AI Agent Systems for Supply Chains: Structured Decision Prompts and Memory Retrieval

Extended Abstract

Konosuke Yoshizato
National Institute of Advanced Industrial Science and Technology
Tsukuba, Japan
Nagoya Institute of Technology
Nagoya, Japan
yoshizato.konosuke@otsukalab.nitech.ac.jp

Kazuma Shimizu
National Institute of Advanced Industrial Science and Technology
Tsukuba, Japan
NEC Corporation
Tokyo, Japan
smzkzm2019@nec.com

Ryota Higa
National Institute of Advanced Industrial Science and Technology
Tsukuba, Japan
NEC Corporation
Tokyo, Japan
r-higaryouta@nec.com

Takanobu Otsuka
National Institute of Advanced Industrial Science and Technology
Tsukuba, Japan
Nagoya Institute of Technology
Nagoya, Japan
otsuka.takanobu@nitech.ac.jp

ABSTRACT

This paper studies large language model (LLM)-based multi-agent systems (MASs) for inventory management. Although these systems have gained attention for their potential to overcome limitations of conventional approaches, it remains unclear whether they can reliably produce optimal ordering policies and adapt across diverse operating situations. To address these questions, we evaluate an LLM-based MAS with a fixed-ordering strategy prompt that specifies stepwise procedures and a safety-stock strategy, and we find that it derives the optimal ordering policy in a restricted scenario. To enhance adaptability, we introduce AIM-RM, which retrieves similar past experiences. Our empirical results indicate that AIM-RM outperforms existing benchmark methods across various supply chain scenarios, showing robustness and adaptability. An extended version of this paper is available at: http://arxiv.org/abs/2602.05524.

KEYWORDS

Supply Chain Management; Inventory Management; LLMs; Multi-agent systems

ACM Reference Format:
Konosuke Yoshizato, Kazuma Shimizu, Ryota Higa, and Takanobu Otsuka. 2026. AI Agent Systems for Supply Chains: Structured Decision Prompts and Memory Retrieval: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/UJUM3065

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/UJUM3065

1 INTRODUCTION

Multi-agent systems (MASs) are well suited to sequential decision-making among coordinating autonomous entities. Recent advances in large language models (LLMs) have strengthened the abilities of AI agents, expanding the practical applicability of MASs across domains [7, 8, 12, 17]. Among these domains, supply chain management (SCM) has emerged as a prominent area for LLM-based MASs, and research applying such systems to SCM has grown rapidly [4–6, 11, 15]. In this paper, we focus on inventory management, a core SCM function that supports procurement, production, and distribution throughout the supply chain.

A limitation of prior research on inventory management is that heuristic methods often require scenario-specific tuning [3], while reinforcement learning (RL) can be computationally expensive [1]. By contrast, LLM-based MASs have gained attention for their potential to overcome limitations of existing methods. Therefore, assessing whether current LLM foundation models can independently address inventory management is an important first step toward exploring AI agents for supply chain optimization.

This study makes two contributions. First, we evaluate LLM-based MASs for inventory management across diverse supply chain scenarios. We show that under a specific condition, a fixed prompt can achieve an optimal ordering policy without scenario-specific tuning [9, 13, 14, 18]. Second, to enhance adaptability across diverse scenarios, we design an adaptive LLM-based MAS that retrieves similar past experiences from historical transaction data to condition decisions, coordinate agents implicitly, and improve robustness across scenarios.

2 METHODOLOGY

2.1 Multi-Echelon Inventory Management

We define the multi-echelon inventory management problem studied in this paper. We consider an M-tier supply chain indexed by

Table 1: This table shows the average total rewards in 5 episodes. The rewards are expressed as the relative gap $\Delta = |(\mathrm{Opt}-r)/\mathrm{Opt}|$, where Opt is the optimal value and r is the average total reward over all tiers and periods. The bolded numbers represent the best scores among models excluding reinforcement learning models (IPPO, MAPPO).

