MORSE: An Adaptive Decision-Making Framework Combining Reinforcement Learning and Multi-Objective Evolutionary Algorithms for Dynamic Inventory Control

Niki Kotecha * **Antonio del Rio Chanona** *

* *Sargent Centre for Process Systems Engineering, Imperial College London, SW7 2AZ, United Kingdom*

Abstract: In supply chain management, decision-making often involves balancing multiple conflicting objectives, such as cost reduction, service level improvement, and environmental sustainability. Traditional methods for multi-objective optimization, such as linear programming and evolutionary algorithms, have proven useful but struggle to adapt in real-time or handle the dynamic nature of supply chains. In this paper, we propose a novel approach that combines Reinforcement Learning (RL) and Multi-Objective Evolutionary Algorithms (MOEAs) to address these challenges. Our method leverages MOEAs to search the parameter space of policy neural networks, resulting in a Pareto front of policies. This equips the decision-maker with a swarm of policies that can be dynamically switched based on the current system conditions and objectives, ensuring flexibility and adaptability in real-time decision-making. We demonstrate the effectiveness of this hybrid approach through a series of case studies that showcase its ability to respond to the changing dynamics of supply chain environments. We also outperform state-of-the-art methods when benchmarking against our inventory management case study. The proposed strategy not only improves decision-making efficiency but also provides a more resilient framework for fast decision-making and handling uncertainty in supply chains.

1. INTRODUCTION

The development of sustainable supply chains has emerged as a key strategic focus for many organisations, driven by the increase in regulatory requirements and pressure from consumers to adopt environmentally friendly practices. As companies strive towards making their value chains more sustainable, the integration of sustainability principles into every aspect of their operations, particularly in decision-making, becomes essential.

Supply chain problems are highly interconnected, complex and operate under uncertain conditions. In the field of inventory management, maintaining optimal control over stock levels and supply chain dynamics is essential for ensuring efficient operations, cost-effectiveness and high service-quality. Traditionally, inventory management has relied on classical techniques such as Economic Order Quantity (EOQ) and Just-In-Time (JIT) approaches due to their simplicity and ease of implementation (13). These methods, while effective in stable environments, often assume predicable demand and supply conditions, which can be unrealistic in today's volatile market characterized by fluctuations in consumer preferences and external disruptions. Disruptions like the COVID-19 pandemic highlighted the vulnerabilities in supply chains, emphasizing the need for fast decision-making and adaptive strategies, further stressing the need to leverage data-driven tools to mitigate the effects of disruptions and optimize our systems (12).

Over the years, the complexity of modern systems has driven the development of data-driven tools from dynamic programming to stochastic and distributionally-robust optimization approaches. While these methods show promise, they often struggle with scalability and require a priori knowledge of the system's underlying distributions, which can limit their practical applicability in real-time decision-making.

Reinforcement learning (RL) has emerged as a promising data-driven decision-making framework due to its ability to handle complex, dynamic and uncertain environments. RL learns optimal policies by interacting with the environment and learning through trial-and-error.

With the growing strategic sustainability focus, organizations are challenged with building behavioral policies that trade-off conflicting objectives while still aligning with overarching business objectives. This highlights the need to develop decision-making frameworks that can handle multiple objectives in dynamic environments.

1.1 Related Work & Motivation

Multi-objective optimization (MOO) techniques are essential for addressing complex decision-making problems with conflicting objectives. These methods aim to find a set of optimal solutions, known as the Pareto optimal, recognizing that a single optimal solution is often not enough due to inherent trade-offs between different objectives. The collection of non-dominated solutions forms the Pareto front, and decision-makers can then choose the most appropriate solution from the Pareto front based on their preference. A key challenge in MOO is balancing

proximity (how close solutions are to the true Pareto front) with diversity (how well the solutions span the objective space), particularly in problems with many objectives or large search spaces. While linear programming (LP), evolutionary algorithms (EAs), and decomposition-based approaches have advanced MOO, they often struggle to handle dynamic, real-time changes in objectives.

