12th IFAC International Symposium on
Advanced Control of Chemical Processes
July 14-17, 2024. Toronto, Canada

Leveraging reinforcement learning and evolutionary strategies for dynamic multi objective decision making in supply chain management

Yue Qiu * **Niki Kotecha** * **Antonio del Rio Chanona** *

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

Abstract: Reinforcement learning (RL) has been widely applied in supply chain management due to its performance in dynamic, uncertain environments. However, most RL studies focus on a single objective, differentiable reward functions, and lack the ability to handle multiple conflicting non-differentiable objectives which is the case in many real-world problems such as in inventory control. The proposed multi-objective algorithm deploys a derivative-free approach to effectively optimize non-differentiable objective functions. The framework leverages the advantages of both reinforcement learning (RL) methods and multi-objective evolutionary algorithms (MOEAs) to obtain a Pareto set of policies. The effectiveness of our method is demonstrated through two case studies, each illustrating the adaptability of the policy of choice in varying scenarios. Our methodology finds a diverse set of policies, which allows decision-makers to better handle and mitigate the consequences of disruptions.

1. INTRODUCTION

Supply chains are vital to ensure the smooth transition of goods and services, serving as the backbone of the modern economy to meet our needs in the 21st century. The significance of supply chains becomes evident when considering the chemical and pharmaceutical sector which boasts a market value of over $6 trillion worldwide. Current supply chains are highly interconnected, complex and operate under uncertain environments. The complexity causes risks in disruptions and sub-optimal performance due to operational failure or lack of coordination between different entities. Two important disruption types within a supply chain are the bullwhip effect (amplification of demand variability) and the ripple effect (disruption propagation). These effects can cause issues such as excess or lack of inventory and sub-optimal performance. In the face of disruption, fast decision-making becomes paramount.

In light of the global movement towards net-zero, supply chains should move towards an era where they encompass the triple bottom line sustainability principle - balancing profitability, social responsibility, and environmental sustainability. This involves considering factors such as profitability, fair labour conditions, ethical practices, reducing emissions, and conserving natural resources and biodiversity.

1.1 Related Work & Motivation

Traditional approaches to optimization and decision-making have paved the way for solving complex multi-objective problems. For example, a powerful methodology to address the need for balancing conflicting objectives in supply chain management and other domains is the use of multi-objective optimization (MOO). These methods aim to find a set of optimal solutions, known as the Pareto optimal, recognizing that a single optimal solution is often not meaningful due to inherent trade-offs between different objectives. The collection of all Pareto optimal solutions forms the Pareto front and decision-makers can then choose the most appropriate solution from the Pareto front based on their preference. To guide effective decision-making, solutions should not only approximate the true Pareto front closely (proximity) but also span a broad section of the objective space (diversity), providing a diverse array of choices. Attaining this balance of proximity and diversity is challenging, especially in problems with many objectives or vast search spaces. Techniques such as Linear Programming (LP), Evolutionary Algorithms (EAs), and decomposition-based algorithms have been developed to find a balanced representation of the Pareto front efficiently. MOO techniques can be extended to dynamic multi-objective optimization (DMOO) methods where the objective can change over time. DMOO requires algorithms that can adapt to changes in the problem definition, which is more complex than in static MOO.

MOO in supply chain management has been well studied and trends can be drawn from previous literature. The mixed-integer linear programming (MILP) model is predominantly used (1; 2; 3), which is then extended to Fuzzy MILP to take into account inherent uncertainty present in supply chains (4). Fundamental LP 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 being the most commonly used. This shift indicates a recognition of the limitations of traditional methods, such as their inability to handle the complexity or non-linearity of relationships and high levels of uncertainty.

