Optimizing inventory management using a multi-agent LLM system

Zhihong Li

South China University of Technology, Guangzhou, China, bmzhhli@scut.edu.cn

Albaraa Ksibi

South China University of Technology, Guangzhou, China, baraa.k@live.com

Xiaoying Xu

South China University of Technology, Guangzhou, China, bmxyxu@scut.edu.cn

Li, Z., Ksibi, A., & Xu, X. (2024). Optimizing inventory management using a multi-agent LLM system. In Li, E.Y. *et al.* (Eds.) *Proceedings of The International Conference on Electronic Business, Volume 23* (pp. 308-318). ICEB'24, Zhuhai, China, October 24-28, 2024.

Optimizing Inventory Management using a Multi-Agent LLM System

Zhihong Li [1]
Albaraa Ksibi [2,]
Xiaoying Xu [3,*]

*Corresponding author
[1] South China University of Technology, Guangzhou, China, bmzhhli@scut.edu.cn
[2] South China University of Technology, Guangzhou, China, baraa.k@live.com
[3] South China University of Technology, Guangzhou, China, bmxyxu@scut.edu.cn

ABSTRACT

Effective inventory management requires a comprehensive capability to forecast demand and optimize stock levels, traditionally reserved for human expertise. Emerging AI methods, while providing effective solutions through deep learning models and data analytics, often lack the flexibility to incorporate dynamic market insights and real-time data. By leveraging the diverse capabilities of multiple dynamically interacting large language models (LLMs), we can overcome these limitations and develop a new class of AI-driven inventory management systems. This paper presents a multi-agent framework comprising a project manager agent, a sales forecasting agent, and an inventory manager agent, which autonomously collaborate to address inventory management challenges. The agents dynamically adjust inventory plans and maintain product availability through self and mutual corrections. Simulation results demonstrate a significant increase in the inventory turnover ratio, reduced shipping costs and holding fees, and a substantial decrease in total cost, all while maintaining a zero stockout rate. Our framework showcases the potential of synergizing the intelligence of LLMs, the precision of statistical modeling, and the dynamic collaborations among diverse agents, opening novel avenues for automating and optimizing supply chain management.

Keywords: Inventory management, multi-agent systems, large language models (LLMs), supply chain optimization.

INTRODUCTION

Inventory management is a critical component of supply chain operations, impacting the efficiency and profitability of businesses (Chukwuemeka et al., 2013). Effective inventory management ensures that products are available to meet customer demand without overstocking, which can lead to increased holding costs. Traditionally, inventory management has relied on statistical methods and human decision-making (Nemtajela et al., 2016), but recent advancements in artificial intelligence (AI) have opened new avenues for optimizing these processes (Giannoccaro et al., 2002; Partovi et al., 2002; Kara et al., 2018).

Traditional approaches to inventory management often require deep technical expertise and extensive experience to perform effectively (Wild, 2017). For instance, developing a robust demand forecasting model demands not only knowledge of statistical and machine learning techniques, but also a capability for programming, debugging, and post-processing within certain software environments. Therefore, inventory management tasks have mainly been reserved for human experts.

An area of interest has been the development of AI-based strategies, which have proven effective in providing efficient approaches to solving complex problems, such as predicting demand patterns, optimizing stock levels, and conducting real-time inventory adjustments (Goli et al., 2018; Lingam et al., 2018). These methods often involve the use of surrogate models or structural deep-learning methods. However, their knowledge is typically baked into the architecture during the training process, offering less flexibility when it comes to generating, collecting, and incorporating new data or insights.

Recently, large language models (LLMs) based on attention mechanisms and transformer architectures have emerged as novel tools for various applications (Chang et al., 2024), including inventory management. For example, the specific class of Generative Pretrained Transformer (GPT) models, such as the ChatGPT and GPT-4 models reported by OpenAI, have demonstrated unprecedented capability in mastering human language, coding, logic, reasoning, and related natural language tasks (Wu et al., 2023). Recent studies have shown that such complex LLMs excel in programming numerical algorithms or debugging code errors in various coding languages (Liu et al., 2023). Based on their conversational capabilities, LLMs have been used to power conversable AI agents to go from AI-human conversations to AI-AI or AI-tools interactions for autonomy. For instance, given a statement of a specific task, AI agents can attempt to break complex problem statements into subtasks and use tools, including data retrieval from the internet, to solve them step-by-step through automatic iterations (Rasal et al., 2024).

Combining these advances, this paper investigates whether, and how, we can utilize the language capabilities, both in natural and coding languages, of LLM-powered agents to solve inventory management problems with reduced or little human intervention. This involves a broad set of tasks, including data analysis, demand forecasting, inventory level optimization, and interactive feedback with human users. To achieve this goal, we propose the use of a multi-agent framework. This multi-agent framework consists of a project manager agent, a sales forecasting agent, and an inventory manager agent. These agents are organized into collaborative teams to apply knowledge of inventory management and supply chain principles to solve problems with minimal human intervention. Through self-correction and mutual corrections within the agent team, our collaborative framework can address various inventory management challenges, such as maintaining optimal stock levels, predicting sales trends, and ensuring just-in-time inventory. Our multi-agent modeling framework demonstrates a promising possibility of moving beyond a human-only paradigm for solving inventory management problems, thus opening a novel avenue for automation of problem-solving in supply chain management and preparing for the coming era of human-AI collaboration in business operations.

The plan of this paper is as follows. First, we introduce a strategy for profiling individual agents and organizing them into collaborative teams. Then, we demonstrate the capabilities of the team via an inventory management problem of a product sold on Amazon. We explore a division of labor strategy with the multi-agent group. We conclude the study with a discussion of the results and a broader elaboration of the limitations and potential of agent strategies.

