“Cognitive Sandwich” Architecture for LLM Agents in Logistics

Maksim Ilin
Transport and Telecommunication Institute, Latvia
Ilin.M@tsi.lv

Dmitry Pavlyuk
Transport and Telecommunication Institute, Latvia

Abstract
Traditional decision support systems in supply chain management often lack the flexibility to handle unpredictable disruptions, while emerging autonomous agents powered by large language models (LLMs) struggle with hallucination risks and a lack of mathematically guaranteed plans. To address this challenge, this paper proposes the "Cognitive Sandwich" architecture, a hybrid method that surrounds the LLM's semantic reasoning with strict, deterministic constraints. By utilizing the Model Context Protocol, this approach grounds the agent's decision-making in real-world data from a digital twin and validates it through a mathematical solver. Our experimental results, based on testing across 78 port infrastructure degradation scenarios, confirm the architecture's effectiveness. During these tests, we also identified and classified specific behavioral errors in LLM agents, such as "cognitive escape" (attempting to invent unauthorized network routes) and "Stubborn Agent Syndrome" (falling into infinite loops when attempting to iteratively resolve strict mathematical constraints). From a practical perspective, this verified architecture establishes a strong foundation for building reliable and auditable multi-agent systems in logistics, representing a crucial step toward intelligent, adaptable supply chain orchestration under high uncertainty.

Keywords: Multi-agent systems, LLM agents, Operations Research, Cognitive architecture, Supply chains, Neurosymbolic AI, Model Context Protocol.

1. Introduction
Modern global supply chains operate under conditions of unprecedented volatility caused by geopolitical shifts, climate change, and operational disruptions. Maritime logistics, which accounts for over 80% of global trade, is particularly vulnerable to the degradation of port infrastructure. Traditional centralized management systems like Terminal Operating Systems (TOS) and Transport Management Systems (TMS) often lack the flexibility required for dynamic Supply Chain Orchestration - which, in the context of this study, is strictly defined as the real-time allocation of capacities and re-routing of cargo flows between ports under severe infrastructural constraints (Taiwo et al., 2025). In this context, the integration of autonomous agents based on large LLMs is viewed as a promising path toward creating adaptive and self-organizing logistics ecosystems (Keesara, 2025).

LLMs have demonstrated outstanding capabilities in interpreting unstructured data, extracting constraints from textual incident reports, and generating action plans (Din et al., 2025). Our previous research has confirmed that the use of Few-Shot In-Context Learning methods allows models to be adapted to complex logistics tasks without the need for costly retraining (Ilin & Pavlyuk, 2025b). Furthermore, the implementation of the Model Context Protocol (MCP) has enabled the successful encapsulation of statistical tools as services accessible to agents, thereby increasing the reliability of their recommendations (Ilin & Pavlyuk, 2025c).

However, despite these successes, the autonomous application of LLMs in mission-critical optimization tasks faces fundamental barriers. The main one is the “cognitive gap” between the model’s semantic inference and the rigid mathematical limits of the physical world. As recent benchmarks show (Tsui, 2025), modern models have a “self-correction blind spot,” failing to correct mathematically infeasible plans in most cases. Without external deterministic constraints (guardrails), agents are prone to pathological behavior: from endlessly trying incorrect options (“stubborn agent syndrome”) to “cognitive escape”

- attempts to solve a problem by unauthorized expansion of the operational topology.

In this paper, we present the "Cognitive Sandwich" architecture - a neurosymbolic framework that isolates the LLM's cognitive reasoning process from deterministic mathematical verification. Our solution is based on a multi-level memory architecture (Ilin & Pavlyuk, 2025a) and uses an operational solver (OR-solver) as an absolute feasibility filter.

To address these barriers, this study formulates the following research questions (RQs):

- RQ1: How can the phenomena of 'cognitive escape' and 'stubborn agent syndrome' be mitigated in LLM agents operating within strictly constrained, closed logistics topologies?
- RQ2: To what extent does pre-flight deterministic verification (Terminal Infeasibility) improve the computational efficiency and stability of agentic routing?
- RQ3: Can the semantic translation of mathematical solver errors (IIS logs) ensure deterministic, one-shot convergence for infeasible routing plans?

