Zero-shot Generalization in Inventory Management: Train, then Estimate and Decide

Tarkan Temizöz*[1], Christina Imdahl[1], Remco Dijkman[1], Douniel Lamghari-Idrissi[1,2], and Willem van Jaarsveld[1]

[1]Eindhoven University of Technology, Eindhoven, Netherlands
[2]ASML, Veldhoven, Netherlands

Extended abstract submitted for presentation at the 12th Triennial Symposium on Transportation Analysis conference (TRISTAN XII)
June 22-27, 2025, Okinawa, Japan

March 28, 2025

Keywords: Supply Chain Management; Deep Reinforcement Learning; Zero-shot Generalization

1 INTRODUCTION

The abundant data and computational advances have shifted supply chain management toward data-driven methodologies (Mišić & Perakis, 2020). As a result, there's growing interest in applying machine learning (ML) (Qi *et al.*, 2023) and deep reinforcement learning (DRL) (Gijsbrechts *et al.*, 2022) to develop sophisticated solutions beyond traditional heuristics. However, deploying ML in real-world sequential decision-making faces significant challenges. For example, frequent shifts in demand patterns and supply uncertainties limit reliance on historical data (Gong & Simchi-Levi, 2023). The unpredictable and dynamic nature of operational environments demands policies that are robust and adaptable to unseen conditions (Kirk *et al.*, 2023).

Moreover, decision-makers (DMs) often lack a complete understanding of problem parameters affecting state transitions. Optimal actions depend on these parameters, making it essential to estimate and adapt to the true parameters for effective ML/DRL deployment and requiring frequent parameter updates. Typically, ML/DRL policies are trained for specific parameter sets, but remodeling and retraining when parameters change is computationally intensive, especially when managing numerous tasks like thousands of products. These challenges highlight the need for a unifying framework for sequential decision-making under uncertain and changing parameters. We address this by exploring training generally capable agents (GCAs) under zero-shot generalization (ZSG). In our context, *GCAs* refer to advanced DRL policies that perform well across a broad range of sampled problem instances with diverse supply chain challenges while *ZSG* means that a GCA can be directly applied to new instances with unknown parameters that it was never explicitly trained on. Motivated by bringing DRL closer to practical applications, we propose a unifying MDP formulation and a solution framework to train and deploy a GCA tailored to supply chain management problems.

In particular, we present a unifying Super-Markov Decision Process (Super-MDP) formulation and the Train, then Estimate and Decide (TED) framework to train and deploy a GCA when the problem parameters are unknown. The TED framework consists of three phases: training a GCA on varied problem instances, continuously estimating problem parameters during deployment, and making decisions based on these estimates.

*Corresponding author: t.temizoz@tue.nl

2 PROBLEM DESCRIPTION

In this section, we formally define the problem setting of this paper. Let $\mathcal{D}$ represent a class of sequential decision-making problems, influenced by a predefined parameter space $\mathcal{P}$. Within $\mathcal{D}$, the DM is tasked with managing n independent and distinct tasks, each requiring the solution of an MDP ($\mathcal{M}^{\mathbf{p}}$) tailored to its respective parameterization $\mathbf{p} \in \mathcal{P}$. These parameters may be subject to external factors and can change over time. Consequently, the DM may encounter three common challenges (decision contexts) when managing the sequential decision-making problems within $\mathcal{D}$:

- *(Scalability-1)* The DM must make decisions for n independent tasks, each characterized by different parameterizations. This necessitates solving at least n distinct MDPs, each tailored to its specific parameterization, and employing n different decision-making policies. As n increases, this approach becomes time-consuming and impractical.
- *(Non-stationarity-2)* The parameters (parametrizations) within a specific task may evolve over time, requiring the DM to solve new MDPs for updated parameter combinations. Additionally, new tasks with unknown parameterizations from the established parameter space may emerge at any point, such as through the introduction of new customer groups or products.
- *(Obscurity-3)* The DM often lacks direct observation of the true parameters and must rely on inferences drawn from potentially limited and censored real-time data.

To address the challenges associated with solving $\mathcal{D}$, our primary objective is to train a GCA that can perform effective real-time decision-making across problems with diverse parameterizations without requiring additional training, achieving what is known as ZSG. This agent should be applicable to the three decision contexts outlined above. Since context (1) involves applying the GCA in a scenario with perfect parameter estimates and context (2) typically results in unknown parameters that require estimation, we emphasize that contexts (1) and (2) can be viewed as special cases of context (3). Therefore, our framework and analysis primarily focus on context (3). Moreover, we assume that the DM does not know when or how the parameterization will change. Without this knowledge, the DM cannot utilize probabilistic information to anticipate and optimize for potential parameter changes. This scenario leads to the following assumption:

Assumption 1 (Decision Maker's Optimization Rationale)**.** *The decision maker assumes stationarity of the problem parameters and aims to optimize the policy accordingly, pursuing optimal decision-making as long as the problem's parameterization remains unchanged.*