In supply chain management, MOO has been well studied with mixed-integer linear programming (MILP) models being widely used (1; 2; 3). These models are often extended to Fuzzy MILP to take into account inherent uncertainty present in supply chains (4). Fundamental MOO methods such as weighted sum $\epsilon-$constraint methods are either directly employed or incorporated as components of hybrid methods (3). In recent years, there has been a growing focus on evolutionary algorithms, with NSGA-II, a multi-objective evolutionary algorithm (MOEA), being the most commonly used. Evolutionary algorithms are efficient, practical, and particularly effective in solving non-convex optimization problems, which can be challenging for traditional methods(16).

However, in supply chain management, objectives often evolve over time due to changing conditions such as fluctuating demand, supply chain disruptions, and market dynamics. This requires decision-making frameworks capable of adapting to shifting objectives, making Dynamic Multi-Objective Optimization (DMOO) essential (13). Unlike traditional Multi-Objective Optimization (MOO), which assumes fixed objectives, DMOO incorporates real-time adjustments to the optimization process, ensuring that solutions remain effective as the system dynamics evolve. Therefore, extending MOO to DMOO allows for more responsive and flexible decision-making in supply chains, where objectives need to be balanced under uncertainty.

To address these dynamic challenges, Reinforcement Learning (RL) presents a promising alternative. By allowing agents to learn through trial-and-error in uncertain environments, RL can dynamically adjust to real-time changes. Literature shows that gradient-based RL methods, such as A2C, A3C, PPO are effective and outperform traditional methods due to their ability to perform in uncertain environments (7; 8; 9). However, most RL studies focus on single objectives, predominantly financial ones, and largely ignore other important objectives. This narrow focus limits the real-world applicability of these studies, as supply chain management often involves balancing multiple conflicting objectives such as cost, service level, and environmental impact. Traditional RL methods are not designed to handle these complexities effectively and overlook the potential benefits of alternative, data-driven approaches that are better suited to complex, multi-objective problems.

Multi-Objective Evolutionary Algorithms (MOEAs), for instance, offer a robust way to search across a broad solution space without requiring differentiable objective functions. Their ability to maintain a population of solutions, each addressing different trade-offs between objectives, makes them a natural fit for tackling multi-objective problems in dynamic environments like supply chain management. In this paper, we combine the adaptive decision-making capabilities of RL with MOEAs' ability to balance diversity and proximity to the Pareto front. Specifically, we use MOEAs to search the parameter space of neural network policies, resulting in a Pareto front of policies. This equips the decision-maker with a swarm of policies, allowing for dynamic switching between policies based on the current system objectives and dynamics. This strategy ensures flexibility in real-time decision-making under uncertainty, allowing the system to adapt quickly to changing conditions without the need to evolve policies continuously.

The rest of the paper is organized as follows: Section 2 provides the background on reinforcement learning and evolutionary strategies, Section 3 describes the proposed multi-objective evolutionary algorithm based reinforcement learning framework in more detail. Section 4 discusses the simulation and experimental results and Section 5 presents some benchmarking results. Finally, Section 6 summarizes the paper and provides an outlook for future work.

2. PRELIMINARIES

2.1 Introduction to Reinforcement Learning

In single agent reinforcement learning, the agent aims to learn an optimal policy by interacting with the environment and learning through trial-and-error. In RL, the agent observes the current state $s_t \in S \subseteq \mathbb{R}^{n_s}$, chooses an action $a \in A \subseteq \mathbb{R}^{n_a}$ with probability given by the policy $\pi(a|s)$ and transitions into the next state $s_{t+1} \in S \subseteq \mathbb{R}^{n_s}$ with probability given by the state transition probability function $\mathcal{T}(s_t, a_t, s_{t+1})$ and receives a reward $r_{t+1} = \mathcal{R}(s_t, a_t, s_{t+1}) \in \mathbb{R}$. The agent finds an optimal policy π^* by maximizing the expected sum of rewards over a time horizon defined as:

$$J(\pi) = \mathbb{E}_{(s_t,a_t)\sim\pi}\left[\sum_{t=0}^{T} \gamma^t r_t(s_t, a_t)\right] \tag{1}$$

$$\pi^* \in \arg\max_{\pi} J(\pi) \tag{2}$$

In practice, the policy is parameterized by a policy function such that $\pi^* \approx \pi^*(a|s;\theta)$, where $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ represents the weights of the neural network and Θ is the parameter space. The aim is to optimize parameters θ to maximize the cumulative rewards, learning an effective mapping from states s to actions a.