The main contribution of this work is as follows:

- Hybrid architecture that uses the MCP protocol for seamless integration of LLMs with mathematical solvers and digital twins.
- A mechanism for preventive infeasibility analysis (Terminal Infeasibility Check) is introduced, significantly improving the computational efficiency of the system.
- The phenomena of "cognitive escape" and "stubbornness" of agents in closed logistics systems have been empirically documented and classified.
- Empirical documentation of the limitations inherent in direct Irreducible Infeasible Subsystem (IIS) log feedback loops. We demonstrate that without a sophisticated orchestration harness, a naive ReAct loop fails to elicit LLM self-correction, thereby redefining the baseline OR-solver's role from an interactive tutor to a strict, deterministic guardrail.

The remainder of the article is organized as follows: Section 2.1 reviews related work; Section 2.2 describes the architectural methodology in detail; Section 2.3 is devoted to the description of the experimental environment and test design; Section 3 presents the results of batch testing and their discussion; Section 4 summarizes the findings and outlines directions for further research.

2. Procedure

2.1. Literature Review

The integration of large language models into logistics and supply chain management (SCM) systems is one of the most dynamic areas of research at the intersection of artificial intelligence and operations research. The state-of-the-art in this field is characterized by a transition from isolated chatbots to complex agent-based architectures.

2.1.1. Autonomous Agents in Maritime and Urban Logistics

Application of LLMs to solve operational problems in maritime logistics has confirmed the high effectiveness of these models in interpreting natural language for ship mission planning (Din et al., 2025). However, as noted by Ilin & Pavlyuk (2025b), direct fine-tuning of models for narrow logistics domains is often limited by a lack of data. Research shows that the use of Few-Shot In-Context Learning (ICL) methods allows for high-quality logical reasoning and plan structure without significant computational costs. In urban logistics, the "Agentic Digital Twins" paradigm (Xu et al., 2025) justifies the use of agents as orchestrators capable of linking high-level human instructions with executable simulation scenarios.

2.1.2. Control Architectures and Memory Models

The separation of functional layers in logistics systems is reflected in the use of the Model-Control-View (MCV) architecture. According to Keesara (2025), this structure allows the digital twin (Model) to be isolated from the decision-making mechanisms (Control). The effectiveness of these mechanisms directly depends on the agent's memory architecture. Ilin & Pavlyuk (2025a) propose a hierarchical four-layer memory model (L1 Active Context - L4 Semantic Memory) that mimics human cognitive processes. This ensures the reliability and auditability of reasoning, which is critically important for minimizing risks when managing disruptions, especially under conditions of climate volatility (Taiwo et al., 2025).

2.1.3. Neurosymbolic Integration and Tool Utilization

Modern agents go beyond text generation by utilizing tool libraries via standardized protocols. The "Tulip Agent" paradigm (Ocker et al., 2024) and the MCP-Bench benchmark (Wang et al., 2025) emphasize that an agent's key competency is not only planning but also the correct invocation of external services. In a logistics context, this is implemented via the Model Context Protocol (MCP), which allows agents to supplement their reasoning with precise statistical calculations and predictive models (Ilin & Pavlyuk, 2025c), ensuring the grounding of semantics in quantitative data.

2.1.4. Optimization and Compliance with Hard Constraints

The problem of adhering to hard constraints remains a stumbling block for neural network approaches. Reviews in the field of reinforcement

learning (RL) for combinatorial optimization (Mazyavkina et al., 2021) and the research by Zhao et al. (2023) point to the need to use deterministic blocks to guarantee the feasibility of plans on graphs.

The most pressing challenge is described in the work of Ao et al. (2026b), which introduces the concept of "Solver-in-the-Loop." The researchers discovered a "self-correction blind spot": leading API models (GPT-5, Gemini 2.5) are unable to effectively correct mathematical errors, restoring rationality in only 27.6% of cases. The OptiRepair framework (Ao et al., 2026a) offers iterative diagnostics via the solver's IIS logs; however, the problem of low efficiency in "blind" plan revision remains unresolved.