Model (Reasoning effort: Medium)	Const-Uni	Dec-Div	Dec-Uni	Inc-Div	Inc-Uni	Average
InvAgent (w/ step desc)	3.33	79.22	437.78	104.55	137.88	152.55
InvAgent (w/ step desc and SS strategy)	**0.00**	115.06	315.56	264.88	425.00	224.10
AIM-RM (w/o RL log)	69.17	66.27	219.11	162.64	175.77	138.59
AIM-RM (w/ RL log)	60.00	**56.02**	**171.11**	**95.12**	**74.09**	**91.27**
InvAgent (w/o strategy)	46.67	100.30	202.22	171.90	93.18	122.85
InvAgent (w/ strategy)	183.33	118.67	200.00	179.33	155.30	167.33
Base-Stock	146.67	140.36	340.00	162.81	112.12	180.39
Tracking-Demand	200.00	150.30	584.44	205.37	243.93	276.81
IPPO	0.00	25.30	137.78	38.01	12.88	42.79
MAPPO	30.00	16.27	60.00	44.21	19.70	34.04

$m \in \{0, \ldots, M-1\}$, where $m = 0$ denotes the retailer. Each tier holds materials for production. One unit of product consumes one unit of material, and production is limited by facility capacity c_m. Finished products are not stored and are shipped immediately downstream: items produced at tier m are sent to tier $m-1$, arrive after the lead time L_m, and become its inventory for subsequent production.

Next, we explain the procedure within an episode, which consists of T periods. In each period $t \in [T] = \{1, \ldots, T\}$, the following four steps are executed sequentially: (i) Materials ordered L_m periods earlier arrive and replenish inventory; (ii) Each tier m places an upstream order $O_{m,t}$ and observes downstream demand; (iii) Each tier determines a shipment quantity $R_{m,t}$ after which the remaining inventory becomes $I_{m,t}$, and any unmet demand is recorded as backlog $B_{m,t}$; (iv) The reward $P_{m,t}$ is calculated based on sales revenue, ordering cost, backlog penalties, and inventory holding cost. Each agent observes a state $s_{m,t}$ summarizing inventory and backlog conditions and temporal information needed for decision-making.

2.2 Proposed Agent

Our proposed agent comprises two modules: a decision module and a memory module. The decision module uses an LLM guided by a decision prompt $\mathcal{P}_{\mathrm{DM}}$ that takes $(s_{m,t}, \mathcal{H}, \mathcal{D})$ as inputs, where $\mathcal{H}$ denotes historical data and $\mathcal{D}$ denotes a natural-language description of customer demand. Given these inputs, the decision module outputs an order quantity together with its rationale. To enhance order quantity decisions, we use a step-by-step description, denoted by $\mathcal{P}_{\mathrm{SD}}$, to make the procedure explicit. In addition, we augment $\mathcal{P}_{\mathrm{DM}}$ with a safety-stock (SS) strategy that determines the order quantity by jointly considering inventory during the lead time and forecasted demand, using an SS level to buffer uncertainty.

To enhance adaptability, we provide a memory module that maintains past experiences $\mathcal{M}_m = \{(\phi(s_{m,t}), O_{m,t}, P_{m,t})\}_t$, where ϕ maps each state $s_{m,t}$ to a vector representation. To efficiently leverage the memory function, AIM-RM performs a similarity search in the embedding space by computing Euclidean distances between the current state embedding and the stored embeddings. Experiences whose distances are below a predefined threshold are treated as similar and retrieved with up to K of the closest matches.

3 EXPERIMENTS AND RESULTS

We conduct five deterministic scenarios combining two supply-chain settings (*uniform*, *diverse*), where *uniform* uses identical tier parameters and *diverse* uses tier-dependent parameters, and three demand patterns: *constant* (D_t=4), *increasing* (D_t=2 + $\lceil t/3 \rceil$), and *decreasing* (D_t=2 + $\lceil (12-(t-1))/3 \rceil$).

We evaluate four distinct agent configurations. Two configurations do not use the memory module: InvAgent (w/ step desc) uses $\mathcal{P}_{\mathrm{SD}}$, whereas InvAgent (w/ step desc and SS strategy) uses $\mathcal{P}_{\mathrm{DM}}$ with both $\mathcal{P}_{\mathrm{SD}}$ and $\mathcal{P}_{\mathrm{SS}}$. The remaining two configurations involve the memory module: AIM-RM (w/o RL log) starts without any preloaded historical logs but stores logs during an episode, while AIM-RM (w/ RL log) is initialized with historical logs derived from the IPPO evaluation data. Both AIM-RM (w/o RL log) and AIM-RM (w/ RL log) use $\mathcal{P}_{\mathrm{DM}}$ with $\mathcal{P}_{\mathrm{SD}}$ and a memory usage prompt $\mathcal{P}_{\mathrm{MU}}$ that indicates how to use the retrieved similar cases.

We provide three categories of benchmarks: LLM benchmarks, heuristic benchmarks and RL benchmarks. The LLM benchmarks are InvAgent (w/o strategy) and InvAgent (w/ strategy), which are proposed in [12]. Heuristic benchmarks are Base-Stock and Tracking-Demand. The RL benchmarks are IPPO [2] and MAPPO [16].