To achieve this, policy-gradient methods optimize parameters θ by calculating gradients of the expected reward with respect to parameters θ. This approach contrasts to value-based methods, as they directly update the policy, making it suitable for continuous action spaces and high dimensions. Algorithms like REINFORCE, Proximal Policy Optimization (PPO), and Trust Region Policy Optimization (TRPO) are popular choices here, as they leverage policy gradients and surrogate objectives to improve stability and performance (8). However, policy gradient methods rely on sampling rollouts to estimate gradients. These samples can vary greatly due to the stochastic nature of the environment and policy, leading to high variance in gradient estimates. They also tend to be sensitive to hyperparameters.

An alternative approach is using Evolutionary Strategies (ES). ES methods optimize parameters θ by exploring the parameter space Θ through simulating an evolutionary process. Rather than relying on gradient-based updates, ES methods employ population-based search techniques to evolve policies. In each iteration, a population of candidate policies is evaluated based on a reward metric, with the most successful candidates selected to "reproduce" via mutation and recombination, guiding the search toward high-performing areas of the parameter space.

Evolutionary strategies for reinforcement learning (ES-RL) have shown promising results in optimizing policy parameters due to their lack of need for backpropagation (14). Instead of relying on gradient information, ES-RL evaluate policy performance by sampling multiple trajectories and directly update the parameters towards those that lead to higher performance. This method enhances scalability in distributed settings as its easier to parallelize. There is also fewer hyperparameters to tune compared to gradient-based methods and they are less likely to get stuck in local optima as they are population-based methods so search "globally" rather than relying on stochastic estimates of gradients to optimize the parameters. Although ES-RL has demonstrated competitive performance, as highlighted in OpenAI's work (14), where it showed comparable results to other policy gradient methods on environments like MuJoCo and Atari, it is important to note that ES-RL does not universally outperform policy gradient methods. Rather, ES-RL is particularly advantageous in specific contexts, especially when extended to multi-objective evolutionary reinforcement learning (MOEA-RL) (17; 18). The population-based search techniques of ES-RL can be leveraged to handle multiple, conflicting objectives in complex, dynamic environments providing a framework for optimization problems where traditional RL methods may face limitations.

3. METHODOLOGY

This section presents the proposed multi-objective reinforcement learning framework which integrates traditional reinforcement learning with multi-objective evolutionary strategies.

*3.1 **MORSE** - **M**ulti-**O**bjective **R**einforcement learning via **S**trategy **E**volution*

Due to the aforementioned benefits of evolutionary strategies, the proposed methodology leverages multi-objective evolutionary strategies (MOEA) for multi-objective reinforcement learning (MORL). In this work, we leverage MOEA to directly optimize the parameter space of the policies, building a Pareto set of policies rather than a Pareto set of solutions, as is common in traditional multi-objective optimization methods. This approach provides the decision maker with a diverse set of adaptable and dynamic policies, facilitating rapid decision-making in complex environments. The proposed methodology is shown in Algorithm 1 and Figure 1.

This results in a Pareto front of policies rather than a single policy. This allows decision-makers to have the flexibility to switch between policies and choose one that gives the best trade-off according to the needs in real-time. This is valuable in dynamic environments where priorities may shift due to external disruptions to the system (6; 15).

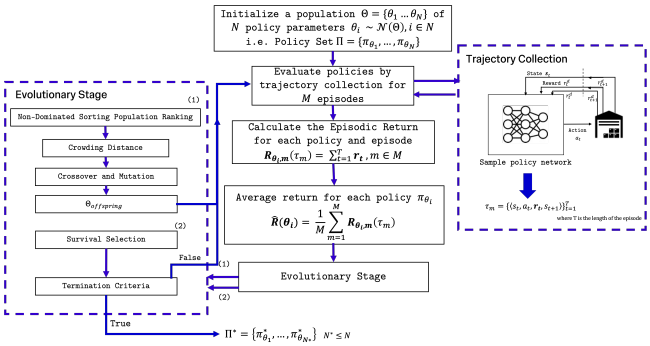

Fig. 1. Schematic overview of our MORSE framework.