Due to large operational uncertainty in supply chain management, the application of MOO requires DMOO under uncertainty to effectively capture the system dynamics DMOO coupled with uncertainty becomes even more intricate due to the need for continuous, real-time adaption to changing conditions. This introduces a need for adaptive decision-making in response to evolving conditions whilst balancing conflicting objectives as the dynamics change over time. In supply chain management, operational-level decisions require a fast response to real-time changes, disruptions, and uncertainties. The real-time, fast response makes the DMOO problem even more challenging. Reinforcement learning (RL), a subset of machine learning, automates goal-driven decision-making by allowing agents to learn through trial-and-error interactions with their environment. Several common trends were observed in literature where RL outperforms traditional methods due to its ability to perform in uncertain environments (7; 8; 9). The integration of RL with efficient MOO algorithms offers a promising synergy. RL can enhance the efficiency of MOO algorithms and shift the computational costs offline for fast, dynamic decision-making in uncertain environments.

Moreover, the majority of 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. Therefore, there is a need for more comprehensive studies that consider a broader range of objectives, potentially employing MOO techniques.

To train RL agents, gradient-based methods such as A2C, A3C, PPO, and DDPG are used given their effectiveness in optimizing single objectives (8). However, this overlooks the potential benefits of data-driven methods which may be more suitable for complex, multi-objective problems. Although derivative-based methods are effective for single-objective optimization, the complexity of real-world supply chain problems, which often involve multiple conflicting objectives and high levels of uncertainty, may necessitate the use of alternative methods. In the context of multi-objective optimization, balancing proximity to the Pareto front with diversity across solutions is also crucial for obtaining a comprehensive and well-distributed set of Pareto optimal solutions.

As previously mentioned, multi-objective evolutionary algorithms (MOEAs) have been used to address multi-objective problems. These algorithms are a subset of EAs, designed to handle problems with multiple conflicting objectives. MOEAs present two key features that make them synergistic with MOO and RL; **population-based search**: MOEAs maintain a population of candidate solutions over multiple generations - this approach addresses the exploration paradigm, and they are built to **handle non-differentiable objective functions**.

Therefore, we propose integrating MOEAs strategies with RL methods to focus on finding a set of adaptable policies capable of responding to disruptions in the system. Harnessing neural networks (NNs) as policies, we diverge from conventional gradient-based methods, choosing instead the robust MOEAs to fine-tune our neural network.

The synergy of RL and MOEAs is motivated by:

- **RL's representational proficiency**: policy net is capable of adeptly capturing the uncertainty inherent in supply chain dynamics (5).
- **MOEA's capability on MOO**: MOEAs provide a robust, data-driven, derivative-free search strategy, searching across a broad solution landscape (5) .

Our approach focuses on identifying a set of decision-making policies for an agent that effectively balances multiple conflicting objectives and manages uncertainty. Rather than evolving the policies over time, we dynamically switch between pre-defined policies based on their suitability to the current environment and objectives.

The rest of the paper is organized as follows: Section 2 introduces the preliminaries, Section 3 introduces the methodology, Section 4 introduces the problem statement, and Section 5 focuses on two different inventory management case study examples.

2. PRELIMINARIES

2.1 Introduction to Reinforcement Learning

Markov Decision Processes (MDPs) provide a mathematical framework for modeling and solving sequential decision-making problems, forming the basis for RL. A finite MDP is a tuple $\langle S, A, P, R, \gamma \rangle$ which encompasses the state space, $\mathbb{S} \subseteq \mathbb{R}^{n_s}$, action space $\mathbb{A} \subseteq \mathbb{R}^{n_a}$, the state transition probability distribution, $P_{ss} : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to \mathbb{R}$, reward function $R : \mathbb{S} \times \mathbb{A} \to \mathbb{R}$ and discount factor $\gamma \in [0, 1]$. Finally, policy π is any function that maps states to actions $\pi : S \to A$. The optimal policy is found by maximizing the expected sum of rewards over a time horizon.

