Industrializing Deep Reinforcement Learning for Operational Spare Parts Inventory Management

Joost F. van der Haar
Research Center for Operations Management, KU Leuven, joost.vanderhaar@kuleuven.be

Willem L. van Jaarsveld
Department of Industrial Engineering & Innovation Sciences, Eindhoven University of Technology w.l.v.jaarsveld@tue.nl

Rob J.I. Basten
Department of Industrial Engineering & Innovation Sciences, Eindhoven University of Technology r.j.i.basten@tue.nl

Robert N. Boute
Research Center for Operations Management, KU Leuven; Vlerick Business School; Flanders Make @KU Leuven
robert.boute@kuleuven.be

We show how Deep Reinforcement Learning (DRL) can improve industrial-scale operational spare parts inventory management. Spare parts inventory is crucial for the timely maintenance of capital goods. Operational spare parts management requires fast decision-making for complex and large-scale service networks. Therefore, existing work focuses on computationally-light heuristics to maintain tractability, even if that comes at a cost in performance. This is problematic, as lower performance in this context may mean unnecessary downtime on a bottleneck machine, or excessive spending on inventory procurement and expedited shipments. DRL models can theoretically be trained to take near-optimal decisions for such complex problems almost instantaneously, yet training DRL models for industrial-scale inventory systems remains an open challenge. We propose a novel DRL approach that builds on three techniques: *global learning* to scale over an arbitrary number of SKUs, *action space decomposition* to cope with a large number of locations, and *reward smoothing* to efficiently handle stochastic and sparse demand. We demonstrate the effectiveness of our approach on the service network of ASML, a leading company in the semiconductor industry. The results show that our DRL approach outperforms existing methods by 3.8% to 5.1% in terms of cost on a fully connected service network with 10,000 stock keeping units and 60 locations. We show how DRL can improve upon the state-of-the-art in operational spare parts management, and extend the applicability of DRL to industrial-scale inventory systems.

Key words: Deep Reinforcement Learning, Inventory Management, Service Logistics, Reward Smoothing, Action Space Decomposition, Global Learning

1. Introduction

Timely availability of spare parts for capital goods is crucial to perform maintenance and keep primary processes running. Unplanned downtime is estimated to cost industrial manufacturers $50bn annually (IndustryWeek and Emerson 2017), as it leads to opportunity costs and excessive maintenance. Manufacturers lose on average $125,000 per hour due to

asset downtime (ABB 2023). For example, each minute a lithography machine is down can lead to thousands of euros in missed production (ASML 2024). However, spare parts availability requires significant capital investment, and high stock levels may lead to excess and obsolescence (Hu et al. 2018). Capital goods are often designed to last multiple decades, yet many of their components have life cycles of only a few years (Zheng et al. 2015). Balancing these availability and procurement risks is a key capability for maintenance service providers, which are often the Original Equipment Manufacturers (OEMs). OEMs' service operations contribute to 26% of revenues and 46% of profits according to Glueck et al. (2007), who surveyed 120 OEMs worldwide with a combined revenue of over $1500bn. Moreover, industrial repair and maintenance service revenues are estimated to amount to $158.74bn in the US alone (US Census Bureau 2020).

Because of this profound impact, spare parts management has been the subject of much research, especially on a tactical level (see, e.g., Hu et al. 2018, Basten and van Houtum 2023). Literature on *operational* spare parts management, however, is sparse (Topan et al. 2020). Operational spare parts management concerns the day-to-day and even hour-to-hour inventory management of spare parts, and can be found in almost any company with maintenance service operations. It is closely related to tactical spare parts management, but faces different challenges (Topan and van der Heijden 2020): (1) Operational decisions influence short-term rather than steady-state performance, (2) various types of real-time information are available when taking operational decisions, such as information on current inventory levels, and (3) strategic and tactical decisions can be considered as given when taking operational decisions. The characteristics of operational spare parts management make identifying good decisions challenging, perhaps even more so than for tactical spare parts management (Basten and van Houtum 2023).

Our research was initiated when we were asked to work on the operational spare parts planning system of ASML, a leading producer of lithography machines for the semiconductor industry. Lithography machines are a bottleneck resource in semiconductor fabrication plants due to their high purchase prices, which often exceed €100mn. Customers therefore require high uptime on these machines, making operational spare parts planning a key competency for ASML. The team responsible for the spare parts inventory management consists of hundreds of supply chain planners globally, who are supported by increasingly

sophisticated algorithms. These algorithms are the result of more than a decade of continuous research into spare parts inventory management at ASML, as evidenced by Kranenburg (2006), Vliegen (2009), Van Aspert (2015), Bakker (2016) and Lamghari-Idrissi (2021). On the operational planning level, this research culminated in the currently-used Network-Oriented Replenishment Automation (NORA) algorithm.

The goal of ASML's operational spare parts planning is to ensure the timely availability of spare parts at minimum cost, using a combination of replenishment and transshipment decisions. Replenishment decisions involve prioritizing the replenishment of one local warehouse over the other, whereas transshipment decisions involve proactively shipping from one local warehouse to another to alleviate shortages. These decisions are complicated by the scale of ASML's service network, which contains in the order of 10,000 Stock Keeping Units (SKUs) and 60 locations worldwide. We develop a model to take joint replenishment and transshipments decisions for this large network in near real-time, with the objective of minimizing transportation costs and stockout costs, and thereby outperforming NORA.

Many OEMs face operational spare parts inventory problems similar to ASML's, often complicated by their large-scale. For example, Topan et al. (2020) collaborated with more than ten such companies. Identifying good decisions for large-scale operational spare parts systems, however, is not straightforward, especially within an operational time frame. The number of feasible actions grows rapidly in the number of locations and SKUs, and evaluating actions is computationally costly due to demand stochasticity and the problem's multi-period horizon. The state of the art therefore resorts to heuristics and business rules. However, these heuristics and business rules typically consider only a limited number of actions and are evaluated on only a single inventory system (Topan et al. 2020), which may lead to poor or nongeneralizable performance. The impact can be considerable, as a wrong decision can lead to downtime at the customer, who could then incur extremely high costs.

Deep Reinforcement Learning (DRL) is a promising technique that could theoretically identify cost-minimizing decisions for multi-period inventory problems in near real-time. DRL is a special type of machine learning that is geared towards sequential decision-making problems. It differs from traditional optimization approaches in that it trains a model to efficiently map input states to cost-minimizing output decisions. This training process might be computationally-intensive, but a trained model can be re-used during

many operational planning cycles. Moreover, DRL models have already been successfully applied to multi-location (Gijsbrechts et al. 2022, Vanvuchelen et al. 2024) and multi-SKU (Madeka et al. 2022, Qi et al. 2023) inventory problems.

However, spare parts networks are characterized by their scale, and training DRL models for industrial-scale inventory systems remains an open challenge (Boute et al. 2022, Rolf et al. 2023). For example, Madeka et al. (2022) and Qi et al. (2023) are able to handle thousands of SKUs, but restrict themselves to a single-location problem. Gijsbrechts et al. (2022) and Vanvuchelen et al. (2024) train their DRL models for a distribution network of eleven locations and a fully connected network of ten locations, respectively, but only consider a single SKU. New techniques are hence necessary to scale DRL for industrial-scale inventory systems with both thousands of SKUs and several dozens of locations, such as ASML's spare parts network.

We leverage (spare parts) inventory management domain characteristics to train DRL models for industrial-scale spare parts networks. To this end, we propose a combination of three techniques: (1) We present a new method to standardize a DRL model's input and output space for multi-location inventory problems, such that it can be trained for an arbitrary number of SKUs using *global learning*. (2) We present a new *action space decomposition* approach to decompose the action space into a sequence of per-shipment subdecisions, such that it grows linearly rather than exponentially in the number of locations. (3) We propose a new *reward smoothing* approach and algorithm to address demand stochasticity and sparsity inherent to operational spare parts management problems, and are thereby able to speed up DRL model training. More specifically, we augment the reward function such that variance in rewards is reduced.

Our contribution is twofold. First, we establish the effectiveness of DRL for operational spare parts inventory management. We demonstrate the strong performance and practical applicability of our approach on a case study of ASML's spare parts network, where we find that it outperforms both the currently-used NORA algorithm and other existing heuristics. Moreover, our analysis of the model's proposed actions offers insights into how our DRL model is able to outperform these heuristics, and can serve as a starting point for the future development and improvement of other methods.

Second, we contribute to literature on the scalability of DRL models for inventory management. Our experiments demonstrate that our scaling methods allow us to train a DRL

model for a fully connected inventory network with 10,000 SKUs and 60 locations, thereby improving upon the current state-of-the-art. Moreover, the underlying ideas of reward smoothing to address stochasticity and action space decomposition to simplify the action space may be transferred to the wider field of operations management.

The paper proceeds as follows. In Section 2, we summarize key characteristics of operational spare parts management, and review methods for scaling DRL applied to inventory management. Next, we formalize our operational spare parts management problem in Section 3, and present our approach to scaling DRL for operational spare parts management in Section 4. In Section 5, we demonstrate the scalability and effectiveness of our DRL approach by benchmarking it against the currently-used NORA algorithm and other existing approaches on spare parts networks of varying sizes, and obtain insights into our approach's performance. We validate our approach on historical ASML data together with a domain knowledge expert from ASML and obtain further insights in Section 6. Finally, we summarize our findings and provide future research directions in Section 7.

2. Literature Review

As our work contributes to the literature on spare parts inventory management and scalable deep reinforcement learning, we review both streams. Section 2.1 provides an overview of spare parts management literature, including both its characteristics and optimization. Section 2.2 discusses different approaches to scale DRL for inventory management, and relates these approaches to our contributions.

2.1. Spare Parts Inventory Management

Spare parts management possesses several characteristics that sets it apart from other inventory problems. We describe key characteristics of spare parts inventory problems, and review methods to optimize them.