Assumption 1 enables the decomposition of the problem into a sequence of independent and stationary MDPs, each characterized by a distinct parameterization. Hence, we decompose the problem $\mathcal{D}$ into independent and stationary MDPs and define Super-MDP. The Super-MDP can be understood as a population of all MDPs related to our decision problem $\mathcal{D}$ and thus formally defines the problem $\mathcal{D}$ as follows:

Definition 1 (Super-Markov Decision Process)**.** *A Super-Markov Decision Process is defined by the tuple $\mathcal{M}_{\mathcal{S}} = (\mathcal{P}, \mathcal{S}, \mathcal{A}, \mathcal{H}, \mathcal{F})$, where:*

- *$\mathcal{P}$ represents the parameter space, containing the true problem parameterizations $\mathbf{p} \in \mathcal{P}$.*
- *$\mathcal{S}$ denotes the finite state space, encompassing all possible states $\mathbf{s} \in \mathcal{S}$.*
- *$\mathcal{A}$ indicates the finite action space, where each action $a \in \mathcal{A}$, and $\mathcal{A} = \{0, 1, \ldots, m\}$.*
- *$\mathcal{H}$ is the distribution over the parameter space $\mathcal{P}$, generating problem parameterizations $\mathbf{p} \sim \mathcal{H}$.*

- *$\mathcal{F}$ is the mapping function that relates each parameterization $\mathbf{p}$ to the corresponding elements of $\mathcal{M}^{\mathbf{p}}$ (i.e., $f^{\mathbf{p}}$, $C^{\mathbf{p}}$, $\mathbf{s}_0^{\mathbf{p}}$). Both the state space and action space are defined universally across parameterizations, such that $\mathcal{S}^{\mathbf{p}} \subseteq \mathcal{S}$ and $\mathcal{A}^{\mathbf{p}} \subseteq \mathcal{A}$.*

Note that the true parameter space $\mathcal{P}$ and the distribution of parameterizations $\mathcal{H}$ may be unknown. We later define and utilize a probable parameterization space and establish a parameter sampling function to conceptualize the Super-MDP.

The definition of the Super-MDP aligns with the objective of training a GCA. If a policy is a GCA, it should generate effective actions for a problem instance generated by $\mathcal{H}$ from the parameter space. A policy for a Super-MDP, denoted as π_S, can be defined as a function from state-parameterization pairs to actions, $\pi_S : \mathcal{S} \times \mathcal{P} \to \mathcal{A}$. Our objective is to identify a jointly optimal policy π_S^*, defined as $\pi_S^* = \arg\min_\pi \mathbb{E}_{\mathbf{p}\sim\mathcal{H}} \left[\lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1} C^{\mathbf{p}}(\mathbf{s}_t, \pi(\mathbf{s}_t, \mathbf{p}))\right]\right]$.

Since finding the optimal policy for even moderately sized MDPs is intractable, we must rely on approximation methods.

3 TRAIN, THEN ESTIMATE AND DECIDE

Achieving an approximately optimal policy for a Super-MDP requires a solution approach capable of simultaneously addressing MDPs with varying parameterizations, thereby obtaining a GCA. Our primary strategy involves separating the training and deployment phases of this policy, ensuring that it does not require retraining when the problem's parameterization changes, thus achieving ZSG. We propose such a solution within our *Train, then Estimate and Decide-TED* framework.

In the *Train* phase, we first construct the Super-MDP for the problem class $\mathcal{D}$. Since the true parameter space $\mathcal{P}$ and the distribution over the parameter space $\mathcal{H}$ are unknown, we define a probable parameter space $\hat{\mathcal{P}}$ and an associated distribution $\hat{\mathcal{H}}$, such that $\hat{\mathcal{P}}$ fully contains the true parameter space, i.e., $\mathcal{P} \subseteq \hat{\mathcal{P}}$, and $\hat{\mathcal{H}}$ ensures uniform coverage of $\mathcal{P}$. The policy (GCA) is parameterized as a neural network, and we sample from the probable parameter space for training (see Figure 1, left part of the Train box) using $\hat{\mathcal{H}}$. The policy is trained for each sampled parameterization under the assumption that the parameterization remains fixed during training (see Figure 1, right part of the Train box). The parameterization of the problem is incorporated into the policy (the neural network) as input features, enabling the neural network to generalize to unseen parameterizations during the deployment phase, which comprises the *Estimate* and *Decide* steps.