Algorithm 1 MORSE

Input: Number of policies N, Maximum generations G, Evaluation episodes E
Output: Pareto front set of policies $\mathcal{F}_{\text{Pareto}}$

Step 1: Initialization
Generate a population of policies $\Pi = \{\pi_1, \pi_2, \dots, \pi_N\}$, where each policy π_i is parameterized by θ_i
Step 2: Policy Evaluation
for each policy $\pi_i \in \Pi$ **do**
  for each episode $e = 1, 2, \dots, E$ **do**
    Initialize state s_0 from the environment
    for each time step $t = 0, 1, \dots, T$ **do**
      Select action $a_t \sim \pi_{\theta_i}(s_t)$ based on the policy
      Execute action a_t, observe reward r_t, next state s_{t+1}
      Accumulate discounted reward for each objective
    end for
  end for
  Compute average objective values over E episodes
end for
Step 3: Non-dominated Sorting
Sort policies into fronts based on dominance relationships: $\mathcal{F}_1, \mathcal{F}_2, \dots$
Step 4: Crowding Distance Calculation
for each front $\mathcal{F}_j$ **do**
  Compute crowding distance $d(\pi_i)$ for each policy π_i
end for
Step 5: Selection and Reproduction
Binary tournament selection, then crossover & mutation:
$\theta_{\text{offspring}} = \text{Crossover}(\theta_i, \theta_j) + \text{Mutation}(\theta_k)$
Step 6: Survival Selection
Combine parent population $\mathcal{P}$ and offspring $\mathcal{P}_{\text{offspring}}$. Select top N policies based on non-domination rank and crowding distance:
$\mathcal{P}' = \text{Top-}N(\mathcal{P} \cup \mathcal{P}_{\text{offspring}})$
Step 7: Termination Criteria
if Termination criteria met **then**
  break
end if
Step 8: Pareto Front Identification
Identify non-dominated solutions in the final population:
return Pareto front set $\mathcal{F}_{\text{Pareto}}$; $\Pi = \{\pi_1^*, \pi_2^*, \dots, \pi_{n*}^*\}$

3.2 Multi-Objective Markov Decision Process

The sequential decision-making problem is formulated as a Multi-Objective Markov Decision Process, which can be defined as a tuple $\langle S, A, \mathcal{T}, \gamma, \mu, \mathbf{R} \rangle$ where S is the state space, A is the action space, $\mathcal{T}$ is the probability transition function where $\mathcal{T} : S \times A \times S \to [0, 1]$, $\gamma \in [0, 1]$ is the

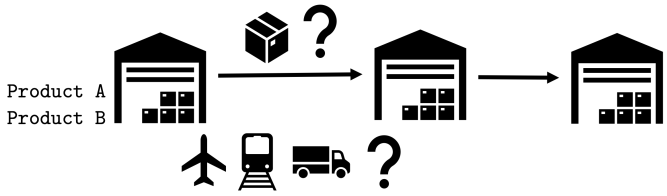

Fig. 2. Schematic representation of the inventory management system setup, illustrating key components and their interactions.

discount factor, $\mu : S \to [0, 1]$ is a probability distribution over initial states and $\mathbf{R} : S \times A \times S \to \mathbb{R}^d$ is a vector valued reward function where $d \geq 2$ is the number of objectives. The vector-valued reward function $\mathbf{R}$ is one of the differences between single-objective RL and multi-objective RL. Finally, a policy $\pi : S \to A \in \Pi$, maps states to actions where Π is a set of all the possible policies. Another notable difference between multi-objective and single-objective MDPs is the vector valued value function, $\mathbf{V}^\pi \in \mathbb{R}^d$ which is conditioned on the number of objectives and is defined as $\mathbf{V}^\pi(s) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_{t+1}\right]$ where $\mathbf{r}_{t+1} = \mathbf{R}(s_t, a_t, s_{t+1})$. A note that due to the multi-objective nature of the definition, it is possible to encounter a situation where for objectives j and k where $j, k \in \{1, 2, \ldots, d\}$ and policies π and π' where $\pi, \pi' \in \Pi$, both of the following inequalities can hold true:

$$V_j^\pi(s) > V_j^{\pi'}(s) \quad \text{(Policy } \pi \text{ is better for objective } j\text{)}$$
$$V_k^\pi(s) < V_k^{\pi'}(s) \quad \text{(Policy } \pi' \text{ is better for objective } k\text{)}$$

This indicates that while policy π outperforms policy π' in objective j, it underperforms in terms of objective k. This illustrates the concept of trade-offs in multi-objective reinforcement learning, where optimizing for one objective can lead to suboptimal performance in another. Therefore, the agent must make decisions based on the relative importance of each objective.