Characteristics: Demand for spare parts is usually assumed to follow a Poisson distribution. This assumption is reasonable when component failures are exponentially distributed, or when a location serves a sufficiently large number of machines (Van Houtum and Kranenburg 2015). Moreover, demand rates are typically low, as parts with high failure rates get redesigned (Ge et al. 2020, Akkermans et al. 2024). Demand for spare parts can hence be characterized as both stochastic and sparse, which complicates numerical evaluation of decisions.

When demand cannot be fulfilled locally, spare parts networks usually allow sourcing from other warehouses through so-called lateral transshipments (Basten and van Houtum 2023). When no stock is available at these other warehouses, demand is often assumed to be met from another warehouse through an emergency shipment. Transshipments can be categorized into proactive and reactive transshipments (Topan et al. 2020). Proactive transshipments occur in anticipation of future demand, whereas reactive transshipments are used to meet demand that has already occurred.

Spare parts networks often contain a large number of SKUs and locations. For example, Howard et al. (2015) examine a 900-location system, Van der Auweraer and Boute (2019) look at 40,000 SKUs, and Topan and van der Heijden (2020) consider a system with both 6,000 SKUs and 30 locations. As the number of locations and SKUs grows, the sets of possible states and feasible actions grows accordingly.

In sum, spare parts inventory management possesses several characteristics that make it challenging to optimize. A successful optimization method needs to be able to handle stochastic and sparse demand, a large number of feasible actions and a large state space.

Optimization: The characteristics of spare parts inventory problems have led to the development of specialized methods to efficiently address them. We discuss methods for related multi-location (spare parts) problems on both a tactical and operational level.

Research on tactical spare parts inventory management typically seeks to optimize stocking parameters, such as basestock levels. Sherbrooke (1968) examines a two-echelon distribution system with backordering. Axsäter (1990) extends their analysis to a two-echelon system with both reactive lateral transshipments and backordering. Alfredsson and Verrijdt (1999) replace backordering with reactive emergency shipments from a central warehouse. Kukreja et al. (2001) study a single-echelon version of the problem studied by Axsäter (1990). Finally, Kranenburg and van Houtum (2009) study a single-echelon system with fixed-order reactive lateral transshipments and emergency shipments from a central warehouse.

A recurring theme in each of these papers is the complexity of the investigated problem, leading to the development of an approximate evaluation procedure to identify near-optimal basestock levels. Operational spare parts management methods often deviate from the static replenishment and transshipment policies assumed on the tactical level, such as the First-Come-First-Serve (FCFS) replenishment assumed in each of the aforementioned

papers. We review literature on operational (spare parts) replenishment and transshipment policies, and thereby highlight differences with tactical assumptions.

Literature on operational replenishment policies has a long history. Miller (1974) proposes a look-ahead policy based on the expected number of backorders. Federgruen and Zipkin (1984) present a myopic replenishment policy for locations with normally distributed demand. Büyükkurt and Parlar (1993) use simulation to compare the performance of the highest-backorder, FCFS and origin-based replenishment decision rules. Caggiano et al. (2006) introduce a series of mathematical programming formulations and solve them using linear programming and greedy algorithms. Marklund and Rosling (2012) analyze a relaxation of the replenishment problem, and derive a ship-up-to-S heuristic that optimizes the relaxed problem. These papers improve our understanding of what makes a good replenishment policy. For problems with transshipments, however, the quality of the replenishment policy depends on the transshipment policy and vice versa.

Several articles investigate operational policies for proactive and reactive transshipments (Lee et al. 2007, Paterson et al. 2012, Seidscher and Minner 2013, Topan and van der Heijden 2020, Gerrits et al. 2022). Earlier inventory literature focuses on the analysis and development of heuristics for stylized single-echelon problems with transshipments, and establishes the added value of proactive and reactive transshipments (Lee et al. 2007, Paterson et al. 2012, Seidscher and Minner 2013). Inspired by these promising results, later spare parts literature develops problem-specific heuristic methods for practice-inspired two-echelon problems that include both replenishment and transshipment decisions (Topan and van der Heijden 2020, Gerrits et al. 2022). However, the development of generalizable joint decision-making methods for large-scale systems remains an open challenge (Topan et al. 2020).

We build upon operational spare parts management literature by proposing a generalizable and well-performing approach to take joint replenishment and transshipment decisions. The potential of DRL as a well-performing general-purpose technique for inventory management has already been established by Gijsbrechts et al. (2022). We show how it can be industrialized for operational spare parts management by leveraging knowledge of domain-wide characteristics, such as the large number of locations and SKUs.

2.2. Scaling Deep Reinforcement Learning

Deep Reinforcement Learning (DRL) problems are modelled using finite Markov Decision Processes (MDPs). These MDPs consist of a state space, an action space, a reward function, transition dynamics between states and, if applicable, a discount factor for future costs. The way in which an MDP is formulated can have a profound impact on the scalability of DRL algorithms. We therefore review how inventory management domain knowledge on reward functions, state spaces and action spaces has been used to scale DRL, and explain how our method builds on and differs from this earlier literature.

State space: The size and representation of the state space considerably impact how much training a DRL model requires to obtain a well-performing policy. As the state space can grow quickly in the number of SKUs and locations, recent literature has examined leveraging similarities between SKUs to reduce the required training time per SKU.

Global models are models trained on data from multiple SKUs, whereby SKU-specific features are included in the model input to enable SKU-specific predictions. These models have shown strong empirical performance over many different studies in the forecasting domain (Petropoulos et al. 2022, Wellens et al. 2023). Conceptually, they work by learning cross-SKU patterns in both the feature and label data. Cross-learning can achieve superior forecasting results by fully utilizing available data (Makridakis et al. 2022a,b), while simultaneously reducing computational costs (Makridakis et al. 2022a).

Global models have also been used for single-location inventory decisions. Qi et al. (2023) train a single global supervised learning model for a dataset of 24,333 SKUs. Madeka et al. (2022) build a single global DRL model for a dataset of 80,000 SKUs. They leverage symmetries between SKUs by using standardized output actions during prediction, which are subsequently multiplied by an SKU-specific scaling factor to obtain order quantities. However, such an SKU-based approach does not work for multi-location systems where the scale of demand does not only vary between SKUs, but also between locations.

We extend the application of global learning to the multi-location problem of operational spare parts management. Global learning relies on standardization of model input and output, which we achieve through decomposition of the (output) action space and careful (input) feature engineering.

Action space: The size and composition of the action space can also considerably impact the ability of DRL algorithms to converge to a well-performing policy. As the action

space for inventory problems often quickly grows in the number of SKUs and locations, previous works have examined means to reduce its size by masking undesirable actions and decomposing it into subdecisions.

The size of the action space can be reduced to enable faster convergence to a well-performing policy. Vanvuchelen et al. (2023) and Akkerman et al. (2024) use a continuous action representation to reduce their action space to a single decision per SKU, after which they map their DRL model's real number outputs to feasible integer order sizes. Van Hezewijk et al. (2022) use an eligibility mask to hide infeasible and low-quality actions from their DRL model.

Alternatively, faster convergence can be achieved by decomposing the action space. If a decision consists of multiple subdecisions, these subdecisions can be taken in parallel or sequentially instead of simultaneously. For example, Kaynov et al. (2024) use a multi-discrete action representation such that their action space grows linearly in the number of locations, instead of exponentially. Van Hezewijk et al. (2024) decompose decisions into a set of sequential per-SKU subdecisions. Oroojlooyjadid et al. (2022) and Van Dijck et al. (2024) decompose decisions into a set of sequential per-location subdecisions.

We go two steps further and decompose decisions into a set of per-shipment subdecisions, which we subject to a problem-specific eligibility mask to enable even faster convergence to a well-performing policy.

Reward function: The choice of reward function significantly affects DRL model training times. When inventory problems have stochastic and infrequent demand, their rewards are also often stochastic and infrequent. Highly stochastic rewards make it difficult for (deep) reinforcement learning algorithms to evaluate the quality of actions (Trimponias and Dietterich 2023). Moreover, sparse reward functions with infrequent non-zero rewards make it hard to identify actions that lead to state improvements (Sutton and Barto 2018).

Reward shaping is the process of artificially altering or augmenting a reward function. If applied properly, it can speed up training while still allowing for convergence to the optimal solution (Ng et al. 1999). It was applied to inventory management by Oroojlooyjadid et al. (2022) and De Moor et al. (2022). Oroojlooyjadid et al. (2022) apply DRL to the beer game and use reward shaping to align location-dependent rewards with the total reward of the supply chain. De Moor et al. (2022) augment their reward function with terms that

measure action dissimilarity from a teacher heuristic, which enables the DRL model to learn from this heuristic and improve upon it.

Another concept related to reward functions is that of variance reduction, which aims to reduce stochasticity in rewards. It is hence especially effective for highly stochastic problems. For example, the DRL algorithm by Temizöz et al. (2023) was developed specifically for stochastic inventory management and uses two types of variance reduction. First, it uses a roll-out policy that simulates the consequences of multiple different actions on the same starting state to identify the best action. Second, it uses a common random number approach such that the sequence of simulated events is the same for all roll-outs.

We introduce reward smoothing to reduce reward variance. More specifically, we shape the reward function such that it returns an estimate of the expected reward, instead of a realization of the reward.

3. Problem Formulation

Consider the following operational spare parts inventory management problem. Let $I := \{0, 1, ..., m-1\}$ denote the set of all SKUs, and let $J := \{0, 1, ..., n-1\}$ denote the set of all locations. For each location-SKU pair $(i, j), i \in I, j \in J$, we have a fixed basestock level $S_{i,j}$ that is determined and optimized at the tactical planning level. Stock is replenished one-for-one at the central warehouse $0 \in J$ after replenishment leadtime $t_i^{\text{repl}} > 0, i \in I$, which we assume to be constant and known. Consequently, for each SKU $i \in I$, the sum of inventory positions over all locations in the system $\sum_{j \in J} IP_{i,j}$ always equals the sum of the locations' basestock levels $\sum_{j \in J} S_{i,j}$.