LITERATURE REVIEW

Inventory Management

Inventory management has long been a critical function within supply chain management, with various traditional approaches employed to optimize stock levels. Traditional models like the Economic Order Quantity (EOQ) and Reorder Point (ROP) focus on mathematical formulations to determine optimal order quantities and reorder levels, as initially discussed by Harris (Harris, 1990). Just-in-Time (JIT) inventory management aims to reduce in-process inventory and associated carrying costs by ensuring materials and products are available only as needed (Schonberger, 1982). Vendor-managed inventory (VMI) shifts inventory management responsibility to suppliers, enhancing efficiency and reducing stockouts (Waller et al., 1999). ABC analysis, based on the Pareto Principle, helps businesses focus on the most critical items impacting inventory costs, as noted by Ramanathan (Ramanathan, 2006).

Despite their widespread use, these traditional methods often require significant manual intervention and rely on historical data, which can lead to inefficiencies in rapidly changing market conditions. Additionally, the accuracy of demand forecasts is crucial, as any deviations can result in either stockouts or excess inventory, both of which are costly for businesses.

Multi-Agent Systems and Their Applications

Multi-agent systems (MAS) are a critical subfield of artificial intelligence, they are focused on developing computational models and algorithms to simulate and control interactions among multiple autonomous agents. Recent advancements have expanded the application and understanding of MAS in various domains. Game theory, which studies strategic interactions among rational agents, has been foundational in analyzing MAS (Barron, 2024). The Nash equilibrium, a state where no agent can improve their utility unilaterally, has been applied to numerous strategic situations, including economic and social dilemmas (Zhang et al., 2024). Coevolutionary algorithms, another crucial development, model adaptive processes where agents adjust their strategies in response to others, leading to cooperation, competition, and specialization (Gen et al., 2023). MAS have significantly impacted smart grid management by optimizing energy distribution and enhancing grid stability (Ghazimirsaeid et al., 2023). The transportation sector also benefits from MAS, particularly in managing autonomous vehicles to optimize traffic flow and safety (Bastarianto et al., 2023). Recent research has also explored integrating large language models (LLMs) like GPT-3.5-turbo into MAS, enhancing agents' decision-making through real-time strategic recommendations (Lei et al., 2024). These advancements underscore MAS's potential in solving complex problems across various fields, driven by the interaction and cooperation of autonomous agents.

Large Language Models (LLMs)

Large language models (LLMs) have revolutionized natural language processing and AI with their ability to understand, generate, and manipulate human language. Models like GPT-3 and GPT-4, developed by OpenAI, have demonstrated exceptional capabilities in various tasks, including text generation, translation, summarization, and even programming (Kocoń et al., 2023). These models leverage transformer architectures, which utilize attention mechanisms to process and generate language with high accuracy and coherence.

The integration of LLMs with MAS offers a powerful combination for complex problem-solving (Sun et al., 2024). LLMs can enhance the capabilities of individual agents by providing advanced language understanding and generation, enabling them to interact more effectively with human users and other agents. Additionally, LLMs can assist in coding, debugging, and automating repetitive tasks, further increasing the efficiency of MAS in applications like inventory management.

While significant advancements have been made in both inventory management and AI, there are still several gaps that need to be addressed. Traditional inventory management methods often lack the flexibility to adapt to dynamic market conditions and require extensive manual input. Although MAS offers a promising solution, their application in inventory management is still relatively unexplored, particularly in integrating advanced LLMs to enhance their functionality.

This paper aims to fill these gaps by proposing a novel multi-agent LLM framework for inventory management. By leveraging the capabilities of LLMs, the proposed framework can automate and optimize various aspects of inventory management with minimal human intervention. This approach not only improves the efficiency and accuracy of inventory management but also provides a flexible and scalable solution that can adapt to changing market conditions

METHODOLOGY

Data Collection

The foundation of this study is built on the analysis of real sales data from Amazon. The product used is a Grinding Wheels Pack owned by a company in China. The company has granted permission to use their data for this study. The dataset spans one year, with daily records, and is divided into two parts: the first three quarters (nine months) are used for training, and the last quarter (three months) is reserved for testing and validation. Key attributes of the dataset include order date, order status (shipped or canceled), quantity, stock inventory, shipping costs, daily storage fees, minimum order quantities, and lead time. This comprehensive dataset provides the necessary inputs for both the multi-agent LLM framework and the traditional human-managed approach, allowing for a robust comparative analysis.

Framework Overview

The proposed multi-agent LLM framework aims to enhance inventory management by leveraging the capabilities of large language models (LLMs) within a multi-agent system (MAS). This framework consists of three specialized agents: the Project manager agent, the sales forecasting agent, and the inventory manager agent, as well as a supervisor (role played by a human) and a code executer (executes code on the local machine). Each agent is designed to perform specific tasks within the inventory management process, working collaboratively to ensure efficient and accurate operations.

The different role of each agent is defined via agent profiling As listed in Table 1, different profiles (Column 3) for agent roles (Column 2) are written as initial prompts to influence the LLMs' behaviors during the chat.

Table 1: Profiling for each agent in the group.