In RL, the agent observes the current state $s_t \in \mathbb{S}$ and chooses an action $a_t \in \mathbb{A}(s)$ with probability given by the policy $\pi(a_t \mid s_t)$. Given s_t and a_t, the agent transition into the next state $s_{t+1} \in S$ with probability given by the state transition probability function $P(s_t, a_t, s_{t+1})$ and receives a reward $R_{t+1} = \mathcal{R}(s_t, a_t, s_{t+1})$ where $R_{t+1} \in \mathcal{R} \subseteq \mathbb{R}$ where $\mathcal{R}$ is a reward function. Through trial and error, the agent finds an optimal policy π^* by maximizing the expected sum of rewards over a time horizon defined as:

$$\mathbb{E}_\pi[r^0 + \gamma r^1 + \gamma^2 r^2 + \cdots] \equiv \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \gamma^t R(s_t, a_t, s_{t+1}) \right]$$

In practice, the policy is parameterized by a policy function such that $\pi^* \approx \pi^*(\mathbf{a}|\mathbf{s}; \theta)$, where $\theta \in \mathbb{R}^{n_\theta}$.

2.2 Introduction to MOEAs

EAs such as genetic algorithms (GAs) are a population-based methods that generate a diverse set of solutions. This not only allows simultaneous evaluation of a set of solutions, but is crucial for MOO, as the evolving population inherently forms a Pareto front. GAs are well-suited to handle complexities such as non-differentiable or discontinuous functions, as well as naturally balance the exploitation of new solutions and the exploration of promising ones. This balance is facilitated through the

selection of Pareto-dominated solutions from the current population, and the subsequent reproduction of offspring solutions from the selected set.

In this research, we utilize the AGE-MOEA due to its proficiency in accurately estimating the geometry of the Pareto front(10). Upon the foundation of the renowned method NSGA-II, this algorithm is distinguished by its innovative selection mechanism, which significantly improves the representation of solutions(10; 11). Its effectiveness in delivering both accurate and efficient solutions underscores its suitability for complex scenarios, such as those encountered in supply chain management, which makes it an ideal choice for our multi-objective, derivative-free RL frameworks.

3. METHODOLOGY

3.1 RL-MOEA

The supply chain network examined in this study is modelled as a Multi-Objective Fully Observable MDP (MO MDP) similar to the single objective fully observable MDP defined in section 2.1. A MOMDP is a tuple $\langle S, A, P, R, \gamma \rangle$ which encompasses the state space, $\mathbb{S} \subseteq \mathbb{R}^{n_s}$, action space $\mathbb{A} \subseteq \mathbb{R}^{n_a}$, the state transition probability distribution, $P_{ss} : \mathbb{S} \times \mathbb{A} \times \mathbb{S} \to \mathbb{R}$, a vector-valued reward function $\mathbf{R} : \mathbb{S} \times \mathbb{A} \to \mathbb{R}^d$, where d is the number of objectives, $d \geq 2$, and discount factor $\gamma \in [0, 1]$. The primary difference is the vector-valued reward function $\mathbf{R}$, which is the length of the number of objectives and gives a numerical reward for each of the objectives. Finally, policy $\pi \in \Pi$ where π maps states to actions, $\pi : S \to A$, and Π is a set of all possible policies. In contrast to single objective MDPs, the value function is also a vector-value conditioned on the number of objectives, d. The value function, $\mathbf{V}^\pi \in \mathbb{R}^d$, is equal to $\mathbf{V}^\pi(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \mathbf{r}_{t+1} | \pi, s_t = s]$ where $\mathbf{r}_{t+1} = \mathbf{R}(s_t, a_t, s_{t+1})$. Contrary to single-objective MDPs, it is possible to encounter a situation where for objectives i and j, and policies π and π', $V_i^\pi > V_i^{\pi'}$ and $V_j^\pi < V_j^{\pi'}$, both hold true. Therefore, for solutions to MO MDPs, we have a set of possible optimal value vectors and policies. (6; 14)

In this work, we leverage the efficiency of MOEAs in MOO to directly optimize the parameters of the policies (13) and build a Pareto set of policies that are adaptable and dynamic to a series of conflicting objectives. Our methodology has the following steps as shown in Figure 1 and Algorithm 1 :

(1) **Initialization**. A population of policies is randomly generated. Each policy represents a set of parameters for a neural network (NN).
(2) **Evaluation** Each policy in the population is evaluated based on its performance across multiple conflicting objectives.
(3) **Non-dominated Sorting**. The evaluated policies are sorted into different fronts based on their dominance relationships. Solutions in the first front are non-dominated by any other solution, those in the second front are dominated only solutions in the first front and so on. This helps identify Pareto optimal solutions.