Demand for an SKU i at a location j is assumed to arrive according to a Poisson process with rate $\lambda_{i,j} \geq 0$. If demand arrives at location j and cannot be met from the warehouse's local inventory, it is automatically met using a reactive emergency shipment from the first location k with available stock in emergency sourcing sequence $\sigma(j)$ at a cost of $c_{k,j}^{\text{em}} > 0$. Emergency sourcing sequences $\sigma(j) := (\sigma_0(j), \sigma_1(j), ..., \sigma_{n-2}(j))$ are set on the tactical planning level based on the expected emergency leadtime to location j. In case of a system-wide stockout, we assume stock is emergency-sourced from a factory near the Central Warehouse (CW) at a cost of $c_{0,j}^{\text{em}}$. Figure 1 visualizes a small example system with one SKU, $i = 0$, and $|J| = 6$ locations, namely one CW and five Local Warehouses (LWs).

The goal of the operational planning is to proactively ship stock in such a way that costs are minimized. To this end, stock can be proactively shipped from any location $j \in J$ to any

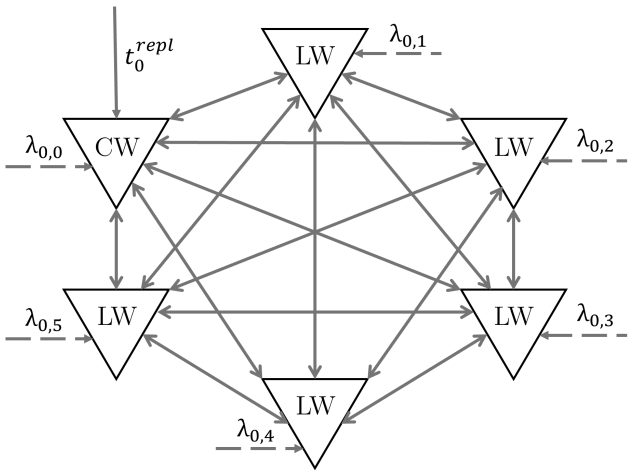

Figure 1 **Example of a fully connected spare parts network with $|I| = 1$ SKU and $|J| = 6$ locations**

other location $k \in J \setminus \{0, j\}$ using a routine shipment at a constant cost of $0 < c_{j,k}^{\text{rout}} < c_{j,k}^{\text{em}}$, and with a constant leadtime of $t_{j,k}^{\text{rout}} > 0$.

We formulate our cost minimization objective function over a (possibly infinite) time horizon T. Let $\mathbb{E}\left[x_{j,k,t}^{\text{rout},\pi}\right]$ be the number of units of SKU $i \in I$ proactively shipped from $j \in J$ to $k \in J \setminus \{0, j\}$ at the start of period $t \in \{1, \ldots, T\}$ under policy $\pi \in \Pi$, where $x_{j,k,t}^{\text{rout},\pi}$ is a decision variable. Let $\mathbb{E}\left[x_{j,k,t}^{\text{em},\pi}\right]$ be the expected number of emergency-sourced units of SKU $i \in I$ from j to k during period t under policy π, where the number of emergency shipments $x_{j,k,t}^{\text{em},\pi}$ is a consequence of earlier proactive shipments $x_{j,k,t'}^{\text{rout},\pi}, t' \leq t$ and demand realizations. The objective function is then given by:

$$\min_{\pi \in \Pi} \sum_{t \in \{1,\ldots,T\}} \sum_{i \in I} \sum_{j \in J} \sum_{k \in J \setminus \{j\}} \left(c_{j,k}^{\text{rout}} \mathbb{E}\left[x_{i,j,k,t}^{\text{rout},\pi}\right] + c_{j,k}^{\text{em}} \mathbb{E}\left[x_{i,j,k,t}^{\text{em},\pi}\right] \right) \tag{1}$$

To formalize the balance equation, we introduce variables $I_{i,k,t}$ and $D_{i,k,t}$. Let $I_{i,k,t}$ denote the inventory of SKU $i \in I$ in warehouse $j \in J$ at the start of period t after shipments for the period have arrived. Let $D_{i,k,t} \sim \text{Poisson}(\lambda_{i,k})$ denote the demand for the same SKU i and warehouse k during period t. The balance equation is then given by:

$$I_{i,k,t} = \begin{cases} I_{i,k,t-1} + \sum_{j \in J \setminus \{k\}} \left(x_{i,j,k,t-t_{j,k}^{\text{rout}}}^{\text{rout},\pi} + x_{i,j,k,t}^{\text{em},\pi} - x_{i,k,j,t}^{\text{rout},\pi} - x_{i,k,j,t}^{\text{em},\pi} \right) - D_{i,k,t}, & k > 0 \\ I_{i,k,t-1} + \sum_{j \in J \setminus \{k\}} \left(x_{i,j,k,t}^{\text{em},\pi} - x_{i,k,j,t}^{\text{rout},\pi} - x_{i,k,j,t}^{\text{em},\pi} \right) + \sum_{j \in J} \left(D_{i,j,t-t_i^{\text{repl}}} \right) - D_{i,k,t} & k = 0 \\ + \left[\sum_{j \in J} \left(I_{i,k,t} - D_{i,k,t} \right) \right]^+ - \left[\sum_{j \in J} \left(I_{i,k,t-t_i^{\text{repl}}} - D_{i,k,t-t_i^{\text{repl}}} \right) \right]^+, & \end{cases} \tag{2}$$

whereby the terms for local warehouses $k > 0$ correspond to leftover inventory, net inventory from transshipments and demand respectively. The terms for central warehouse $k = 0$ correspond to leftover inventory, net inventory from lateral transshipments, replenishments, demand and net emergency shipments due to system-wide stockouts, respectively.

4. Methodology

We are now ready to formalize our problem as a Markov decision process and explain how we scale DRL for industrial-scale operational spare parts management. Building on the transition dynamics in Section 3, Section 4.1 discusses our state space representation and our approach to enable global learning. In Section 4.2, we define our decomposed action space. We present the reward function and explain how we apply reward smoothing for variance reduction in Section 4.3. Finally, we discuss and motivate our choice of deep reinforcement learning algorithm in Section 4.4.

4.1. State Space and Global Learning

We represent our state space in line with the principles of global learning to train DRL models for any number of SKUs. Spare parts networks can contain tens of thousands of SKUs. Training one model for each SKU is possible, but training tens of thousands of such SKU-dependent models would require a significant amount of computational resources, especially if these models need to be retrained each time demand rate forecasts change. We therefore train a single global model to take decisions for all SKUs, regardless of demand rate or basestock level.

For a global model to efficiently learn cross-SKU patterns, both the model input and output need to be standardized. Previous applications of global learning to inventory management only considered single-location problems, which allows them to simply scale input features and output actions based on (average) historic demand for that single location. This approach cannot be applied to multi-location problems due to interactions between the different locations. To know whether we should replenish a location, we not only need to know its stockout risk, but also the probability that we can source from nearby locations. Moreover, as the scale of demand and inventory differs between warehouse-SKU pairs, we cannot scale output actions for the different shipping lanes and SKUs by the same number.

We solve this problem by introducing a new way to standardize our model's input and output. Standardization of the output space, or action space, is discussed in the next

section. We standardize the input space, or feature space, by expressing inventory levels in terms of fillrates under locally generated demand. In particular, let $\beta_{i,j}$ be the fillrate for SKU $i \in I$ at location $j \in J$ under locally generated demand, $I_{i,j}$ the corresponding inventory on-hand and $\lambda_{i,j}$ the corresponding demand rate. $\beta_{i,j}$ can then be calculated as follows (see Appendix A):

$$\beta_{i,j}(I_{i,j}) = \frac{I_{i,j}}{\lambda_{i,j}} - e^{-\lambda_{i,j}} \sum_{x=0}^{I_{i,j}-1} \frac{(I_{i,j}-x)\lambda_{i,j}^{x-1}}{x!} \tag{3}$$

Expressing inventory levels in terms of fillrates helps the model recognize that locations with lower fillrates are generally at risk, whereas locations with higher fillrates could be used to alleviate this risk. Nonetheless, providing only the fillrates for the current inventory levels to the DRL model is insufficient, as the impact of shipment decisions is different for low-demand locations and SKUs than for high-demand locations and SKUs. We therefore also include the fillrates for inventory levels that are one unit higher or lower in the model input features.

The state as observed by the DRL model is given by the input features in Table 1. Aside from the fillrate-based features, we use replenishment leadtime and inventory level features as we find they improve performance. We also add transportation costs and leadtimes for each potential receiving location when deciding on which one to ship to. These features allow our model to account for distance from the sourcing location, and are set to 0 when selecting a location to ship from.

4.2. Action Space and Decomposition

We decompose the action space to limit its growth and facilitate global learning. Any location $j \in J$ can send any unit of its inventory I_j to any other location $k \in J \setminus \{0, j\}$ at a given moment. A classical single-decision formulation of the action space would thus allow for $\mathcal{O}\left(\prod_{j \in J} |J|^{I_j}\right)$ possible actions, leading to prohibitive growth of the action space in the number of locations. As spare parts networks often contain many locations, measures are necessary to keep the action space manageable. We therefore split decisions into multiple consecutive subdecisions, such that the number of actions per decision grows only linearly in the number of locations.

Specifically, we model the decision process for each decision period $t \in \{1, ..., T\}$ as a two-step cycle of subdecisions. In the first step, our DRL model can either choose a location

Feature type	Subtype	Amount
Replenishment leadtime	-	1
Inventory level	Inventory on hand	$\|J\|$
Inventory level	Inventory position	$\|J\|$
Fillrate	Inventory on hand - 1	$\|J\|$
Fillrate	Inventory on hand	$\|J\|$
Fillrate	Inventory position	$\|J\|$
Fillrate	Inventory position + 1	$\|J\|$
Transportation costs	Routine costs	$\|J\|$
Transportation leadtimes	Routine leadtimes	$\|J\|$

Table 1 Input features for the DRL model

$j \in J$ with $I_j > 0$ to source from, or choose to not send anymore shipments for the given decision period t. If a location j is chosen to source from, the DRL model proceeds to the second step, where it can choose a location $k \in J \setminus \{0, j\}$ with inventory position $IP_k < S_k$ to ship towards. Once k is selected, a shipment is sent from j to k and the state of the MDP is updated accordingly. The decision cycle then returns to the first step, where the decision cycle can be repeated for however many shipments need to be made.