Agent Number	Agent Role	Profile
1	Project Manager Agent	You are an expert in project management. You are working with an inventory manager and a forecasting manager to ensure the project of inventory management runs correctly. If the message comes from the forecasting agent, you will inspect the sales forecasting plan and evaluate the accuracy of the proposed model, if the model is performing poorly, you will either suggest improving the model or changing it. If the message comes from the inventory manager agent, you will inspect the inventory plan and ensure that it takes into account all the factors, including inventory levels, shipping cost, storage fees, MOQ, and lead time.
2	Forecasting Agent	You are an expert in sales data analysis and demand forecasting. You are working with a project manager and a code executor to analyze and forecast sales data. Follow these steps separately: Inspect the data and know the data types. Build a forecasting model. Add safety stock to the forecasted quantity to avoid stockouts based on the accuracy of the model. Save the forecasting for every day of the next quarter.
3	Inventory Manager Agent	You are an expert in inventory management and optimization. You are working with a project manager to create an inventory plan to ensure no stock shortage or overstock while minimizing costs. Follow these steps seperately: Load and inspect the current inventory levels, shipping cost, storage fees, shipping time, MOQ, and lead time from the inventory file. Load and inspect the forecasted sales data from the forecasted sales file. Describe the code to create an inventory plan with the reorder date and reorder quantity to ensure no stock shortage or overstock, while minimizing costs. Consider shipping time and lead time, shipping cost, storage fees, and MOQ.
4	Code Executor	You will execute the code in the message and return the output.

Source: This study.

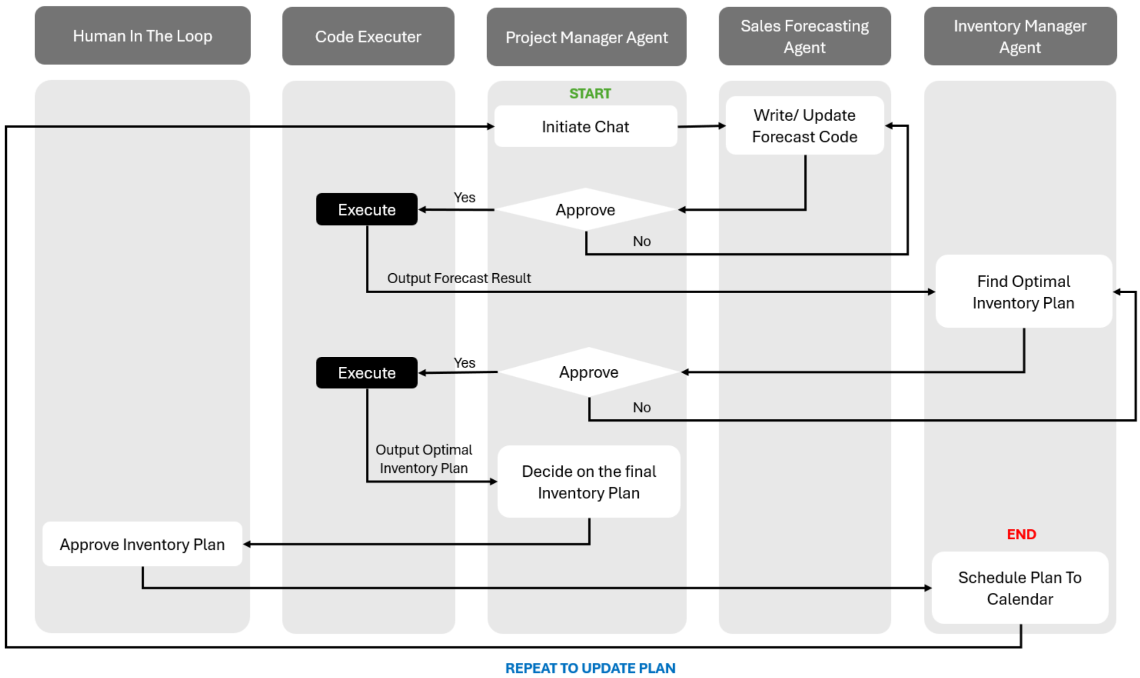
The process of the multi-agent system is shown in Figure 1, it begins with the Project manager agent initiating a chat to kickstart the inventory management process. This agent is responsible for overseeing the entire operation, ensuring smooth communication and coordination among the agents. Upon initiation, the sales forecasting agent takes charge of writing or updating the forecast code. This agent analyzes historical sales data and predicts future demand, which is crucial for determining inventory requirements. Once the forecast code is prepared, it is sent to the Code Executor for execution.

The Code Executor is responsible for running the forecast code and generating the forecast results. The forecast results are then reviewed by the Project manager agent. If the results meet the expectations and accuracy standards, the project manager agent approves them. If not, the sales forecasting agent must revise the forecast code and repeat the process, this process is repeated for a maximum of 2 rounds, to avoid reaching the token limit of the session.

Upon approval of the forecast results, the workflow transitions to the inventory manager agent. This agent is tasked with finding the optimal inventory plan based on the forecasted demand. The inventory manager agent determines the best inventory levels to meet the predicted demand while minimizing holding and ordering costs. The optimized inventory plan is then executed and outputted for review by the project manager agent.

The project manager agent reviews the optimal inventory plan. If the plan is deemed satisfactory, it is approved and scheduled into the calendar. If the plan does not meet the required standards, revisions are made to ensure it aligns with the overall inventory management strategy. This step ensures that all plans are meticulously vetted before implementation.

Finally, the approved inventory plan is scheduled, and the process is set to repeat every quarter, ensuring continuous monitoring and updating of the inventory management strategy. This cyclical process ensures that the framework remains adaptive to changing demand and market conditions, providing a robust solution for effective inventory management.



Source: This study.