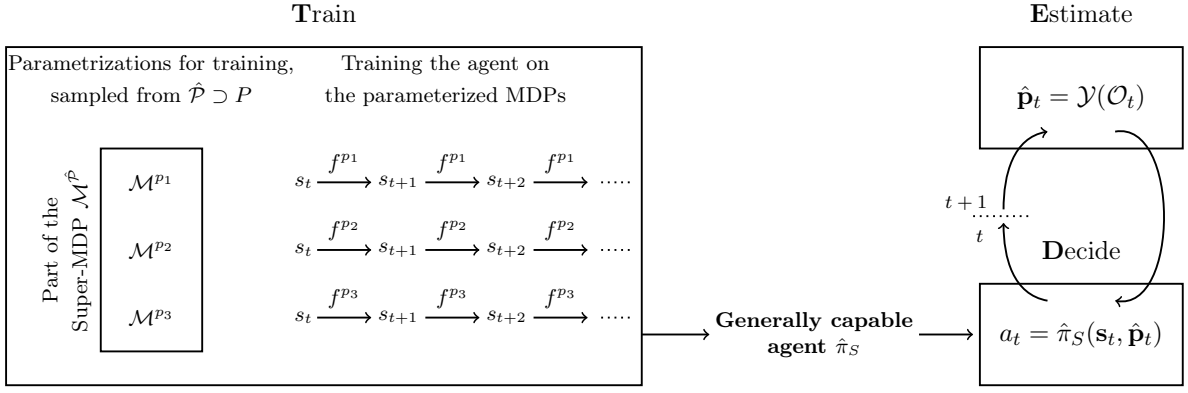

Figure 1 – *Train, then Estimate and Decide framework for solving sequential decision-making problems with dynamic parameter estimation*

In the *Estimate* phase, we estimate the parameterization $\hat{\mathbf{p}}_t$ of the actual system at time t using a function $\mathcal{Y}$ based on the observations collected up to time t ($\mathcal{O}_t$). In the *Decide* phase,

this parameterization estimate is fed into the pre-trained policy along with the current state. The policy then outputs the action to be taken at time t (see right part of Figure 1).

The main novelty of this approach is that the training step, during which the policy is optimized, precedes the estimation step. This contrasts with current prevalent approaches, which first estimate parameters and then optimize based on parameter estimates; see Lagos *et al.* (2024) and Lyu *et al.* (2024) for similar applications covering online learning in last-mile logistics and inventory management.

4 RESULTS AND DISCUSSION

To demonstrate the effectiveness of our TED framework, we apply it to a class of inventory problems involving periodic review, and further characterized by lost sales, and possibly by cyclic demand patterns and stochastic lead times. For this class of inventory problems, we define a Super-MDP and train a GCA during the Train phase, named *Generally Capable Lost Sales Network (GC-LSN)*. To validate it, GC-LSN is rigorously benchmarked versus the base-stock and capped base-stock policies on a wide range of problems assuming full availability of the parameters of each problem instance, both for GC-LSN and for the benchmarks. We find that GC-LSN consistently outperforms these well-performing benchmarks. Moreover, as part of the TED framework, we test GC-LSN under conditions where demand and lead time distributions are initially *unknown* and must be estimated. For the Estimate phase, we adopt the non-parametric Kaplan-Meier estimator (Kaplan & Meier, 1958) to estimate the demand distribution and construct a relative frequency distribution for the lead time distribution. Our experiments show that TED outperforms a range of benchmarks specifically designed for online learning (Lyu *et al.*, 2024). To our knowledge, TED stands uniquely as the only general-purpose algorithmic framework capable of addressing these diverse inventory challenges collectively, particularly when key information such as demand and lead time distributions are initially unknown and when data is initially limited and censored. Our code can be accessed through our GitHub repository https://github.com/tarkantemizoz/DynaPlex.

We believe that logistics providers can leverage our approach to improve inventory, routing, and distribution decisions in real-time, adapting to disruptions such as transportation delays or sudden shifts in customer demand. By enhancing decision-making under uncertainty and dynamic conditions, our work offers a flexible and efficient solution that can increase the responsiveness and resilience of supply chains in a rapidly evolving market environment.

References

Gijsbrechts, Joren, Boute, Robert N., Van Mieghem, Jan A., & Zhang, Dennis J. 2022. Can Deep Reinforcement Learning Improve Inventory Management? Performance on Lost Sales, Dual-Sourcing, and Multi-Echelon Problems. *Manufacturing & Service Operations Management*, **24**(3), 1349–1368.

Gong, Xiao-Yue, & Simchi-Levi, David. 2023. Bandits atop Reinforcement Learning: Tackling Online Inventory Models with Cyclic Demands. *Management Science*.

Kaplan, Edward L., & Meier, Paul. 1958. Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, **53**(282), 457–481.

Kirk, Robert, Zhang, Amy, Grefenstette, Edward, & Rocktäschel, Tim. 2023. A Survey of Zero-Shot Generalisation in Deep Reinforcement Learning. *Journal of Artificial Intelligence Research*, **76**(1), 201–264.

Lagos, Tomás, Auad, Ramón, & Lagos, Felipe. 2024. The Online Shortest Path Problem: Learning Travel Times Using a Multiarmed Bandit Framework. *Transportation Science*.

Lyu, Chengyi, Zhang, Huanan, & Xin, Linwei. 2024. UCB-Type Learning Algorithms with Kaplan–Meier Estimator for Lost-Sales Inventory Models with Lead Times. *Operations Research*, **0**(0), null.

Mišić, Velibor V., & Perakis, Georgia. 2020. Data Analytics in Operations Management: A Review. *Manufacturing & Service Operations Management*, **22**(1), 158–169.

Qi, Meng, Shi, Yuanyuan, Qi, Yongzhi, Ma, Chenxin, Yuan, Rong, Wu, Di, & Shen, Zuo-Jun (Max). 2023. A Practical End-to-End Inventory Management Model with Deep Learning. *Management Science*, **69**(2), 759–773.