3.3 Inventory Management

The modeled system is a multi-echelon, multi-product inventory management network, with nodes connected by predefined distances as seen in Figure 2. For each node and each product at each time step in the simulation, the policy outputs two key actions:

- **Order Replenishment**: A continuous action within the range $[-1, 1]$ represents the replenishment amount for each product at each node. The policy, parametrized by a neural network, outputs a mean and standard deviation of a Gaussian distribution, from which we sample the continuous action. After sampling, a min-max scaling step is used within the environment to map this continuous value to a feasible quantity. This approach enhances scalability and helps avoid the combinatorial explosion problem often seen in discrete and mixed-integer optimization problems.
- **Transportation Mode**: A discrete action that represents the mode of transportation selected for product movement, which can include air, rail, or truck.

Our inventory management model integrates three cumulative objective functions throughout the time horizon: Maximize the profit across all nodes, minimize the transportation emission across all nodes, minimize the lead time across all nodes. The resulting multi-objective optimization problem can be formulated as follows:

$$\max \sum_{m=1}^{M} \sum_{p=1}^{P} \sum_{t=1}^{T} P_s^{m,p} s_r^{m,p}[t] - C^{m,p} o_r^{m,p}[t] - T^{m,p} L^{m_u} o_r^{m,p}[t] - I^{m,p} i^{m,p}[t] - B^{m,p} b^{m,p}[t] \tag{3}$$

$$\min \sum_{m=1}^{M} \sum_{p=1}^{P} \sum_{t=1}^{T} E^m L^{m_u} o_r^{m,p}[t] \tag{4}$$

$$\min \sum_{m=1}^{M} \sum_{p=1}^{P} \sum_{t=1}^{T} \tau_r^{m,p}[t] \tag{5}$$

$$\begin{aligned}
&i^{m,p}[t] = i_0^{m,p}[t] - s_r^{m,p}[t] + a_r^{m,p}[t], \quad \forall m, \forall p, \forall t, \\
&b^{m_d,p}[t] = b_0^{m_d,p}[t] - s_r^{m_d,p}[t] + d_r^{m_d,p}[t], \quad \forall m, \forall p, \forall d \in D_m, \\
&s_r^{m_d,p}[t] \leq b_0^{m_d,p}[t] + d_r^{m_d,p}[t], \quad \forall m, \forall p, \forall t, \forall d \in D_m, \\
&s_r^{m,p}[t] \leq i_0^{m,p}[t] + a_r^{m,p}[t], \quad \forall m, \forall p, \forall t, \\
&a_r^{m,p}[t] = s_r^{m_u,p}[t - \tau_r^m], \quad \forall m \neq 1, \forall p, \forall t \geq \tau_r^m, \\
&a_r^{1,p}[t] = s_r^{1,p}[t - \tau_r^1], \quad \forall p, \forall t \geq \tau_r^1, \\
&d_r^{m_d,p}[t] = o_r^{d,p}, \quad \forall m, \forall p, \forall d \in D_m, \\
&d_r^{m,p}[t] = c^{m,p}[t], \quad \forall m \in C, \forall p, \forall t, \\
&o_r^{m,p}[t] \leq O_{r_{\max}}^m, \quad I^m[t] \leq I_{\max}^m, \quad \forall m, \forall p, \forall t.
\end{aligned}$$

The goal is to ascertain the optimal action for each node m and each product p during each time period t spanning over a total of T time periods within a discrete-time setup.

s_r is the amount of goods shipped to a downstream node (or customers); o_r is the re-order quantity; d_r is the demand from downstream node(s); a_r is the acquisition at the current time step; c corresponds to customer demand; i and b are the on-hand inventory level and backlog at the end of a time period; I_0 and b_0 denote the initial on-hand inventory level and backlog; τ is lead time; L^{m_u} represents the distance from node m to its upstream supplier.