Figure 1: Workflow of the multi-agent system

SIMULATION

Inventory Plan Development

To explore the potential of organized multi-agent modeling frameworks for solving inventory management problems, we present a simple experiment to demonstrate the benefits of self-correction and mutual corrections via agents' teaming strategies and collaborative approaches. We power the agents through a state-of-the-art general-purpose large language model, GPT-4o, via the OpenAI API. Each agent has its profile for initialization, as shown in Table 1, and they are organized into teams of different structures. Depending on their profiles, they can explicitly communicate with each other by sharing information or with humans using natural language; the code executor is also given access to a local environment in which they can execute code.

We demonstrate the performances of the agent team by assigning tasks corresponding to their profiles, the goal is to create an inventory plan for a product sold on Amazon, to minimize the shipping and holding costs of the product, while ensuring no stock shortages or overstocks occur. Further details can be found in the Materials and Methods section. As shown in Figure 1, we design agents that play the roles of the project manager, the sales forecaster, the inventory manager, and the code executor respectively, and organize them into a management team via autonomous, interactive, and dynamic group chatting. For example, the project manager is defined as a manager agent who assigns tasks and can provide insights when asked; the Sales Forecaster is an agent tasked to develop sales forecasts based on historical data; The inventory manager is in charge of formulating the inventory replenishment strategy using the forecasts; the code executor runs the scripts with access to a local environment and returns the outcome information to the group chat; To organize these working agents, we construct a Group-Chat Manager agent; this agent coordinates a dynamic conversation among the group by performing the following steps repeatedly:

- Choosing a speaker based on the context and the agent profiles
- Collecting the inputs from the selected speaker, and
- Broadcasting the message to the whole group.

While in principle it is possible to gather human input at various stages of the problem-solving process, when being asked during the group chat, we choose to skip and provide no additional input from humans.

Conversation 1: Sales Forecast

The project manager agent initiates the conversation and commands the forecasting agent to analyze the sales data, the following command is sent from the project manager to the forecasting agent:

Project manager (to forecasting agent):

Please analyze the sales data and recommend an appropriate forecasting method. Conduct the necessary data analysis and preprocessing steps to prepare the data for the forecasting model.

In response, the forecasting agent begins by writing code to identify the columns in the dataset and perform an exploratory data analysis (EDA) of the complete data. The code is executed and returns the following analysis:

Code executor (to project manager agent):

- Data Distribution: The `quantity` column shows a skewed distribution with a mean of 1.279 and a maximum value of 90. Most orders are likely small quantities, with a few significantly larger orders.
- No Missing Values: There are no missing values in the data.
- Outliers: The boxplot indicates potential outliers in the `quantity` data.

Based on the insights gained from the EDA, the project manager proposes the Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX) forecasting method as the most suitable approach for the sales data:

Project manager (to forecasting agent):

Let's start the analysis and forecasting process using the following suggestions: Given that we are dealing with time series data (sales over time), the SARIMAX forecasting method is the most suitable approach for the sales data.

The forecasting agent initiates the process. Utilizing the advanced capabilities of GPT-4, the forecasting agent develops an initial script to predict sales for the upcoming quarter and evaluate the model's performance. Upon executing this initial model by the code executor, the evaluation results are forwarded to the project manager agent. The project manager agent notes that the forecasting model's performance on the data is not optimal. Following the initialization instructions, it recommends enhancing the model by calculating the optimal parameters.

Project manager (to forecasting agent):

To enhance the accuracy of our predictions, I recommend optimizing the model by calculating the optimal parameters. Let's proceed with tuning the model to achieve better performance. I will provide the necessary instructions for parameter optimization…

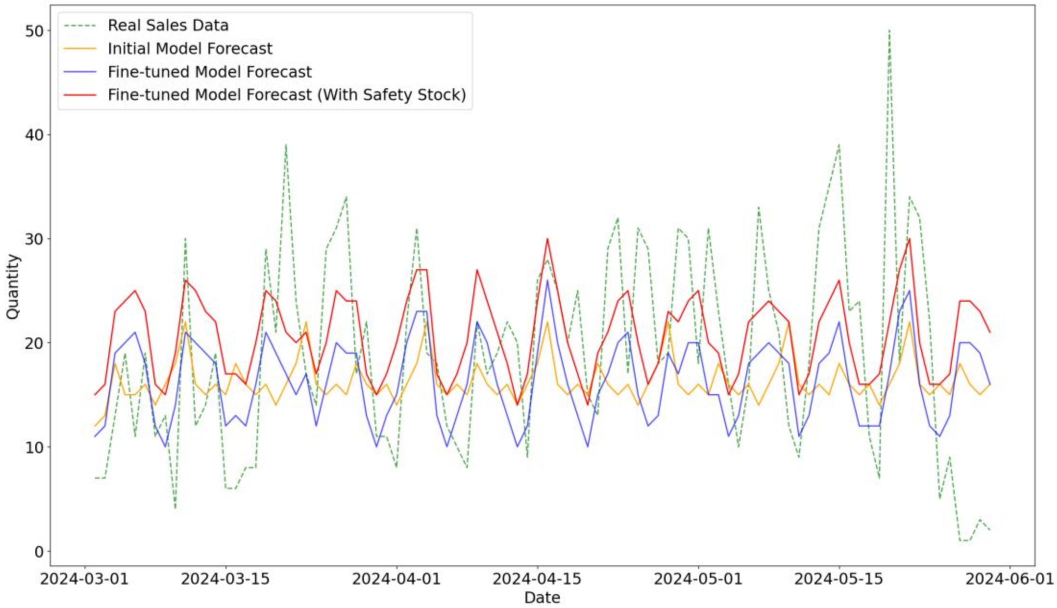
In response to this feedback, the forecasting agent implements a grid search to identify the optimal parameters for the model. The grid search tests various parameter combinations to find the best-performing configuration. After executing the optimized model, the evaluation results demonstrate improvement, although further refinement is necessary to achieve optimal accuracy. Additionally, following the initialization instructions, the forecasting agent is tasked with calculating a safety stock. The purpose of the safety stock is to mitigate the risk of stockouts caused by potential inaccuracies in the model's predictions. In the final step of this iteration, the forecasting agent integrates a safety stock component into the initial predictions, thereby enhancing the reliability of the inventory management process. The forecasting for each day of the next day is saved in a file for the next agent to read and process. Table 2 shows the evaluation of each forecasting model developed in this round, with the Fine-tuned SARIMAX (Including Safety Stock) being the one used for the forecasting.