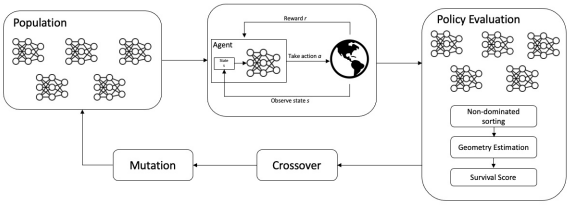

Fig. 1. An overview of the RL-MOEA algorithm proposed as seen in Algorithm 1.

(4) **Geometry Estimation**. This helps guide the evolutionary search towards regions of interest on the Pareto front and improve the diversity of solutions.
(5) **Selection and Reproduction**. Solutions in the Pareto front are selected for reproduction based on their survival score. These undergo crossover and mutation to generate offspring policies.
(6) **Survival Selection**. The offspring policies, along with some parent policies, are selected to form the next generation population based on their survival scores. This ensures the population evolves towards better solutions while maintaining diversity.
(7) **Termination Criteria**. The evolutionary process continues for a certain number of generation or until some termination criteria are met, such as convergence of the Pareto front.
(8) **Pareto Front Identification**. After the evolutionary process is complete, the non-dominated solutions remaining in the final population represent the undominated Pareto front set of policies.

Algorithm 1 RL-MOEA

Require: M: Number of objectives
Require: N: Population size
Ensure: Pareto set of policies π
  $\pi \leftarrow$ RANDOM-POPULATION(N)
  while not stop_condition **do**
    for $n = 1, ..., N$ **do**
      Reset initial state $s_{t=0}^n$
      for $t = 0, ..., T$ **do**
        Observe current state s_t^n, select an action a_t^n through the selection policy π, observe next state s_{t+1}^n, calculate the reward R_t^n for each objective M. Store information.
      end for
    Optimize with AGE-MOEA. Crossover, Mutation, Evolve to get the next set of policies π
    end for
  end while
  return Pareto set of policies π

Our Pareto Front (PF) is an undominated set of policies where $PF(\Pi) = \{\pi \in \Pi | \nexists \pi' \in \Pi : \mathbf{V}^{\pi'} \succ_P \mathbf{V}^\pi\}$ where $\succ_P$ is the Pareto dominance relation (6). This implies that for every policy in the Pareto Front, there is no other policy that performs equal or better across all objectives (6).

A notable observation is that traditional RL methodologies often rely on gradient-based optimization strategies like the Back-propagation algorithm or Q-learning for updating the NN parameters (12). These strategies, although effective in certain scenarios, require the accu-

rate estimation of the gradient to ensure the policy is updated and improved. Therefore, these can sometimes be limited in their ability to traverse intricate solution spaces or might converge prematurely to local optima. Our methodology goes beyond traditional gradient-based methodologies, where instead of employing conventional optimization strategies for the neural network parameters within the RL framework, we leverage on MOEA strategies. Moreover, for non-linear, non-differentiable objective functions, which is the case in many real world applications, gradient-based methods face challenges such as slow convergence, sensitivity to initialization, and the potential to get stuck in poor local optima(13). Therefore, the motivation behind our methodology is to leverage the efficiency of derivative-free approaches and the advantages of traditional reinforcement learning frameworks.

4. PROBLEM STATEMENT

The sequential inventory management decision-making problem is modelled as a multi-objective Fully Observable MDP as described in Section 3. The system consists of three fundamental nodes: manufacturer, wholesaler, and retailer. These nodes are interconnected with predetermined distances. For each node at each time step in the simulation, it's crucial to establish:

- Replenishment order quantity;
- Transportation mode from each node's supplier.

In alignment with the perspective on inventory management, our 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.