P_s, C, T, I, B, are cost coefficients - selling price, cost of re-order, transportation, stock, backlog, respectively; E is unit transportation emission; $O_{r_{\max}}$ and $I_{\max}$ represent the maximal re-order amount and node storage capacity, respectively. The subscript u refers to the upstream node, d denotes the downstream node. Moreover, customer demand is modeled as a non-stationary Poisson distribution to reflect its variability over time, while lead time is modeled as a Poisson distribution to capture the inherent unpredictability in transportation durations.

4. CASE STUDIES

In this section, we examine the adaptability of our methodology through several case studies. Using our approach, we derive a Pareto set of policies that optimally balance competing objectives. When a disruption affects the system, this Pareto set allows for swift policy switching, enabling us to select a policy that best meets the real-time needs and constraints. By dynamically adjusting to changing conditions, we demonstrate how our method offers resilience and flexibility in complex environments.

4.1 Case study 1 - Emission Penalties

As environmental concerns continue to grow, governments are increasingly implementing regulations to mitigate the

impacts of climate change. These regulations, such as emission taxes, not only directly affect profitability but also require firms to adapt their operational strategies in order to meet environmental standards.

In this case study, we simulate the impact of an emission tax, which penalizes firms when emissions exceed a predefined threshold within a given time period.

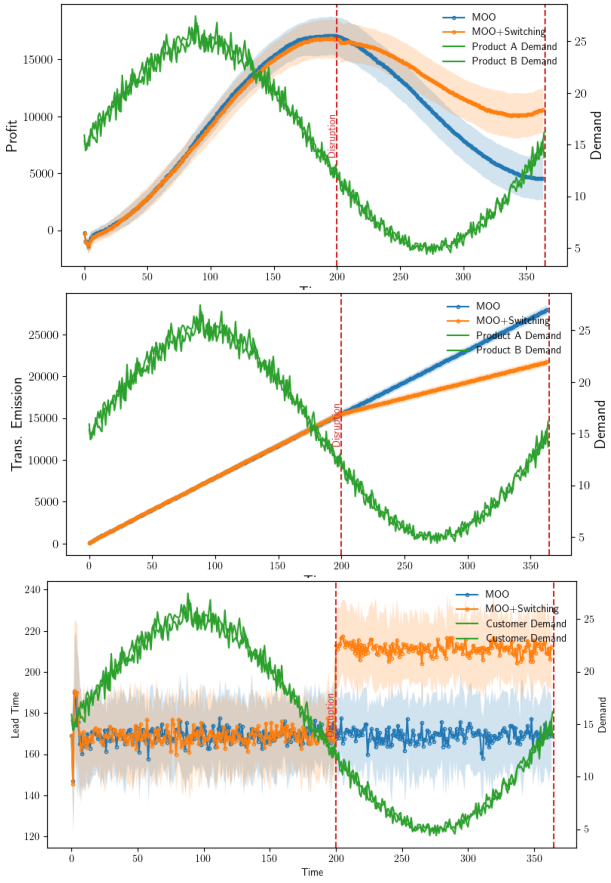

Fig. 3. Dynamics of **cumulative profit**, **cumulative transportation emission**, and **non-cumulative lead time** under emission tax introduction scenario.

As seen in Figure 3, the emission tax leads to a decline in cumulative profits due to penalties for exceeding the emission threshold. Our adaptive strategy protects profits and minimizes emissions to comply with environmental regulations. By adjusting operational strategies in real time, the system can balance the trade-off between profitability and sustainability, at the expense of higher lead times. Therefore, by using our multi-objective approach, which results in a Pareto set of policies, the system is capable of making real-time adjustments and decisions based on the current state of the environment. This flexibility allows the system to effectively navigate trade-offs between competing objectives, ensuring that operational goals are met while adhering to system disruptions and constraints. As a result, the system exhibits enhanced resilience and adaptability, enabling it to respond dynamically to evolving conditions.

4.2 Case study 2 - Geopolitical Tensions

In light of increasing geopolitical tensions, businesses are facing heightened risks and uncertainties that can lead to a rise in operational costs. Geopolitical disruptions, such as trade restrictions, sanctions, or supply chain instability, often require rapid strategic adjustments to maintain resilience and profitability. These conditions can lead to increased costs in procurement and transportation, directly impacting a firm's financial performance. In this scenario, we simulate the impact of geopolitical tensions by increasing the costs by 10% over a certain duration.