Modelling the decision process as a two-step cycle speeds up the learning process by facilitating cross-learning between actions. For example, if we learn that shipping towards a certain location from a random sample of other locations is a poor idea, exploring actions where we ship towards this location from other locations is unlikely to be worth it. Moreover, our model's ability to converge is unaffected, as all possible actions can still be reached through combinations of subdecisions.

We formalize the action space and the constraints thereupon as follows. Let $\mathcal{A} = \{0, 1, ..., 2|J|\}$ be the set of possible actions. Let actions $\{0, 1, ..., |J| - 1\}$ represent the decisions to ship towards locations $0, 1, ..., |J| - 1$, respectively. Define action $|J|$ as the decision to send no further shipments in the current period. Finally, let actions $\{|J| + 1, |J| + 2, ..., 2|J|\}$ denote the decisions to source from locations $0, 1, ..., |J| - 1$, respectively. Consequently, the number of possible actions per subdecision is linear in the number of locations, with at most at most $|J| + 1$ possible actions per subdecision.

The action space is subject to an eligibility mask to ensure feasibility and minimize unnecessary exploration. A shipping action $a \in \{0, 1, ..., |J| - 1\}$ is feasible when its inventory position IP_a is below its basestock level S_a, and a is not the same as the sourcing location. Stopping action $|J|$ is only allowed when there is no overstocking at the replenishment location, i.e., if $I_0 \leq S_0$. A sourcing action $a \in \{|J| + 1, |J| + 2, ..., 2|J|\}$ is feasible if the sourcing location has non-zero stock, i.e., $I_{a-|J|-1} > 0$, at least one location $k : k \neq a - |J| - 1$ exists with $IP_k < S_k$, and either the replenishment location is not overstocked, or a corresponds to sourcing from the replenishment location (i.e., $a = 0$).

4.3. Reward Function and Smoothing

We propose an alternative reward function formulation to reduce reward variance for stochastic and sparse inventory problems. For inventory problems, the reward function is usually defined as the negative of the cost function. Demand for spare parts, as discussed in Section 2.1, is typically both stochastic and sparse. As costs follow from demands that cannot be met, costs and hence rewards for spare parts systems are also typically both stochastic and sparse. This stochasticity and sparsity makes it more difficult to train a well-performing DRL model (Sutton and Barto 2018).

The key idea behind our approach is to address stochasticity and sparsity by smoothening the reward function. Cost $C^{\pi}_{i,t} = \sum_{j \in J} \sum_{k \in J \setminus \{j\}} \left(c^{\text{rout}}_{j,k} x^{\text{rout},\pi}_{i,j,k,t} + c^{\text{em}}_{j,k} x^{\text{em},\pi}_{i,j,k,t} \right)$ for SKU i in period t under policy π is stochastic and sparse. If we are able to calculate expected cost $\mathbb{E}[C^{\pi}_{i,t}]$ and use it instead of $C^{\pi}_{i,t}$, we could eliminate stochasticity and sparsity of $C^{\pi}_{i,t}$ without affecting our model's ability to converge to the optimal policy. We formalize these statements in Proposition 1, and prove them in Appendix B.

PROPOSITION 1. *Let $C^{\pi}_{i,t}$ be the cost for SKU i in period t under policy π, $\mathbb{E}[C^{\pi}_{i,t}]$ its expectation with respect to the demand, and assume a non-zero system-wide stockout rate. Then (1) minimizing the expected long-run discounted smoothed cost $\sum_{t=1}^{T} \gamma^t \mathbb{E}[C^{\pi}_{i,t}]$ is equivalent to minimizing the expected long-run discounted cost $\sum_{t=1}^{T} \gamma^t C^{\pi}_{i,t}$, (2) sparsity in rewards is eliminated as $\mathbb{E}[C^{\pi}_{i,t}] > 0$, and (3) the variance of $\mathbb{E}[C^{\pi}_{i,t}]$ is strictly less than that of $C^{\pi}_{i,t}$.*

Unfortunately, calculating $\mathbb{E}[C^{\pi}_{i,t}]$ is computationally too heavy to make the reduction in stochasticity and sparsity worth the effort. However, several fast and accurate cost approximation algorithms exist for tactical spare parts management. In line with this

literature, we propose to use estimated expected costs $\hat{\mathbb{E}}[C^{\pi}_{i,t}]$. We introduce an approximate evaluation algorithm to obtain $\hat{\mathbb{E}}[C^{\pi}_{i,t}]$, which is inspired by the algorithm by Kranenburg and van Houtum (2009). Their algorithm uses the Poisson distribution to approximate the emergency demand flow from out-of-stock locations to emergency sourcing locations for a steady-state system. We propose Algorithm 1, which does the same for a system with a single-period horizon.

Our algorithm consists of three parts. First, the cost estimate is initialized using the (exact) costs for routine shipments in the first line of the algorithm. Second, it estimates the expected outgoing emergency demand for a given SKU i from all locations $j \in J$ in line 3. This estimate $\widehat{\mathbb{E}}[O]$ is obtained by multiplying the expected locally generated demand $\lambda_{i,j}$ by the probability $(1 - \beta_{i,j}(I_{i,j}))$ that locally generated demand has already depleted the inventory, whereby $\beta_{i,j}(I_{i,j})$ is calculated using Equation (3).

Third, the algorithm estimates from where overflowing emergency demand is met for each location $j \in J$, and stores the associated costs in $\widehat{\mathbb{E}}[C^{\pi}_{i,t}]$ (lines 4-9). The probability with which a location $l \in \sigma(j)$ is able to meet incoming overflow demand from a location $j \in J$ is estimated using $\hat{\beta}_{i,j,l} = \beta_{i,j}(P)\beta_{i,l}(I_{i,l})$, which we base on the approximation $\beta_{i,j,l} \approx \beta_{i,j}(P)\beta_{i,l}(I_{i,l})$. In this equation, $\beta_{i,j,l}$ is the fillrate encountered by overflow demand from location j at location l, and P is the amount of stock that needs to be depleted for demand at location j to encounter a stockout at location l. The algorithm finishes by assuming that overflow demand that could not be met from any location is fulfilled from a factory near central warehouse 0 at a cost of $c^{\text{em}}_{0,j}$, and updates $\widehat{\mathbb{E}}[C^{\pi}_{i,t}]$ accordingly (line 10).

4.4. Choice of Algorithm

The recent popularity of DRL has led to many applications of DRL algorithms to inventory management problems (Boute et al. 2022). Value-based algorithms such as Deep Q-Networks (DQNs, Mnih et al. 2013) were applied to inventory management problems by Oroojlooyjadid et al. (2022) and De Moor et al. (2022). Gijsbrechts et al. (2022), Van Hezewijk et al. (2022) and Vanvuchelen et al. (2023) applied actor-critic algorithms such as A3C (Mnih et al. 2016) and Proximal Policy Optimization (PPO, Schulman et al. 2017). However, the stochasticity in reward and state transitions inherent to many inventory management problems makes it difficult for these general-purpose algorithms to converge to well-performing policies.

Algorithm 1 Approximate Evaluation Algorithm

Initialize $\widehat{\mathbb{E}}[C^{\pi}_{i,t}] \leftarrow \sum_{j \in J} \sum_{k \in J \setminus \{j\}} c^{\text{rout}}_{j,k} x^{\text{rout},\pi}_{i,j,k,t}$
for all $j \in J$ **do**
    Initialize $\widehat{\mathbb{E}}[O] \leftarrow (1 - \beta_{i,j}(I_{i,j}))\lambda_{i,j}$
    Initialize $P \leftarrow I_{i,j}$
    for all $l \in \sigma(j)$ **do**
        Update $P \leftarrow P + I_{i,l}$
        Estimate $\hat{\beta}_{i,j,l} \leftarrow \beta_{i,j}(P)\beta_{i,l}(I_{i,l})$
        Update $\widehat{\mathbb{E}}[C^{\pi}_{i,t}] \leftarrow \widehat{\mathbb{E}}[C^{\pi}_{i,t}] + \widehat{\mathbb{E}}[O]\hat{\beta}_{i,j,l} c^{\text{em}}_{l,j}$
        Update $\widehat{\mathbb{E}}[O] \leftarrow (1 - \hat{\beta}_{i,l})\widehat{\mathbb{E}}[O]$
    Update $\widehat{\mathbb{E}}[C^{\pi}_{i,t}] \leftarrow \widehat{\mathbb{E}}[C^{\pi}_{i,t}] + \widehat{\mathbb{E}}[O] c^{\text{em}}_{0,j}$
Return $\widehat{\mathbb{E}}[C^{\pi}_{i,t}]$

To address this stochasticity, Temizöz et al. (2023) propose the variance-reducing Deep Controlled Learning (DCL) algorithm discussed in Section 2.2. They find their algorithm outperforms PPO on the lost sales problem, and Van Hezewijk et al. (2024) reach the same conclusion for the capacitated lot sizing problem. Similarly, Verleijsdonk et al. (2024) and Van Dijck et al. (2024) obtain strong results for other stochastic operations management problems using the DCL algorithm. Considering the algorithm's theoretical underpinnings and promising results on various stochastic operations management problems, we adopt it as well.