Table 2: Evaluation of the forecasting models developed in the Conversation 1

Model Name	MAE	MSE	RMSE	MAPE	Safety Stock Included
SARIMAX	8.17	104.84	10.24	92.85%	No
Fine-tuned SARIMAX	7.45	90.1	9.49	93.63%	No
Fine-tuned SARIMAX (Including Safety Stock)	7.56	85.45	9.24	117.63%	Yes

Source: This study.

Additionally, we compared the results of all the models generated by the forecasting agent (as shown in Table 2) against the real data, Figure 2 shows the forecasted sales quantity of each day for the testing quarter compared to the real sales data.



Source: This study.

Figure 2: Comparison of forecasted sales data against real sales data

Conversation 2: Inventory Plan Development

Project manager (to inventory manager):

Please create an inventory plan based on the sales forecast and running costs. The sales forecast data can be found at D:/forecast.xlsx, and the inventory details, including costs, are located at D:/inventory.xlsx. Perform the necessary data analysis and design an inventory plan accordingly.

The objective of inventory management is to design an inventory plan that ensures no stock shortages occur on any day during the next quarter, while simultaneously minimizing shipping and storage costs. In this context, the inventory manager agent is responsible for executing these tasks by inspecting data files containing sales forecasts generated by the forecast agent and current inventory data. The inventory data includes critical information such as shipping costs per item, storage fees per item per month, lead time and shipping time (encompassing both manufacturing and shipping time to the Amazon warehouse), minimum order quantity (MOQ), and the current available inventory. Table 3 shows the data types and values inside the inventory data file.

In the initial round of planning, the inventory manager agent formulates a plan that considers all factors, but it ignores the shipping and storage costs, which makes the inventory plan not optimized for minimum costs. This plan is then reviewed by the project manager agent. Upon review, the project manager agent identifies the lack of cost optimization in the initial plan and recommends employing the Economic Order Quantity (EOQ) model and a linear programming solver to enhance the inventory plan by minimizing costs. Here is a part of the reply of the project manager:

Table 3: Inventory data used of the selected product

COGS	Storage Fees (Per	Shipping Cost (Per	Lead Time	Shipping Time	MOQ (Minimum Order	Inventory

	Item Per Month)	Item)	(Days)	(Days)	Quantity)	
7.96	0.96	0.48	7	15	400	456

Source: This study.

Project manager (to inventory manager):

- EOQ Calculation: Calculate the EOQ while considering MOQ constraints.
- Reorder Point: Determine when to reorder based on lead time and demand.
- Optimization Algorithm: Use a linear programming approach to minimize the total cost, considering storage and shipping costs.

Following the instructions from the project manager agent, the inventory manager agent modifies the initial code to incorporate cost considerations. This involves using the EOQ formula to calculate the reorder quantity while comparing it against MOQ and the difference between forecasted demand and current inventory. Finally, the inventory manager agent sends the final code to the code executor to output the final inventory plan. Table 4 shows the inventory plan developed by the multi-agent framework. This plan is scheduled to the inventory simulation in the Simulation section.

Table 4: The inventory plan developed by the multi-agent framework.

Reorder Date	Quantity
2024-03-02	400
2024-03-21	400
2024-04-09	400
2024-04-29	400

Source: This study.

Simulation

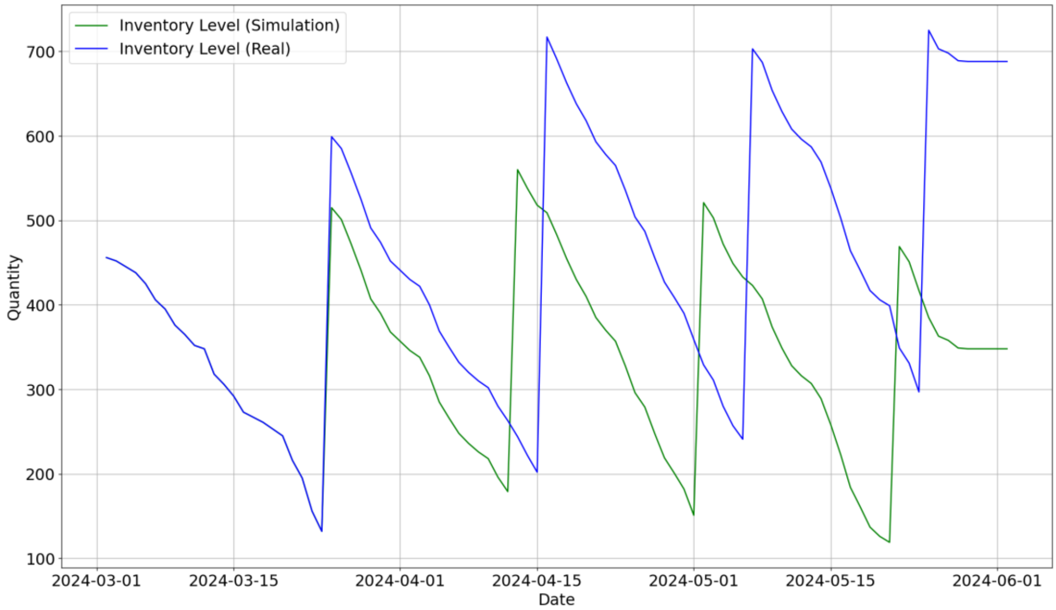
Now that we have the inventory plan for the next quarter, we will put it to use in a sales simulation where each day we will simulate the real sales obtained from the testing data, and inspect the cost and inventory, our main focus is to ensure that no out of stock happens and no over stock, while comparing the inventory plan to the actual inventory plan that the company was using throughout the quarter. Table 5 shows the actual inventory plan used by the company for the quarter between 2024-03-01 to 2024-05-31.