Results. Table 1 shows the results obtained with OpenAI o4-mini under medium reasoning effort [10]. The safety-stock strategy performs well under the constant demand scenario but degrades markedly under time-varying demand scenarios, reflecting its limited ability to track demand trends. In contrast, AIM-RM (w/ RL log) ranks first among the compared models in all experimental scenarios and achieves average rewards comparable to the RL benchmark models IPPO and MAPPO. Furthermore, comparing AIM-RM (w/ RL log) with AIM-RM (w/o RL log) demonstrates that preloading prior memories consistently improves performance. This indicates that AIM-RM (w/ RL log) can effectively leverage retrieved experiences, which supports indirect coordination among distributed agents and coherent system-wide behavior by enabling agents to reference consistent simulated results. Overall, these results show that AIM-RM (w/ RL log) offers robust and practically useful adaptation across diverse supply chain conditions.

REFERENCES

[1] Robert N Boute, Joren Gijsbrechts, Willem Van Jaarsveld, and Nathalie Vanvuchelen. 2022. Deep Reinforcement Learning for Inventory Control: A Roadmap. *European journal of operational research* 298, 2 (2022), 401–412.

[2] Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. 2020. Is Independent Learning All You Need in The StarCraft Multi-Agent Challenge? *arXiv preprint arXiv:2011.09533* (2020).

[3] Kevin Geevers, Lotte Van Hezewijk, and Martijn RK Mes. 2024. Multi-echelon inventory optimization using deep reinforcement learning. *Central European Journal of Operations Research* 32, 3 (2024), 653–683.

[4] Valeria Jannelli, Stefan Schoepf, Matthias Bickel, Torbjørn Netland, and Alexandra Brintrup. 2024. Agentic LLMs in the Supply Chain: Towards Autonomous Multi-Agent Consensus-Seeking. *arXiv preprint arXiv:2411.10184* (2024).

[5] Samuel N Kirshner, Yiwen Pan, Jason Xianghua Wu, and Alex Gould. 2025. Talking terms: Agent information in LLM supply chain bargaining. *Decision Sciences* (2025).

[6] Beibin Li, Konstantina Mellou, Bo Zhang, Jeevan Pathuri, and Ishai Menache. 2023. Large Language Models for Supply Chain Optimization. *arXiv preprint arXiv:2307.03875* (2023).

[7] Yuan Li, Bingqiao Luo, Qian Wang, Nuo Chen, Xu Liu, and Bingsheng He. 2024. CryptoTrade: A Reflective LLM-based Agent to Guide Zero-shot Cryptocurrency Trading. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1094–1106. https://doi.org/10.18653/v1/2024.emnlp-main.63

[8] Guannan Liang and Qianqian Tong. 2025. LLM-Powered AI Agent Systems and Their Applications in Industry. *arXiv preprint arXiv:2505.16120* (2025).

[9] Lilian Ngweta, Kiran Kate, Jason Tsay, and Yara Rizk. 2025. Towards LLMs Robustness to Changes in Prompt Format Styles. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*. 529–537.

[10] OpenAI. 2025. OpenAI o3 and o4-mini System Card. https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf

[11] Yinzhu Quan and Zefang Liu. 2024. Econlogicqa: A Question-Answering Benchmark for Evaluating Large Language Models in Economic Sequential Reasoning. *arXiv preprint arXiv:2405.07938* (2024).

[12] Yinzhu Quan and Zefang Liu. 2024. Invagent: A Large Language Model based Multi-Agent System for Inventory Management in Supply Chains. *arXiv preprint arXiv:2407.11384* (2024).

[13] Abel Salinas and Fred Morstatter. 2024. The Butterfly Effect of Altering Prompts: How Small Changes and Jailbreaks Affect Large Language Model Performance. In *Findings of the Association for Computational Linguistics ACL 2024*. 4629–4651.

[14] Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations*.

[15] Haojie Wang, Jiuyun Jiang, L Jeff Hong, and Guangxin Jiang. 2025. LLMs for Supply Chain Management. *arXiv preprint arXiv:2505.18597* (2025).

[16] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *Advances in neural information processing systems* 35 (2022), 24611–24624.

[17] Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, et al. 2024. A Multimodal Foundation Agent for Financial Trading: Tool-Augmented, Diversified, and Generalist. In *KDD 2024-Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 4314–4325.

[18] Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine A Heller, and Subhrajit Roy. 2024. Batch Calibration: Rethinking Calibration for In-Context Learning and Prompt Engineering. In *The Twelfth International Conference on Learning Representations*.