4.1 Mathematical Formulation

$$\max \sum_{m=1}^{M}\sum_{t=1}^{T} P^m S_r^m[t] - C^m o_r^m[t] - T^m L^{m_u} O^r[t] - I^m i^m[t] - B^m b^m[t], \quad \forall m, \forall t, \quad (1)$$

$$\min \sum_{m=1}^{M}\sum_{t=1}^{T} E^m L^{m_u} o^T[t], \quad \forall m, \forall t, \quad (2)$$

$$\min \sum_{m=1}^{M}\sum_{t=1}^{T} \tau_r^m[t], \quad \forall m, \forall t, \quad (3)$$

$$i^m = i_0^m[t] - s_T^m[t] + a_r^m[t], \quad \forall m, \forall t, \quad (4)$$

$$b^{m_d}[t] = b_0^{m_d}[t] - s_r^{m_d}[t] + d_r^{m_d}[t], \quad \forall m, \forall d \in D_m, \quad (5)$$

$$s_r^{m_d}[t] \leq b_0^{m_d}[t] + d_r^{m_d}[t], \quad \forall m, \forall t, \forall d \in D_m, \quad (6)$$

$$s_r^m[t] \leq i_0^m[t] + a_r^m[t], \quad \forall m, \forall t, \quad (7)$$

$$a_r^m[t] = s_r^{m_u}\left[t - \tau_r^m\right], \quad \forall m \neq 1, \forall t \geq \tau_r^m, \quad (8)$$

$$a_r^1[t] = s_r^1\left[t - \tau_r^1\right], \quad \forall t \geq \tau_r^1, \quad (9)$$

$$d_r^{m_d}[t] = o_r^d, \quad \forall m, \forall d \in D_m, \quad (10)$$

$$d_r^m[t] = c^m[t], \quad \forall m \in C, \forall t, \quad (11)$$

$$o_r^m[t] \leq O_{r_{\max}}^m, I^m[t] \leq I_{\max}^m, \quad \forall m, \forall t. \quad (12)$$

The goal is to ascertain the optimal action for each node m during each time period t spanning over a total of T time periods within a discrete-time setup. For each node at each time step in the simulation, the agent is subject to two continuous actions: (1) Replenishment order quantity; and (2) Transportation mode from each node's supplier.

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, 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.

Equations (1), (2), and (3) correspond to the aforementioned objectives, respectively. Equations (4) and (5) describe the evolution of inventory and backlog over time. Equations (6) and (7) limit the quantity a node can ship downstream. Equations (8) and (9) relate to the acquisition of goods at the regular nodes and the root node, respectively, with the lead time indicating the duration required for goods production/transportation. Equations (10) and (11) outline the relationship between demand and the re-order quantity from downstream nodes (or customers). Equation (12) denotes the storage capacity constraint and re-order constraint.

4.2 Sources of Uncertainty

In our supply chain environment, two primary sources of uncertainties are present, allowing for a more comprehensive and realistic representation of real-world scenarios.

- *Customer Demand:* Assumed to follow a Poisson distribution, which is frequently used to model random events and independent customer demand patterns.
- *Transportation Time:* Modeled with a uniform distribution to capture the unpredictability of lead time.

5. CASE STUDIES

Using RL-MOEA, we obtained a Pareto set of policies, with each targeting our three main objectives in varying degrees. With this broad spectrum of policies, decision-makers gain the freedom to adapt their strategies according to real-time changes in the supply chain landscape. Future work should consider baselines against algorithms such as (14)

To demonstrate the effectiveness of our proposed method, a multi-echelon supply chain problem is presented with two case studies. In each case, we compared the performance of two distinct strategies under irregular conditions in our simulated environment:

- **Steady Policy:** Consistently employs a multiobjective (MO) policy, targeting all three objectives throughout the simulation, regardless of disturbances.
- **Adaptive Policy:** Initiates with the MO policy. When anomalies arise in the environment, it dynami-

cally switches to another policy from the Pareto front to better address the immediate challenge.