Table 5: The actual inventory plan used by the company.

Reorder Date	Quantity
2024-03-02	484
2024-04-12	524
2024-04-14	472
2024-05-02	460

Source: This study.

The simulation commenced on March 2, 2024, with an initial inventory level of 456 items. Two threads were executed concurrently, each simulating sales throughout the specified period and incorporating an inventory plan scheduling system. This system enabled us to simulate inventory replenishment processes. Specifically, we reviewed the inventory plans detailed in Tables 4 and 5. Whenever the current date matched a reorder date, an order was placed within the simulation. The ordered inventory would arrive after 22 days (comprising both Shipping Time and Lead Time). Throughout the simulation, we closely monitored key performance metrics, including the inventory turnover ratio, stockout rate, shipping cost, holding cost, and total cost. Figure 3 shows the inventory levels of the real-world system against the simulation.



Source: This study.

Figure 3: Real Inventory Level and Simulated Inventory Level

Evaluation

To assess the performance of the proposed multi-agent framework for inventory management, we conducted a comprehensive simulation using real-world sales data and inventory details. The evaluation focuses on several key performance indicators: Inventory Turnover Ratio, Stockout Rate, Shipping Cost, Holding Fees, and Total Cost. These metrics provide a holistic view of the system's efficiency and effectiveness compared to traditional inventory management approaches. The subsequent sections detail the comparative analysis of the simulation data against real data, highlighting the significant improvements achieved through the multi-agent system. Table 6 shows the evaluation results of the real-world system and simulation.

Table 6: Evaluation metrics of the real-world system and simulation.

Data Type	Inventory Turnover Ratio	Stockout Rate	Shipping Fees	Holding Fees	Total Cost
Real Data	23.77	0.00%	$931.19	$1229.01	$2160.21
Simulation Data	33.82	0.00%	$768.00	$931.53	$1699.53

Source: This study.

The Inventory Turnover Ratio is a measure of how frequently inventory is sold and replaced over a specific period and is a critical indicator of inventory management efficiency. In the simulation data, the Inventory Turnover Ratio is significantly higher at 33.82 compared to 23.77 in the real data. This suggests that the multi-agent framework has been able to optimize inventory levels more effectively, leading to faster turnover. A higher turnover ratio indicates that the inventory is being sold and restocked more frequently, reducing the holding period and associated costs. This efficiency can be attributed to the precise demand forecasting and timely inventory replenishment enabled by the multi-agent system.

The Stockout Rate represents the frequency at which inventory is unavailable when needed. Both the real data and simulation data show a Stockout Rate of 0.00%, indicating that the multi-agent framework has maintained the same level of product availability as the real-world system. This is crucial as it demonstrates that the proposed system does not compromise on the availability of products, ensuring that customer demand is consistently met. The ability to maintain a zero stockout rate while optimizing other aspects of inventory management highlights the robustness and reliability of the multi-agent approach.

Shipping Cost is another significant component of inventory management, influencing overall cost efficiency. The simulation data shows a notable reduction in Shipping Cost, with costs reduced to $768.00 compared to $931.19 in the real data. This reduction can be attributed to the optimized decision-making capabilities of the multi-agent system, which likely resulted in more efficient shipping schedules and reduced unnecessary shipments. By better-aligning shipments with actual demand, the system minimizes the frequency of urgent and potentially more expensive shipping methods, thereby lowering costs.

Holding Fees represent the costs associated with storing unsold inventory. In the simulation data, Holding Fees are considerably lower at $931.53 compared to $1229.01 in the real data. This reduction is a direct result of the improved

Inventory Turnover Ratio. By maintaining lower inventory levels through more accurate demand forecasting and optimized stock replenishment strategies, the multi-agent framework minimizes the cost associated with storing excess inventory. Lower holding fees also indicate better space utilization and reduced risk of inventory obsolescence.

The Total Cost combines all expenses related to inventory management, including Shipping Costs and Holding Fees. The simulation data reveals a substantial decrease in Total Cost, with costs reduced to $1699.53 compared to $2160.21 in the real data. This significant cost reduction highlights the overall efficiency and effectiveness of the multi-agent system in managing inventory. The optimized processes result in lower operational costs, which directly contribute to increased profitability. This comprehensive cost-saving demonstrates the potential of multi-agent frameworks to enhance financial performance through smarter inventory management practices.

DISCUSSION

The results of this study underscore the significant potential of leveraging multi-agent systems powered by large language models (LLMs) to address inventory management challenges. The integration of LLMs into a collaborative multi-agent framework has demonstrated noteworthy improvements across various performance metrics, highlighting the transformative capabilities of AI in supply chain operations. However, alongside these positive outcomes, there are also areas that require further attention and optimization.