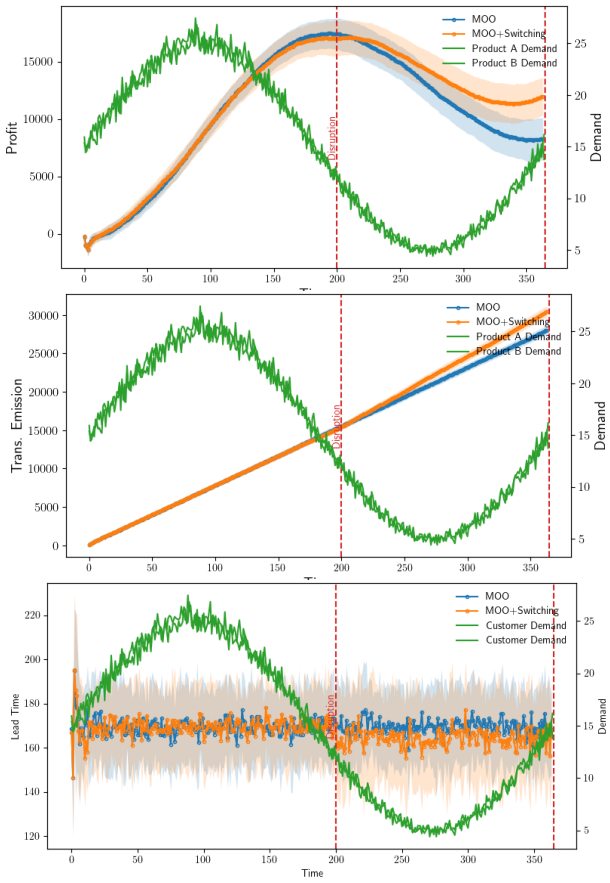

Fig. 4. Dynamics of **cumulative profit**, **cumulative transportation emission**, and **non-cumulative lead time** under geopolitical tension scenario.

As seen in Figure 4, the increase in costs due to geopolitical tensions leads to a decrease in cumulative profits. In response to these rising costs, our adaptive strategy aims to shield profits by dynamically adjusting operations. This ensures that the system can mitigate the impact of ongoing disruptions, optimizing performance despite the external challenges. By incorporating flexibility and real-time decision-making, our strategy allows for effective navigation through uncertain geopolitical environments, safeguarding long-term profitability.

5. BENCHMARKING

In this section, we benchmark our methodology against two state-of-the-art MORL approaches: Concave Augmented Pareto Q-Learning (CAPQL) (19) and Multi-Objective Natural Evolution Strategy (MONES) (20).

As shown in Figure 5 our methodology outperforms both CAPQL and MONES, achieving superior performance across objectives in the inventory management case study presented in Section 3.3. This analysis highlights the advantages of our approach, particularly in optimizing complex, multi-objective systems.

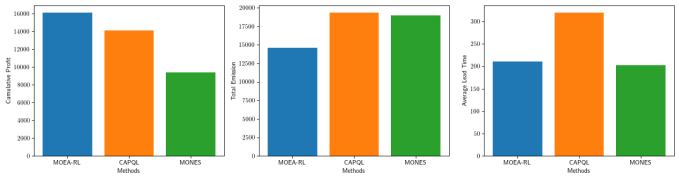

Fig. 5. Performance comparison of our methodology against other MORL methods: CAPQL and MONES.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a strategy that integrates RL with MOEAs to address the complex, dynamic nature of supply chain management. We demonstrate that using MOEA to search the policy neural network parameter space results in a Pareto front of policies. This approach equips the decision-maker with a swarm of policies that can be dynamically switched based on the current system objectives. This strategy enhances fast decision-making, resilience, and flexibility in uncertain and changing environments. We demonstrate the effectiveness of our method through a series of case studies, which showcase its adaptability and ability to balance multiple conflicting objectives in real-time. Our approach proves beneficial for optimizing complex, multi-objective problems typically encountered in supply chain management.