5.1 Case study 1 - Mitigating Surge Demand

In the global economy, unexpected events like the COVID-19 outbreak can result in a sudden and significant increase in demand for certain products. This rapid rise puts immense pressure on supply chains globally, leading to product shortages, longer delivery times, and significant financial losses, namely the bullwhip effect.

In this case study, we simulate a scenario where customer demand surges to five times its usual rate in a certain time period, deviating from the typical Poisson distribution. An anticipated significant backlog could lead to reduced profits and longer lead times. In response, our adaptive strategy is designed to temporarily shift its focus towards maximizing profit and reducing lead times under such conditions

The simulation results illustrate the dynamics of the two policies, as depicted in Fig. 2.

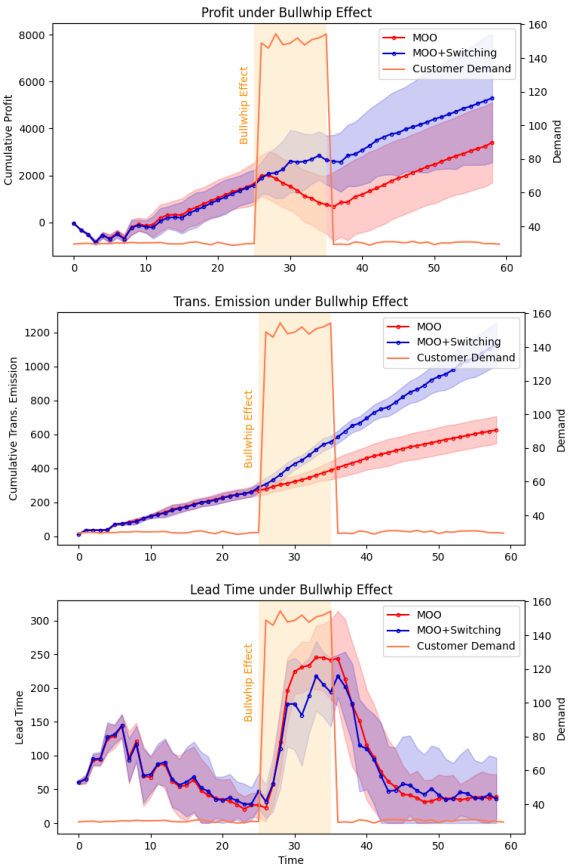

Fig. 2. Dynamics of **cumulative profit**, **cumulative transportation emission**, and **lead time** under demand surge scenario.

(1) **Dynamics during surge**: For the steady strategy, a decline in cumulative profit and spike in lead time is observed during t=25˜35. To counteract this, the adaptive strategy switches to faster transportation modes and implements a more frequent re-order strategy. While this choice incurs a higher transportation cost, the significant reduction in backlog and stock-out costs offsets this. This strategy not only stabilizes cumulative profit but also ensures a consistent lead time, facilitating timely delivery of orders.
(2) **Emission Trade-offs**: Although faster transportation modes can benefit cumulative profits and lead times, they lead to higher transportation emissions, highlighting the environmental trade-off.
(3) **Post-Demand Surge Dynamics**: After t=35, the policy prioritizes profit and lead time optimization to address the demand disruptions, regardless of the impact on transportation emissions.

The findings from this simulation emphasize the agility and resilience offered by policy switching in the face of unexpected demand surges. While it offers considerable advantages in terms of cumulative profit and lead time, it also highlights the environmental trade-offs of such decisions.