One of the key findings is the substantial improvement in the Inventory Turnover Ratio observed in the simulation data (33.82) compared to the real data (23.77). This increase suggests that the multi-agent system can manage inventory more dynamically and efficiently, ensuring that products are sold and restocked at an optimal rate. This improvement is crucial for reducing excess inventory and minimizing holding costs. The enhanced turnover ratio indicates that the system effectively aligns inventory levels with demand, leading to more frequent restocking and better inventory management overall. Nonetheless, while this high turnover is beneficial, it also raises the need to ensure that supply chains are resilient enough to handle the increased frequency of inventory cycles without disruption. In this simulation, we did not consider manufacturing and shipping delays, which are important factors to consider in a production environment, future research can explore the ability of multi-agent LLM systems to react to delays in production and shipping, making the system more robust and effective.

The reduction in Shipping Costs and Holding Fees also emphasizes the system's efficiency. By optimizing the reorder dates and quantities, the multi-agent framework reduced Shipping Costs from $931.19 to $768.00 and Holding Fees from $1229.01 to $931.53. Consequently, the Total Cost decreased significantly from $2160.21 to $1699.53, illustrating the economic benefits of employing AI-driven solutions in inventory management. This cost efficiency highlights the potential for significant savings and increased profitability. However, it is important to note that these cost reductions are contingent upon the accuracy and reliability of the forecasting models and the system's ability to adapt to real-time changes in demand and supply chain dynamics.

Maintaining a zero stockout rate in both the real and simulation data is a critical achievement. It demonstrates that the multi-agent system can ensure product availability without overstocking, balancing customer satisfaction with cost efficiency. This reliability is vital for businesses aiming to meet customer demands promptly while avoiding the pitfalls of excess inventory. While maintaining zero stockouts is commendable, it is also essential to consider the system's ability to manage unexpected demand spikes and supply chain disruptions. Future iterations of the system could benefit from incorporating more sophisticated safety stock strategies to mitigate these risks.

The integration of LLM-powered multi-agent systems in inventory management offers a promising avenue for automating and optimizing supply chain operations. These systems can handle complex tasks with minimal human intervention, leading to significant improvements in efficiency, cost reduction, and responsiveness to market demands. However, for broader implementation, it is essential to address potential limitations, such as the reliance on historical data and the need for continuous model updates to maintain accuracy. Also, one of the most challenging parts of this system is forecasting the sales data, the forecasting agent isn’t always able to output working models on the first try. In fact, in this simulation, we had to retry multiple times before the forecasting agent was able to output a satisfactory result. Future research should focus on further enhancing these systems by incorporating real-time data and more sophisticated safety stock strategies. Additionally, exploring the potential of multi-agent frameworks in various industries and under different market conditions can provide deeper insights and broader applications. The ability of LLMs to dynamically allocate, retrieve, and generate new data opens up new possibilities for continuous learning and improvement, making these systems more adaptive and intelligent over time.

CONCLUSION

In this study, we investigated the potential of organizing collaborative teams of AI agents powered by large language models (LLMs) to tackle inventory management problems autonomously. Our multi-agent framework, which includes a project manager agent, a sales forecasting agent, and an inventory manager agent, demonstrated significant enhancements in inventory management efficiency through self and mutual corrections. The results highlight the ability of these agents to dynamically adjust inventory levels and maintain product availability with minimal human intervention.

As evidenced by our simulations, the multi-agent framework significantly improved the Inventory Turnover Ratio and reduced both Shipping Costs and Holding Fees. This led to a substantial decrease in the overall Total Cost, showcasing the economic benefits of AI-driven inventory management systems. The zero stockout rate maintained throughout the simulation further emphasizes the reliability of the system in ensuring continuous product availability, a critical factor for customer satisfaction and operational efficiency.

Our findings suggest that AI-driven multi-agent systems can revolutionize inventory management, offering substantial improvements in efficiency, cost-effectiveness, and responsiveness to market demands. The ability of these systems to dynamically allocate, retrieve, and generate new data on the fly opens up new possibilities for continuous learning and optimization, making them more adaptive and intelligent over time. Future research should focus on further enhancing these systems by incorporating real-time data and more sophisticated safety stock strategies. Additionally, exploring the potential applications of multi-agent frameworks across various industries and under different market conditions can provide deeper insights and broader utility.

In conclusion, the adoption of LLM-powered multi-agent systems marks a significant advancement in inventory management, paving the way for more automated, efficient, and responsive supply chain solutions. As AI technology continues to evolve, its integration into inventory management will likely expand, heralding a new era of human-AI collaboration in business operations. This study contributes to the growing body of evidence supporting the transformative potential of AI in optimizing supply chain management and enhancing overall business performance.

MATERIALS AND METHODS

Agent Design

In this study, we designed and implemented a multi-agent framework to address inventory management challenges by leveraging the capabilities of large language models (LLMs). The agents were created using GPT-4o and the AutoGen framework, an open-source platform for developing agent-based AI models. This framework supports the orchestration, optimization, and automation of workflows, enabling individual AI agents to interact seamlessly and perform various tasks autonomously. The AutoGen framework facilitates the development of applications with multiple conversable agents capable of interacting with each other. These agents are highly customizable and can operate in various modes, incorporating a mix of LLMs, code development and execution, human input, and various tools programmed via function calls. Although function calling was not utilized in this study, the LLM's innate ability to write code was leveraged.

We utilized the "gpt-4o-2024-05-13" model from OpenAI, which has a context window of 128k tokens and training data up to May 2024, to power the agents via OpenAI API ports. The agent team was designed to solve well-defined inventory management problems through self-correction, and handling tasks such as demand forecasting and inventory optimization. The assistant agent was constructed using the AssistantAgent class, while the human user proxy agent was created using the UserProxyAgent class from AutoGen.