5. Numerical Experiments

We demonstrate the scalability and effectiveness of our approach through numerical experiments on spare parts networks of varying sizes, which we base on ASML's service network. After describing our input parameters and experimental setup in Sections 5.1 and 5.2 respectively, we first establish the effectiveness of our reward smoothing and action space decomposition techniques. To this end, we compare DRL models with none, one or two of these techniques in Section 5.3. In Section 5.4, we show that our approach outperforms existing heuristics, including the currently-used NORA heuristic introduced in Section 1. Finally, we dive deeper into how DRL is able to outperform NORA in Section 5.5.

5.1. Input Parameters

To meet confidentiality requirements and ensure reproducibility, we generate a publicly shareable dataset that is representative of ASML's service network, as confirmed by ASML employees. ASML's service network contains in the order of 60 locations and 10,000 SKUs. We generate data for four distinct scenarios with a varying number of locations to examine the effectiveness of our scaling methods. More precisely, we generate scenarios with $|J| = 6, 12, 30, 60$ locations, all containing $|I| = 10,000$ SKUs.

For each scenario, we generate demand rates $\lambda_{i,j}$ and replenishment leadtimes t_i^{repl}. We obtain $\lambda_{i,j}$ by multiplying an SKU-dependent variable $\lambda^i \sim \text{LogUniform}(10^{-9}, 10^{-3})$ by a location-dependent variable $\lambda^j \sim \text{LogUniform}(1, 10^3)$. The SKU-dependent effect is among others related to the SKU's degradation rate, whereas the location-dependent part is among others related to the number of machines at a given location. We generate t_i^{repl} using the $\text{LogUniform}(20, 500)$ distribution.

A visualization of locations, regions and continents is given in Figure 2. We follow the traditional Americas, Europe-Middle-East-Africa (EMEA) and Asia-Pacific split present in many organisations, and generate locations such that they are spread over three continents. Each continent contains two regions, and each region contains an equal number of locations. The x- and y-coordinates of locations are generated uniformly at random over the area of a region, and used to calculate a Euclidian distance matrix. One location from the northern half of the central continent is randomly selected to serve as the central warehouse.

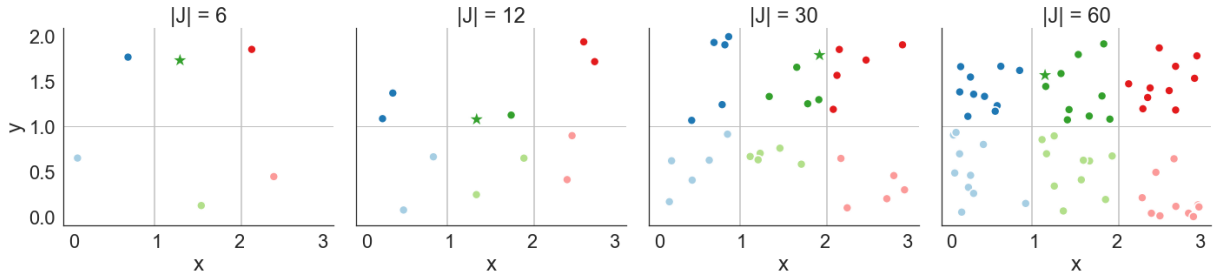

Figure 2 **Plot of locations (points), central warehouse (star), regions (brightness) and continents (colors)**

Emergency sourcing sequences $\sigma(j)$ are determined such that a location always sources from the nearest location with available stock, or the central warehouse in case of a system-wide stockout. Routine transportation leadtimes $t_{j,k}^{\text{rout}}$ between warehouses are obtained by

scaling the distances to the $[1, 14]$ days range. Transportation costs are obtained by multiplying the distances with a factor 100 for routine shipments ($c^{\text{rout}}_{j,k}$) and 300 for emergency shipments ($c^{\text{em}}_{j,k}$).

Basestock levels $S_{i,j}$ are obtained by applying the greedy algorithm described in Kranenburg and van Houtum (2009) to mimic the tactical planning. This algorithm identifies inventory levels that minimize holding costs, while still meeting a set of expected waiting time constraints under `FCFS` replenishment. Holding costs are assumed to be equal for all locations. Emergency shipment leadtimes are set at 25% of routine shipment leadtimes. If all locations are out-of-stock when a demand occurs, the part is assumed to be emergency-sourced through the central warehouse at a procurement leadtime penalty of 48 hours on top of the emergency shipment leadtimes. Expected waiting time constraints (in hours) are generated by multiplying a Uniform$(0.5, 2)$ location-based variable by a Uniform$(0.5, 2)$ SKU-based variable. Location-based variables represent contractual commitments, whereas SKU-based variables result from the trade-off between an SKU's investment cost and effect on expected downtime.

5.2. Experimental Setup and Benchmarks

We perform all experiments on a single node with 336 GiB of RAM and 2 AMD Genoa CPUs. Training and evaluation runs are performed on independently generated data, and hyperparameters are found after some trial and error (see Appendix C). We use the DCL algorithm implementation of the Dynaplex library (Akkerman et al. 2023). The DCL algorithm's training time is regulated by the number of samples to be collected, which is set to the minimum amount our best-performing DRL model needs to comfortably outperform the currently-used algorithm's performance.

To evaluate the performance of our DRL model relative to benchmarks, we randomly select 200 SKUs from our dataset and evaluate each method's performance using simulation. Simulation runs are performed as a continuous run with a warm-up length of 10,000 days to reach a realistic starting state, periods of length 1,000 days to be sufficiently independent of each other, and a total of 100 periods to obtain sufficiently small confidence intervals to prevent overlap in results between different policies. This set-up leads to a total simulation length of 110,000 days per SKU to evaluate the model.

Confidence intervals for the aggregate costs of the evaluated SKUs are constructed using the t-distribution. Similarly, we use a one-sided paired t-test to test whether differences in

aggregate costs between policies are significant, whereby we use the same demand realizations for the simulation run of each policy to obtain paired period samples.

We benchmark our DRL approach against three policies, namely a First-Come-First-Serve (`FCFS`) policy, an Average-Time-To-Stockout (`ATTS`) policy, and the currently-used Network-Oriented Replenishment Automation (`NORA`) algorithm. The `FCFS` policy replenishes locations in order of their demand, and its use is often assumed on the operational level by tactical planning methods. For example, the seminal papers by Sherbrooke (1968) and Kranenburg and van Houtum (2009) assume `FCFS` replenishment on the operational level to optimize basestock levels on a tactical level.

The `ATTS` policy prioritizes replenishment of locations with the shortest expected time until stock is depleted by demand, which can be obtained by dividing the current inventory position by the forecasted demand rate for each location. We include this policy as a benchmark as it is commonly used in practice, and was used at ASML before the `NORA` policy was introduced.

Lastly, the `NORA` policy prioritizes replenishment of locations with the highest estimated local stockout risk, while simultaneously accounting for the stockout risk on a regional and continental level. Minimizing local stockout risk is equivalent to minimizing the number of reactive emergency shipments, while minimizing regional and continental stockout risk ensures that reactive emergency shipments that do occur are as fast and cheap as possible. `NORA` was developed by Bakker (2016), after which it was finetuned in the following years using feedback from practice. It serves as the basis for ASML's current operational spare parts planning, and is expected to be a strong benchmark due to its development history.

`NORA` works as follows. Let $\hat{\mathbb{E}}[\text{NAV}]$ be the estimated amount of non-availabilities for the coming 14 days, whereby a Non-AVailability (NAV) is defined as a unit of demand that cannot be met directly from local stock. `NORA` first identifies the continent with the highest $\hat{\mathbb{E}}[\text{NAV}]$, after which it identifies the region with the highest $\hat{\mathbb{E}}[\text{NAV}]$ within this continent. Finally, it prioritizes replenishment of the location with the highest $\hat{\mathbb{E}}[\text{NAV}]$ within this region. Let IP denote the inventory position and λ the 14-day demand rate. $\hat{\mathbb{E}}[\text{NAV}]$ can then be obtained by multiplying the Erlang loss probability by λ:

$$\hat{\mathbb{E}}[\text{NAV}] = \frac{\frac{1}{IP!}\lambda^{IP}}{\sum_{j=0}^{IP}\frac{1}{j!}\lambda^{j}}\lambda. \tag{4}$$

To assess the effectiveness of the techniques that constitute our approach, we consider four different versions of our approach. The `DCL-RS-AS` policy uses global learning, reward smoothing and full action space decomposition. `DCL-AS` uses global learning and full action space decomposition, but no reward smoothing. `DCL-RS` uses global learning and reward smoothing, but only partially decomposes the action space. More specifically, it jointly selects the receiving and sourcing locations for shipments, instead of selecting them sequentially. This lesser degree of decomposition already goes further than earlier applications of action space decomposition (e.g., Oroojlooyjadid et al. 2022, Van Hezewijk et al. 2024), but we find that it is necessary even for smaller fully connected networks. Finally, `DCL` uses global learning and partial action space decomposition, like `DCL-RS`, but does not use reward smoothing.

5.3. Ablation Study Results

We establish the effectiveness of reward smoothing and action space decomposition by comparing DRL models with both techniques to DRL models with one or none of them. The results are presented in Table 2, whereby observed costs refer to the true costs as in Equation (1), and the smoothed costs refer to the cost estimates by Algorithm 1. We present our results in Table 2. For ease of interpretation, all results are standardized by dividing both the costs and Confidence Interval (CI) half-widths by `NORA`'s costs, such that `NORA` has a standardized cost of 1.000. All best-performing policies indicated in bold in Table 2 perform significantly better than the second-best-performing policies.

The table shows that both action space decomposition and reward smoothing are effective. For small to moderate sized networks, `DCL-RS-AS` is the best-performing policy. `DCL-AS` is the best-performing policy for the larger 60-location problem. We observe that action space decomposition is always beneficial, and that reward smoothing is beneficial for small to moderate sized networks.

The reason reward smoothing is less effective for larger networks is twofold. First, the reward smoothing algorithm is more accurate for smaller to moderate sized networks, as evidenced by the lower disparity between observed and smoothed costs for smaller to moderate sized networks. Second, to face larger problems, the number of training samples and rollouts performed by the DCL algorithm needs to increase accordingly. The resulting reduction in variance lowers the need for variance reduction through reward smoothing.

