be employed to predict the next adoption by minimizing the overall time spent on lifecycle tasks such that the output of the SNIP may correspond to an adoption resource planner through which end-users can implement adoption through the resource consumption with the minimal overall time.

Accordingly, techniques presented herein facilitate constrained policy optimizations that provide for transforming a customer's needs/requests into multiple constraints, which can be used to control the policy optimizations. Unlike generalized reinforced learning scenarios that directly manipulate reward functions to indirectly address constraints in policy searching, techniques herein build such constraints into the reinforced learning environment without manipulating the rewards function. Moreover, techniques presented herein may enable knowledge transfer learning, which allows the SNIP framework to overcome any lack of inventory-customer-supply-chain interaction data.

Additionally, techniques as presented herein may be distinguished from other potential supply chain solutions by incorporating both the dynamics from a supply-chain and customer’s needs into the inventory environment to allow comprehensive action

reinforcement for an inventory management environment that incorporates different factors, such as prioritization, device deadline, budget cap, selected delivery window, among others, that significantly differentiate the inventory planning techniques proposed herein from other potential solutions.

In summary, the novel techniques presented herein may be generalized across different perspectives. From a methodology perspective, the SNIP framework may leverage a reward-free based constrained policy optimization within the reinforcement learning framework to facilitate inventory management or can be broadly applied across any applicable management paradigm (e.g., lifecycle adoption, etc.). Compared to general reinforcement learning scenarios that do not involve policy constraints in supply chain and inventory management, the SNIP framework may enable building a highly comprehensive environment (including multi-fold constraints, such as, upgrade deadlines, customer's choice, service provider's operational cost budget, etc.) to directly respond to and refrain an agent's actions instead of indirectly adding penalties to the reward.

From an application perspective, the leveraged reinforced learning model of the SNIP framework provides a proactive planner that not only passively reacts to a supply chain's volatility, but also optimizes actions through interactions with customers (e.g., via a customer's request for re-entering an upgrading queue or skipping certain devices, requesting a selected delivery window, grouping devices delivering for batch upgrading, etc.).

Finally, from a reinforced learning framework perspective, the SNIP framework innovatively utilizes flexible job-shop planning as an external expertise to enable transfer learning. In other words, the large number of off-policy instances from flexible job-shop planning can be incorporated into the DRL model training for inventory planning, lifecycle planning/adoption, or any other applicable network management scenario in accordance with the techniques presented herein.