Software Versions and Hardware

The multi-agent models were developed using Jupyter Notebook, utilizing Python 3.10.9 and the pyautogen-0.2.27 package. The simulation was developed using python Flask for the backend development and Vuejs for frontend development.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China [Grant Numbers 72071083, 72171089]; the National Social Science Fund of China [Grant Number 23ZDA063]; and the Fundamental Research Funds for the Central Universities [Grant Numbers 2023ZYGXZR021, ZDPY202304, CXTD202403].

REFERENCES

Barron, E. N. (2024). *Game theory: an introduction*. John Wiley & Sons.

Bastarianto, F. F., Hancock, T. O., Choudhury, C. F., & Manley, E. (2023). Agent-based models in urban transportation: review, challenges, and opportunities. *European Transport Research Review, 15*(1), 19. https://doi.org/10.1186/s12544-023-00590-5

Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., ... & Xie, X. (2024). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology, 15*(3), 1-45. https://doi.org/10.1145/3641289

Godwin, H. C., & Onwurah, U. O. (2013). Inventory management: pivotal in effective and efficient organizations. A case study. *Journal of Emerging Trends in Engineering and Applied Sciences*, *4*(1), 115-120.

Gen, M., & Lin, L. (2023). Genetic algorithms and their applications. In Springer handbook of engineering statistics (pp. 635-674). London: Springer London.

Ghazimirsaeid, S. S., Jonban, M. S., Mudiyanselage, M. W., Marzband, M., Martinez, J. L. R., & Abusorrah, A. (2023). Multi-agent-based energy management of multiple grid-connected green buildings. *Journal of Building Engineering, 74*, 106866. https://doi.org/10.1016/j.jobe.2023.106866

Giannoccaro, I., & Pontrandolfo, P. (2002). Inventory management in supply chains: A reinforcement learning approach. *International Journal of Production Economics, 78*(2), 153-161. https://doi.org/10.1016/S0925-5273(00)00156-0

Goli, A., Zare, H. K., Moghaddam, R., & Sadeghieh, A. (2018). A comprehensive model of demand prediction based on hybrid artificial intelligence and metaheuristic algorithms: A case study in dairy industry. *Journal of Industrial and Systems Engineering, 11*(4), 190-203.

Harris, F. W. (1990). How many parts to make at once. *Operations research, 38*(6), 947-950.

Kara, A., & Dogan, I. (2018). Reinforcement learning approaches for specifying ordering policies of perishable inventory systems. *Expert Systems with Applications, 91*, 150-158. https://doi.org/10.1016/j.eswa.2017.08.046

Kocoń, J., Cichecki, I., Kaszyca, O., Kochanek, M., Szydło, D., Baran, J., ... & Kazienko, P. (2023). ChatGPT: Jack of all trades, master of none. *Information Fusion, 99*, 101861. https://doi.org/10.1016/j.inffus.2023.101861

Lei, T., Bai, J., Brahma, S., Ainslie, J., Lee, K., Zhou, Y., ... & Chang, M. W. (2024). Conditional adapters: Parameter-efficient transfer learning with fast inference. *Advances in Neural Information Processing Systems, 36*. https://doi.org/10.48550/arXiv.2304.04947

Lingam, Y. K. (2018). The role of Artificial Intelligence (AI) in making accurate stock decisions in E-commerce industry. *Int. J. Adv. Res. Ideas Innov. Technol, 4*(3), 2281-2286.

Liu, Z., Tang, Y., Luo, X., Zhou, Y., & Zhang, L. F. (2024). No need to lift a finger anymore? Assessing the quality of code generation by ChatGPT. *IEEE Transactions on Software Engineering.* https://doi.org/10.48550/arXiv.2308.04838

Nemtajela, N., & Mbohwa, C. (2016, December). Inventory management models and their effects on uncertain demand. In *2016 IEEE International Conference on Industrial Engineering and Engineering Management* (IEEM) (pp. 1046-1049). IEEE.

Partovi, F. Y., & Anandarajan, M. (2002). Classifying inventory using an artificial neural network approach. *Computers & Industrial Engineering, 41*(4), 389-404.

Ramanathan, R. (2006). ABC inventory classification with multiple-criteria using weighted linear optimization. *Computers & operations research, 33*(3), 695-700.

Rasal, S. (2024). LLM harmony: Multi-agent communication for problem solving. *arXiv* preprint arXiv:2401.01312. https://doi.org/10.48550/arXiv.2401.01312

Schonberger, R. (1982). Japanese manufacturing techniques: Nine hidden lessons in simplicity. *Simon and Schuster, 1*(3), 13-15

Sun, C., Huang, S., & Pompili, D. (2024). LLM-based Multi-Agent Reinforcement Learning: Current and Future Directions. *arXiv* preprint arXiv:2405.11106. https://doi.org/10.48550/arXiv.2405.11106

Waller, M., Johnson, M. E., & Davis, T. (1999). Vendor-managed inventory in the retail supply chain. *Journal of business logistics, 20*(1), 183.

Wild, T. (2017). *Best practice in inventory management*. Routledge.

Wu, T., He, S., Liu, J., Sun, S., Liu, K., Han, Q. L., & Tang, Y. (2023). A brief overview of ChatGPT: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica, 10*(5), 1122-1136.

Zhang, Y., An, B., & Subrahmanian, V. (2024). Computing Optimal Nash Equilibria in Multiplayer Games. *Advances in Neural Information Processing Systems, 36.*