Policy	Number of warehouses							
	6		12		30		60	
	Observed costs							
DCL	0.881	(±0.011)	1.033	(±0.012)	1.000	(±0.006)	-	-
DCL-RS	0.902	(±0.013)	1.027	(±0.012)	0.999	(±0.006)	-	-
DCL-AS	0.829	(±0.013)	1.009	(±0.013)	0.993	(±0.006)	**0.962**	**(±0.004)**
DCL-RS-AS	**0.775**	**(±0.009)**	**0.962**	**(±0.013)**	**0.990**	**(±0.006)**	0.978	(±0.003)
	Smoothed costs							
DCL	0.880	(±0.011)	1.031	(±0.009)	1.000	(±0.006)	-	-
DCL-RS	0.901	(±0.012)	1.026	(±0.010)	0.999	(±0.005)	-	-
DCL-AS	0.829	(±0.012)	1.008	(±0.010)	0.993	(±0.006)	**0.957**	**(±0.002)**
DCL-RS-AS	**0.774**	**(±0.008)**	**0.961**	**(±0.011)**	**0.990**	**(±0.005)**	0.966	(±0.002)

Table 2 Standardized costs and CI half-widths for the ablation study experiments

Consequently, our reward smoothing approach is useful to speed up convergence to a well-performing policy for small to moderate sized networks, whereas an approach without reward smoothing performs better for larger 60-location problems.

5.4. Benchmark Study Results

We demonstrate the strong performance of our approach by benchmarking our best-performing DRL approach against the FCFS, ATTS and NORA policies. The results are provided in Table 3. Results are standardized as before, and all best-performing policies perform significantly better than the second-best-performing policies. The results show several interesting patterns.

First, NORA meets expectations and is able to significantly outperform the FCFS and ATTS policies for all network sizes, often by comfortable margins. The margins by which NORA outperforms benchmarks tend to be larger for smaller problem sizes, which may be explained by the higher average distance to warehouses for emergency sourcing. NORA accounts for regional and continental inventory levels, leading it to better spread out stock over its global service network.

Second, our DRL approach is able to significantly outperform NORA for all network sizes. Most importantly, it is able to outperform NORA by a 3.8% margin on the full-size network. Our scaling techniques hence enable us to train DRL models for industrial-scale networks

with up to 60 locations and 10,000 SKUs, thereby improving upon the state-of-the-art. Moreover, a 3.8% cost reduction would translate to considerable cost savings given the scale and impact of ASML's service network.

	Number of warehouses							
Policy	6		12		30		60	
FCFS	1.041	(±0.012)	1.333	(±0.023)	1.085	(±0.007)	1.097	(±0.004)
ATTS	1.313	(±0.023)	1.615	(±0.029)	1.185	(±0.008)	1.070	(±0.005)
NORA	1.000	(±0.010)	1.000	(±0.012)	1.000	(±0.006)	1.000	(±0.004)
DRL	**0.775**	**(±0.009)**	**0.962**	**(±0.013)**	**0.990**	**(±0.006)**	**0.962**	**(±0.004)**

Table 3 Standardized costs and CI half-widths for the benchmark study experiments

5.5. Explaining Performance

To investigate how DRL policies are able to outperform NORA, we examine the types of shipments proposed by NORA and DCL-AS for the 60-location problem. Figure 3 shows the number of replenishment, proactive and reactive shipments under both policies, divided by the total number of DCL-AS shipments to facilitate interpretation. Replenishment shipments refer to routine shipments from the central warehouse to a local warehouse. Proactive shipments refer to routine lateral transshipments between local warehouses. Reactive shipments refer to emergency shipments between any two warehouses, central or local.

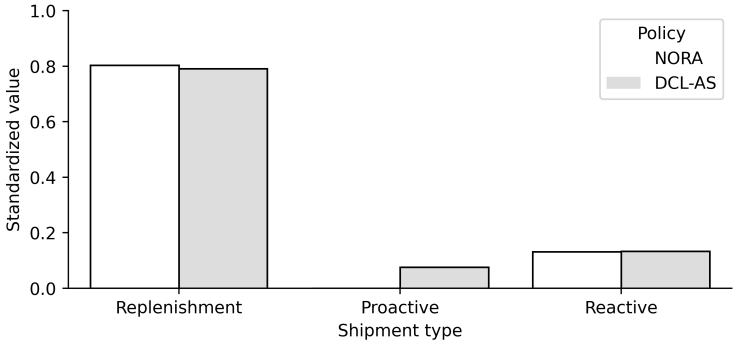

Figure 3 Standardized number of shipments for NORA and DCL-AS on the 60-location problem

Figure 3 shows that DCL-AS differs from NORA in the use of proactive lateral transshipments, whereas their number of replenishments and reactive shipments is approximately the

same. The small decrease in replenishments of `DCL-AS` compared to `NORA` can be explained by `DCL-AS` not always directly replenishing other locations whenever the central warehouse receives stock, unlike `NORA`. Interestingly, the increase in proactive shipments does not decrease the number of reactive shipments, which indicates that proactive shipments instead lead to reduced costs due to reduced reactive shipment waiting times. Indeed, we find that `DCL-AS` achieves a 4.59% reduction in average waiting times compared to `NORA`.

6. Application and Expert Validation

To demonstrate the practical applicability of our method and obtain insights into its performance, we also apply it to historical ASML data and review the resulting DRL policy together with a domain knowledge expert from ASML. More specifically, we train a DRL model on ASML data from the period November 2021 to July 2023, and use it to validate our approach both numerically and together with the expert.

Application to historical data: Although we are not able to share many results using real data due to confidentiality requirements, we can use it to validate our findings from Section 5. To this end, we extract ASML's basestock levels and forecasted demand rates, and evaluate both the `NORA` and `DRL` policy using simulation. The `DRL` model is trained on a local laptop with 32 GB of RAM and a 3.00 GHz Intel Core i7 vPro processor, and tuned using Optuna (Akiba et al. 2019). Similar to Section 5, we evaluate both policies on the same 100 randomly selected SKUs with a 10,000 day warm-up period, 100 simulation periods and 1,000 days per simulation period.

Our findings are in line with our numerical experiments on synthetic data. We observe cost savings of 5.1% for using `DRL` instead of `NORA`, compared to the 3.8% on the data used in Section 5. Moreover, as on our synthetic data, we observe `DRL`'s use of proactive lateral transshipments, relatively small changes to the number of replenishments and reactive shipments, and a reduction in average waiting times. We hence conclude that our results from Section 5 hold when applied to historical ASML data.

Expert validation: To manually review the `DRL` policy with an expert from ASML, we extract the state of ASML's operational spare parts inventory system for July 31st, 2023, and apply our trained DRL model to obtain shipment proposals for all SKUs and all locations. Each shipment proposal corresponds to a proposal to ship one unit of stock from one location to another. The 20 SKUs for which the shipments proposed by `DRL` and `NORA` differ the most are selected for manual review by the expert.

The expert observes three main patterns in the proposed shipments. First, in line with our earlier findings, the DRL policy regularly uses proactive lateral transshipments, unlike NORA. Second, the local warehouses from which these lateral transshipments are made are often geographically close to the receiving warehouse. Third, the DRL sometimes makes use of an expediting strategy, where proactive lateral transshipments are followed directly by a replenishment of the warehouse that sent the proactive lateral transshipment.

We illustrate our findings by discussing the proposed shipments for an SKU where we observe all three patterns. Figure 4 visualizes the proposed shipments for this SKU, whereby nodes represent locations and arrows represent the decision to send stock from one location to another. If shipment quantities are above one, they are indicated on the arrow. Warehouse locations are encoded due to confidentiality. The first part of the encoding refers to the warehouse's region, and the second part can be used to identify the warehouse itself. The central warehouse is encoded as CW.

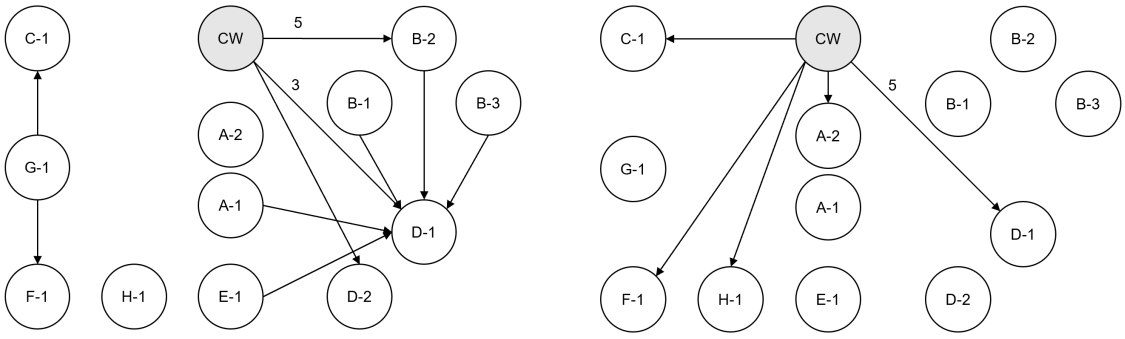

Figure 4 Comparison of shipments proposed by DRL (left) and by NORA (right)

The shipments display the observed patterns in the following ways. Proactive lateral shipments are proposed by the DRL policy from warehouse A-1, B-1, B-2, B-3 and E-1 to D-1, from G-1 to C-1 and from G-1 to F-1. Each of these locations are relatively close to each other. Moreover, the DRL policy expedites replenishment of D-1 by shipping from nearby B-2 to D-1 first, and then from the CW to B-2, whereas NORA ships directly from the CW to D-1. These shipments have clear benefits, as both the average-time-to-shortage and $\hat{\mathbb{E}}$[NAV] are significantly lower for the receiving warehouses than for the sending warehouses.