5.2 Case study 2 - Countering Emission Penalties

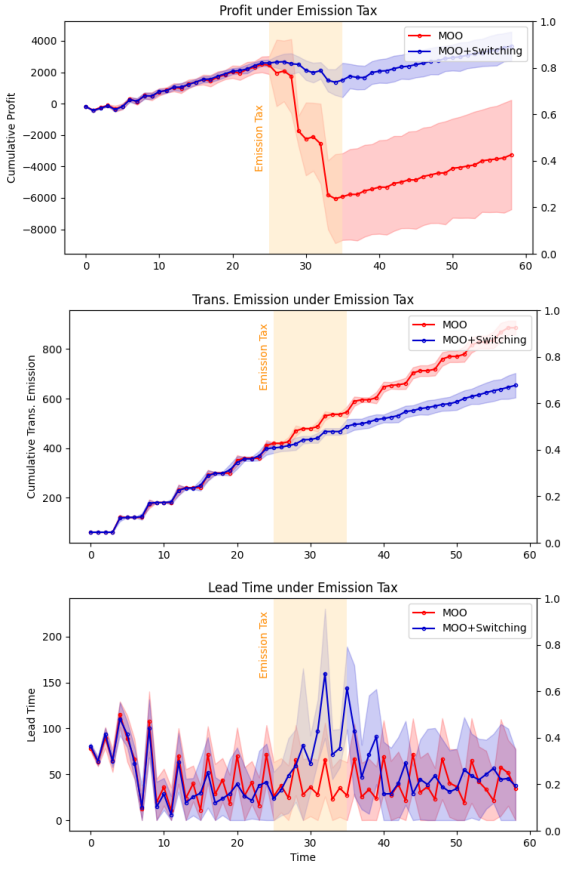

Fig. 3. Dynamics of **cumulative profit**, **cumulative transportation emission**, and **lead time** under emission tax scenario.

In the face of growing environmental concerns, governments are progressively implementing environmental regulations to the businesses to mitigate the impacts of climate change. This not only directly impacts profitability but also requires firms to re-evaluate and adapt their operational strategies to align with environmental goals.

In this scenario, we simulate this evolving environmental concern by introducing an emission tax over a certain duration. When emission of a time step exceeds a predefined threshold, penalties are applied to reflect the real-world financial implications of non-compliance with environmental standards. It's expected that the emission tax will result in considerable profit downturns. In response, our adaptive strategy should not only aim to shield profits but also reduce emissions.

The simulation results of the two policies in this scenario are illustrated in Fig.3.

(1) **Cumulative Profit and Emission Dynamics:** The emission tax results in substantial profit losses due to higher emissions. The adaptive strategy shifts to slower transportation modes and less frequent reorder schedules. Though leading to higher backlog, these adjustments directly reduce emission levels, mitigating the impact of heavy taxes.
(2) **Lead Time Response**: Slower transportation chosen by the adaptive policy results in increased lead time.
(3) **Post-Tax Dynamics:** After t=35, the adaptive strategy remains profit-inclined to offset the severe profit loss due to the effect of emission tax.

Unlike the previous case, this scenario introduces an additional term to the objective function with an emission tax imposition. As shown, the adaptive strategy effectively balances between emission considerations and profitability, highlighting the versatility of our dynamic approach. Throughout both case studies, regardless of the specific disruptions encountered, our method consistently demonstrates its adaptability and robustness across a variety of potential supply chain disruptions.

6. CONCLUSIONS AND FUTURE WORK

This work introduces a novel strategy that leverages MOEA and RL frameworks, aiming to procure a Pareto policy set that focuses on three objectives, namely profitability, environmental impact, and lead-time efficiency. The assortment of policies equips decision-makers with the agility to adapt to fluctuations within the supply chain, thus enhancing stability in the face of irregularities. The robustness of our approach is validated through two case studies, showcasing the efficacy of our adaptive policy strategy across a spectrum of challenges from demand volatility, cost coefficient changes to objective functions modification. The comparative analysis indicates that our dynamic policy adaptation surpasses the performance of conventional static policies, underscoring its enhanced resilience and adaptability.

For future work, we aim to improve the scalability and specificity of our supply chain model. Methodologically, we plan to explore diverse neural network architectures like convolutional neural networks (CNNs) and graph neural networks (GNNs) to provide the decision-making agent with a more comprehensive information framework. We also plan to expand the methodology to accommodate a multi-agent environment and explore the use of a partially observable MDP (POMDPs), both of which are typical scenarios encountered in supply chain contexts.

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, vol. 152, p. 107370. 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] S. E. Li, Reinforcement learning for sequential decision and optimal control. Springer, 2023.
[13] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," 2017.
[14] 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.