For future work, we plan to enhance the proposed approach by integrating human expertise to improve the search efficiency of the evolutionary algorithms. Additionally, we aim to extend our framework to handle partially observable environments, which would be crucial for decision-making in scenarios with incomplete information, while also expanding our method to multi-agent settings to enable collaborative decision-making in supply chains.

REFERENCES

[1] G. Chen, F. kaveh, and A. Peivandizadeh, "Resilient supply chain planning for the perishable products under different uncertainty," Mathematical Problems in Engineering, vol. 2022, p. 1–12, Aug. 2022.

[2] D. Mogale, A. De, A. Ghadge, and E. Aktas, "Multi-objective modelling of sustainable closed-loop supply chain network with price-sensitive demand and consumer's incentives," Computers Industrial Engineering, vol. 168, p. 108105, June 2022.

[3] V. Cantú, C. Azzaro-Pantel, and A. Ponsich. "A novel matheuristic based on bi-level optimization for the multi-objective design of hydrogen supply chains," Computers and Chemical Engineering, 2021.

[4] M. H. Alavidoost, A. Jafarnejad, and H. Babazadeh, "A novel fuzzy mathematical model for an integrated supply chain planning using multi-objective evolutionary algorithm," Soft Computing, vol. 25, pp. 1777–1801, Aug. 2020.

[5] F. Delfani, H. Samanipour, H. Beiki, A. V. Yumashev, and E. M. Akhmetshin, "A robust fuzzy optimisation for a multi-objective pharmaceutical supply chain network design problem considering reliability and delivery time," International Journal of Systems Science: Operations & Logistics, vol.9, pp.155,Dec.2020.

[6] Hayes, C.F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., Verstraeten, T., Zintgraf, L.M., Dazeley, R., Heintz, F. and Howley, E. "A practical guide to multi-objective reinforcement learning and planning." Autonomous Agents and Multi-Agent Systems, vol. 36, no. 1, pp. 26. 2022.

[7] M. Shakya, B.-S. Lee, and H. Y. Ng, "A deep reinforcement learning approach for inventory control under stochastic lead time and demand," in 2022 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, Dec. 2022

[8] F. Stranieri and F. Stella, "A deep reinforcement learning approach to supply chain inventory management," 2022.

[9] T. Demizu, Y. Fukazawa, and H. Morita, "Inventory management of new products in retailers using model-based deep reinforcement learning," Expert Systems with Applications, vol. 229, p. 120256, Nov. 2023.

[10] A. Panichella, "An adaptive evolutionary algorithm based on non-euclidean geometry for many-objective optimization," in Proceedings of the Genetic and Evolutionary Computation Conference, July 2019.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, pp. 182–197, Apr. 2002.

[12] Kotecha, Niki, and Antonio del Rio Chanona. "Leveraging Graph Neural Networks and Multi-Agent Reinforcement Learning for Inventory Control in Supply Chains." arXiv preprint arXiv:2410.18631. 2024.

[13] Qiu, Y., Kotecha, N. and del Rio Chanona, A., Leveraging reinforcement learning and evolutionary strategies for dynamic multi objective decision making in supply chain management. IFAC-PapersOnLine, 58(14), pp.598-603. 2024.

[14] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," 2017.

[15] Van Moffaert, Kristof, and Ann Nowé. "Multi-objective reinforcement learning using sets of pareto dominating policies." The Journal of Machine Learning Research 15, no. 1: 3483-3512. 2014.

[16] Xue, Ke, et al. "Evolutionary Gradient Descent for Non-convex Optimization." IJCAI. 2021.

[17] Li, Kaiwen, Tao Zhang, and Rui Wang. "Deep reinforcement learning for multiobjective optimization." IEEE transactions on cybernetics. 2020

[18] Zou, Fei, et al. "A reinforcement learning approach for dynamic multi-objective optimization." Information Sciences 546: 815-834. 2021.

[19] Lu, Haoye, Daniel Herman, and Yaoliang Yu. "Multi-objective reinforcement learning: Convexity, stationarity and pareto optimality." The Eleventh International Conference on Learning Representations. 2023.

[20] C. F. Hayes, R. Radaulescu, and Bargiacchi. A practical guide to multi-objective reinforcement learning and planning. AAMAS, 2022.