Nonetheless, the proposed shipments also have some disadvantages. DRL leaves locations B-1 and B-3 with low inventory. Sending four net units to location B-2 might be too much, even if it is a high-demand location. Finally, sourcing from region E would leave it without

any stock, although the $\hat{\mathbb{E}}$[NAV] of warehouse E-1 (and region E) is lower than that of warehouse D-1 (and region D) even after the shipment.

Overall, the expert concludes that the strategies of proactive transshipments, transshipments from nearby locations and expediting are sensible. They provide insight into why and how `DRL` is able to outperform `NORA`, and can serve as a starting point for the development of new heuristics.

7. Conclusion

Operational spare parts management requires fast decision-making for complex and large-scale networks. Previous work has therefore focused on computationally-light heuristics. Deep Reinforcement Learning (DRL) is a promising technique for such a problem. Once trained, it can take near-optimal decisions almost instantaneously. Training DRL models for industrial-scale inventory systems with both dozens of locations and thousands of Stock Keeping Units (SKUs), however, has thus far remained an open challenge. We propose a novel approach to efficiently train DRL models for industrial-scale operational spare parts management problems, which combines global learning, action space decomposition and reward smoothing.

We apply our approach on the case of ASML's operational spare parts inventory network, which contains in the order of 10,000 SKUs and 60 locations. The results show that our DRL model achieves 3.8% to 5.1% cost savings compared to the currently-used heuristic and other benchmarks. By analyzing our model's proposed actions with a domain knowledge expert from ASML, we find that this strong performance can be attributed to the use of proactive lateral transshipments, sourcing from nearby locations and expediting shipments to at-risk locations. We conclude that our approach improves upon the state-of-the-art in operational spare parts management and extends the applicability of DRL to industrial-scale inventory systems.

These encouraging results can inform future work on operational spare parts management and deep reinforcement learning for inventory management at scale. They establish the effectiveness of DRL for operational spare parts management, and provide evidence for the value of distance-based proactive lateral transshipments and expediting. Moreover, they show that global learning can be extended to multi-location inventory problems by decomposing the (output) action space and careful (input) feature engineering. Decomposition of the action space can be taken much further than has thus far been the case

to train DRL models in a more efficient and scalable manner. Finally, our newly introduced reward smoothing technique can be used to more efficiently train DRL models for stochastic problems when good approximations for the reward function are available.

Our research thereby further underlines the potential of DRL for inventory management in general, and operational spare parts management in particular. Nonetheless, we see several opportunities for future work. Our approach currently considers SKUs as independent, even though literature on tactical spare parts management (e.g., Van Houtum and Kranenburg 2015) suggests that a system-approach may be more beneficial. Future work may use a Lagrangian relaxation-like approach to find system-wide near-optima, while maintaining the benefits of global learning. We also find that interpretability and explainability are a key concern in practice, and encourage future research to further investigate means to make DRL for inventory management more interpretable and explainable. Alternatively, it would be interesting to investigate if the observed use of distance-based proactive lateral transshipments and expediting can be used to create new, non-DRL, heuristics for operational spare parts management.

Acknowledgments

We are grateful to ASML for making this research possible. In particular, we would like to thank Tim Kragten, Tom Vlassak and Joan Stip. We would also like to thank Pratik Gajane and Tarkan Temizöz for their support during various phases of the research.

Appendix A: Derivation Fillrate under Poisson Demand

In this appendix, we show how to derive a location's fillrate under local demand, i.e., without accounting for incoming overflow demand from other locations. This fillrate is used in the proposed reward shaping method described in Section 4.3. Let $(i,j) : i \in I, j \in J$ denote an SKU-location pair, $I_{i,j}$ its current on-hand inventory, $D_{i,j} \sim \text{Poisson}(\lambda_{i,j})$ its single-period demand, and $\beta_{i,j}(I_{i,j})$ its fillrate under local demand. The fillrate $\beta_{i,j}(I_{i,j})$ can then be derived as follows:

$$\beta_{i,j}(I_{i,j}) = \frac{\mathbb{E}[\min\{D_{i,j}, I_{i,j}\}]}{\mathbb{E}[D_{i,j}]} \tag{5}$$

$$= \frac{\sum_{x=0}^{I_{i,j}-1} x\mathbb{P}\{D_{i,j}=x\} + \sum_{x=I_{i,j}}^{\infty} I_{i,j}\mathbb{P}\{D_{i,j}=x\}}{\mathbb{E}[D_{i,j}]} \tag{6}$$

$$= \frac{\sum_{x=0}^{I_{i,j}-1} \frac{x(\lambda_{i,j})^x e^{-\lambda_{i,j}}}{x!} + \sum_{x=I_{i,j}}^{\infty} \frac{I_{i,j}(\lambda_{i,j})^x e^{-\lambda_{i,j}}}{x!}}{\lambda_{i,j}} \tag{7}$$

$$= \frac{\sum_{x=0}^{I_{i,j}-1} \frac{x(\lambda_{i,j})^x e^{-\lambda_{i,j}}}{x!} + I_{i,j}\left(1 - \sum_{x=0}^{I_{i,j}-1} \frac{(\lambda_{i,j})^x e^{-\lambda_{i,j}}}{x!}\right)}{\lambda_{i,j}} \tag{8}$$

$$= \frac{I_{i,j}}{\lambda_{i,j}} - e^{-\lambda_{i,j}} \sum_{x=0}^{I_{i,j}-1} \frac{(I_{i,j}-x)\lambda_{i,j}^{x-1}}{x!} \tag{9}$$

Appendix B: Proof of Proposition 1

Proof of part (1): Let $C^{\pi}_{i,t}$ denote the costs for SKU i in period t under policy π, and let $\mathbb{E}[C^{\pi}_{i,t}]$ be its expectation with respect to the demand. We then have that:

$$\min_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^t \mathbb{E}[C^{\pi}_{i,t}]\right] \equiv \min_{\pi} \sum_{t=1}^{T} \gamma^t \mathbb{E}\left[\mathbb{E}[C^{\pi}_{i,t}]\right] \equiv \min_{\pi} \sum_{t=1}^{T} \gamma^t \mathbb{E}[C^{\pi}_{i,t}] \equiv \min_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} \gamma^t C^{\pi}_{i,t}\right] \tag{10}$$

whereby the equivalences follow from linearity of expectation, $\mathbb{E}[C^{\pi}_{i,t}]$ being a constant, and linearity of expectation, respectively.

Proof of part (2): For an inventory system with stochastic demand and a non-zero system-wide stockout rate, the non-zero system-wide stockout rate implies non-zero expected stock-out costs. Hence, $\mathbb{E}[C^{\pi}_{i,t}] > 0$ and sparsity in costs is thus eliminated.

Proof of part (3): As $\mathbb{E}[C^{\pi}_{i,t}]$ is a constant, we have that its variance $V[\mathbb{E}[C^{\pi}_{i,t}]] = 0$. Moreover, if an inventory system with stochastic demand has non-zero expected stock-out costs, then these stock-out costs must also be stochastic. Consequently, costs $C^{\pi}_{i,t}$ must also be stochastic and $V[C^{\pi}_{i,t}] > 0$. We thus have $V[\mathbb{E}[C^{\pi}_{i,t}]] < V[C^{\pi}_{i,t}]$.

Remark: Although these proofs are relatively straightforward, these three claims explain why reward smoothing can improve performance. More specifically, claim (1) establishes that reward smoothing does not affect convergence, claim (2) shows that sparsity in rewards is eliminated, and claim (3) confirms that variance in rewards is reduced. As sparsity and stochasticity in rewards are known to negatively impact (D)RL model performance (Sutton and Barto 2018), any method that reduces sparsity and stochasticity without affecting convergence can be expected to lead to stronger performance.

Appendix C: Hyperparameter Settings and Running Times

The hyperparameters for the DRL policies case study experiments (Section 5) are tuned manually. Their tuned values are provided in Table 4. For each setting, we use LeakyReLU as activation function, Adam (Kingma and Ba 2017) as optimizer, an early stopping patience of 15, a learning rate of 0.0001, a mini-batch size of 256, a horizon length of 256, a discount rate of 1.00 (no discounting), a single generation of models and a warm-up length during training of 1000. A comparison of the time to train the DRL policies under these hyperparameter settings is provided in Table 5. The time to evaluate was approximately the same for each policy, as the size of the neural network was the same for each DRL policy.

	Number of warehouses			
Hyperparameter	6	12	30	60
Hidden layers	256,128,64	384,192,96	512,256,128	1024,512,256
Number of rollouts	32	64	128	256
Sample size `DCL`, `DCL-RS`	50,000	100,000	100,000	-
Sample size `DCL-AS`, `DCL-AS-RS`	50,000	100,000	500,000	1,000,000

Table 4 Hyperparameter settings for the case study experiments

	Number of warehouses			
Policy	6	12	30	60
`DCL`	0hr.-01m.	0hr.-09m.	2hr.-41m.	-
`DCL-RS`	0hr.-01m.	0hr.-10m.	3hr.-14m.	-
`DCL-AS`	0hr.-01m.	0hr.-04m.	1hr.-25m.	15hr.-41m.
`DCL-AS-RS`	0hr.-01m.	0hr.-05m.	1hr.-28m.	22hr.-10m.

Table 5 Time to train for the DRL policies evaluated in the ablation study

References

ABB (2023) Value of reliability: ABB survey report 2023. industry's perspective on maintenance and reliability. URL `https://search.abb.com/library/Download.aspx?DocumentID=9AKK108468A6878`.

Akiba T, Sano S, Yanase T, Ohta T, Koyama M (2019) Optuna: A next-generation hyperparameter optimization framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–2631.

Akkerman FR, Begnardi L, Lo Bianco R, Temizöz T, Dijkman RM, Iacob M, Mes MR, Zhang Y, van Jaarsveld WL (2023) DynaPlex. Deep Reinforcement Learning Library.

Akkerman FR, Luy J, van Heeswijk WJA, Schiffer M (2024) Dynamic neighborhood construction for structured large discrete action spaces. *The 12th International Conference on Learning Representations*.

Akkermans H, Basten RJI, Zhu Q, van Wassenhove L (2024) Transition paths for condition-based maintenance-driven smart services. *Journal of Operations Management* 70(4):548–567.

Alfredsson P, Verrijdt J (1999) Modeling emergency supply flexibility in a two-echelon inventory system. *Management Science* 45(10):1416–1431.

ASML (2024) Customer support: Working round the clock to design customer-centric products and keep them running smoothly. URL `https://www.asml.com/en/products/customer-support`.

Axsäter S (1990) Modelling emergency lateral transshipments in inventory systems. *Management Science* 36(11):1329–1338.

Bakker REF (2016) Design of a new way of working to improve the planning of engineering change spare parts. EngD thesis, Eindhoven University of Technology.

Basten RJI, van Houtum GJ (2023) Spare parts inventory planning. Song JSJ, ed., *Research Handbook on Inventory Management*, chapter 19, 455–475 (Edward Elgar Publishing).

Boute RN, Gijsbrechts J, van Jaarsveld WL, Vanvuchelen N (2022) Deep reinforcement learning for inventory control: A roadmap. *European Journal of Operational Research* 298(2):401–412.

Büyükkurt MD, Parlar M (1993) A comparison of allocation policies in a two-echelon repairable-item inventory model. *International Journal of Production Economics* 29(3):291–302.

Caggiano KE, Muckstadt JA, Rappold JA (2006) Integrated real-time capacity and inventory allocation for reparable service parts in a two-echelon supply system. *Manufacturing & Service Operations Management* 8(3):292–319.

De Moor BJ, Gijsbrechts J, Boute RN (2022) Reward shaping to improve the performance of deep reinforcement learning in perishable inventory management. *European Journal of Operational Research* 301(2):535–545.

Federgruen A, Zipkin P (1984) Approximations of dynamic, multilocation production and inventory problems. *Management Science* 30(1):69–84.

Ge Q, van Jaarsveld W, Atan Z (2020) Optimal redesign decisions through failure rate estimates. *Naval Research Logistics* 67(4):254–271.

Gerrits B, Topan E, van der Heijden MC (2022) Operational planning in service control towers–heuristics and case study. *European Journal of Operational Research* 302(3):983–998.

Gijsbrechts J, Boute RN, Van Mieghem JA, Zhang DJ (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.

Glueck JJ, Koudal P, Vaessen W (2007) The service revolution in global manufacturing industries. *Deloitte Research* 1.

Howard C, Marklund J, Tan T, Reijnen I (2015) Inventory control in a spare parts distribution system with emergency stocks and pipeline information. *Manufacturing & Service Operations Management* 17(2):142–156.

Hu Q, Boylan JE, Chen H, Labib A (2018) OR in spare parts management: A review. *European Journal of Operational Research* 266(2):395–414.

IndustryWeek, Emerson (2017) How manufacturers achieve top quartile performance. URL `https://partners.wsj.com/emerson/unlocking-performance/how-manufacturers-can-achieve-top-quartile-performance/`.

Kaynov I, van Knippenberg M, Menkovski V, van Breemen A, van Jaarsveld WL (2024) Deep reinforcement learning for one-warehouse multi-retailer inventory management. *International Journal of Production Economics* 267:109088.

Kingma DP, Ba J (2017) Adam: A method for stochastic optimization. *The 3rd International Conference on Learning Representations*.

Kranenburg B (2006) *Spare parts inventory control under system availability constraints*. Ph.D. thesis, Eindhoven University of Technology.

Kranenburg B, van Houtum GJ (2009) A new partial pooling structure for spare parts networks. *European Journal of Operational Research* 199(3):908–921.

Kukreja A, Schmidt CP, Miller DM (2001) Stocking decisions for low-usage items in a multilocation inventory system. *Management Science* 47(10):1371–1383.

Lamghari-Idrissi DPT (2021) *A New After-Sales Service Measure for Stable Customer Operations*. Ph.D. thesis, Eindhoven University of Technology.

Lee YH, Jung JW, Jeon YS (2007) An effective lateral transshipment policy to improve service level in the supply chain. *International Journal of Production Economics* 106(1):115–126.

Madeka D, Torkkola K, Eisenach C, Foster D, Luo A (2022) Deep inventory management. ArXiv preprint arXiv:2210.03137.

Makridakis S, Spiliotis E, Assimakopoulos V (2022a) M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting* 38(4):1346–1364.

Makridakis S, Spiliotis E, Assimakopoulos V, Chen Z, Gaba A, Tsetlin I, Winkler RL (2022b) The M5 uncertainty competition: Results, findings and conclusions. *International Journal of Forecasting* 38(4):1365–1385.

Marklund J, Rosling K (2012) Lower bounds and heuristics for supply chain stock allocation. *Operations Research* 60(1):92–105.

Miller BL (1974) Dispatching from depot repair in a recoverable item inventory system: On the optimality of a heuristic rule. *Management Science* 21(3):316–325.

Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. *International Conference on Machine Learning*, 1928–1937.

Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M (2013) Playing Atari with deep reinforcement learning. ArXiv preprint arXiv:1312.5602.

Ng AY, Harada D, Russell S (1999) Policy invariance under reward transformations: Theory and application to reward shaping. *International Conference on Machine Learning*, volume 99, 278–287.

Oroojlooyjadid A, Nazari M, Snyder LV, Takáč M (2022) A deep q-network for the beer game: Deep reinforcement learning for inventory optimization. *Manufacturing & Service Operations Management* 24(1):285–304.

Paterson C, Teunter RH, Glazebrook K (2012) Enhanced lateral transshipments in a multi-location inventory system. *European Journal of Operational Research* 221(2):317–327.

Petropoulos F, Apiletti D, Assimakopoulos V, Babai MZ, Barrow DK, Ben Taieb S, Bergmeir C, Bessa RJ, Bijak J, Boylan JE, Browell J, Carnevale C, Castle JL, Cirillo P, Clements MP, Cordeiro C, Cyrino Oliveira FL, De Baets S, Dokumentov A, Ellison J, and others (2022) Forecasting: Theory and practice. *International Journal of Forecasting* 38(3):705–871.

Qi M, Shi Y, Qi Y, Ma C, Yuan R, Wu D, Shen ZJ (2023) A practical end-to-end inventory management model with deep learning. *Management Science* 69(2):759–773.

Rolf B, Jackson I, Müller M, Lang S, Reggelin T, Ivanov D (2023) A review on reinforcement learning algorithms and applications in supply chain management. *International Journal of Production Research* 61(20):7151–7179.

Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. ArXiv preprint arXiv:1707.06347.

Seidscher A, Minner S (2013) A semi-markov decision problem for proactive and reactive transshipments between multiple warehouses. *European Journal of Operational Research* 230(1):42–52.

Sherbrooke CC (1968) Metric: A multi-echelon technique for recoverable item control. *Operations Research* 16(1):122–141.

Sutton RS, Barto AG (2018) *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning series (MIT Press), second edition, ISBN 9780262039246.

Temizöz T, Imdahl C, Dijkman RM, Lamghari-Idrissi DPT, van Jaarsveld WL (2023) Deep controlled learning for inventory control. ArXiv preprint arXiv:2011.15122.

Topan E, Eruguz AS, Ma W, van der Heijden MC, Dekker R (2020) A review of operational spare parts service logistics in service control towers. *European Journal of Operational Research* 282(2):401–414.

Topan E, van der Heijden MC (2020) Operational level planning of a multi-item two-echelon spare parts inventory system with reactive and proactive interventions. *European Journal of Operational Research* 284(1):164–175.

Trimponias G, Dietterich TG (2023) Reinforcement learning with exogenous states and rewards. ArXiv preprint arXiv:2303.12957.

US Census Bureau (2020) Industry revenue of "repair and maintenance" in the U.S. from 2012 to 2024. `https://www.statista.com/forecasts/884972/repair-and-maintenance-revenue-in-the-us`.

Van Aspert M (2015) Design of an integrated global warehouse and field stock planning concept for spare parts. EngD thesis, Eindhoven University of Technology.

Van der Auweraer S, Boute RN (2019) Forecasting spare part demand using service maintenance information. *International Journal of Production Economics* 213:138–149.

Van Dijck TL, Fleuren TWA, Temizöz T, Merzifonluoglu Y, Hendriks M, van Jaarsveld WL (2024) Inventory planning in capacitated high-tech assembly systems under non-stationary demand. Available at SSRN 4843271.

Van Hezewijk L, Dellaert NP, van Jaarsveld WL (2024) Scalable deep reinforcement learning in the non-stationary capacitated lot sizing problem. Available at SSRN 4846298.

Van Hezewijk L, Dellaert NP, van Woensel T, Gademann N (2022) Using the proximal policy optimisation algorithm for solving the stochastic capacitated lot sizing problem. *International Journal of Production Research* 1–24.

Van Houtum GJ, Kranenburg B (2015) *Spare parts inventory control under system availability constraints*, volume 227 (Springer).

Vanvuchelen N, De Boeck K, Boute RN (2024) Cluster-based lateral transshipments for the Zambian health supply chain. *European Journal of Operational Research* 313(1):373–386.

Vanvuchelen N, De Moor BJ, Boute RN (2023) The use of continuous action representations to scale deep reinforcement learning for inventory control. Available at SSRN 4253600.

Verleijsdonk P, van Jaarsveld WL, Kapodistria S (2024) Scalable policies for the dynamic traveling multi-maintainer problem with alerts. *European Journal of Operational Research* 319(1):121–134.

Vliegen IMH (2009) *Integrated planning for service tools and spare parts for capital goods*. Ph.D. thesis, Eindhoven University of Technology.

Wellens AP, Kourentzes N, Udenio M (2023) When and how to use global forecasting methods on heterogeneous datasets. Available at SSRN 4629272.

Zheng L, Terpenny J, Sandborn P (2015) Design refresh planning models for managing obsolescence. *IISE Transactions* 47(12):1407–1423.