Our proposed "Cognitive Sandwich" architecture fills the identified (Research Gap) by combining hierarchical memory, semantic interpretation of IIS logs, and preventive feasibility checking (Terminal Infeasibility), which allows us to overcome pathological "cognitive stubbornness" and agent hallucinations in closed logistics systems.

2.2. Methodology: The Cognitive Sandwich Architecture

To bridge the fundamental gap between the semantic flexibility of large language models and the rigidity of deterministic optimization algorithms, we propose the "Cognitive Sandwich" architecture. This architecture implements a hybrid neurosymbolic approach, enabling the execution of logistics operations within the "closed world" framework of a digital twin.

2.2.1. System Architecture and Tool Integration

The system orchestration is implemented using the LangGraph framework, which allows the decision-making process to be formalized as a directed cyclic state graph. The architecture is logically divided into three functional layers: a cognitive layer represented by an LLM agent that interprets incidents and generates initial plans; a translation layer responsible for data validation and context management; and a deterministic layer featuring a mathematical solver (Pyomo/HiGHS) acting as a physical guardrail.

A critical element of this architecture is the integration of the digital twin and the mathematical solver via the Model Context Protocol (MCP). As demonstrated in our previous research (Ilin & Pavlyuk, 2025c), MCP allows the agent to seamlessly invoke the OR solver and receive structured data from the sandbox. This provides "grounding" for the LLM's semantic reasoning in strict quantitative metrics, minimizing computational overhead and increasing overall reliability of the system.

2.2.2. Neurosymbolic Routing Workflow

The operational pipeline of the architecture is designed to eliminate LLM hallucinations and infinite loops through a sequence of three deterministic and cognitive mechanisms. Initially, to improve computational efficiency, the system performs a preventive analysis called the Terminal Infeasibility Check. Before activating the LLM agent, an algorithmic check confirms whether the total demand is less than or equal to the sum of the available capacity. If a global capacity shortage is detected, the system immediately exits with the status INFEASIBLE. This approach operates with O (1) constant time complexity and prevents the occurrence of "Stubborn Agent Syndrome," where the model would otherwise fall into infinite recursion attempting to resolve a mathematically impossible conflict.

Following the feasibility check, the system employs a State Injection method to achieve high planning accuracy from the first iteration. Information about infrastructure degradation, obtained through the sandbox's MCP interface, is translated into specific numerical limit values and embedded directly into the agent's system prompt. This allows the LLM to construct a routing plan while being fully informed of the system's actual physical constraints.

In the final verification stage, the deterministic layer acts as a strict guardrail. The plan generated by the agent is evaluated by the OR solver for compliance with the optimization model. If constraint violations are detected, the solver identifies an Irreducible Infeasible Subsystem (IIS). This IIS log is automatically converted into a semantic hint for the repair node (node_repair_artifact). The agent analyzes the specific cause of failure, such as exceeding capacity in a particular port, and iteratively corrects the JSON object. This feedback loop ensures guaranteed convergence to a feasible solution within the specified topological boundaries, effectively preventing "Cognitive Escape" models.

2.3. Experimental Setup

To empirically validate the proposed "Cognitive Sandwich" architecture, a distributed test environment was deployed, consisting of a deterministic scenario generator, a digital twin of port infrastructure, and an end-to-end observability system. The primary goal of the experimental setup was to verify the LLM agent's ability to solve routing problems under strict constraints without risk of data contamination between iterations.

2.3.1. Simulation Environment and Scenario Design

The maritime-port-sandbox microservice, implemented using FastAPI, was used as the simulation environment. Acting as a discrete state tracker rather than a continuous simulator, it served as a strict digital twin of the Northern European port network capacity (target nodes: NLRTM,

BEANR, DEHAM, DEBRV). To ensure strict reproducibility of scientific results, a control script called a specialized Admin API endpoint before each graph run to forcibly set the exact "world state," including blocked ports and capacity limits.

collapse), and Operational Restriction (throughput reduced to 50%). Scenarios were generated using a pseudo-random distribution with a fixed seed value, creating a reference dataset of 78 unique, reproducible situations involving the redirection of volumes ranging from 5,000 to 50,000 TEU.

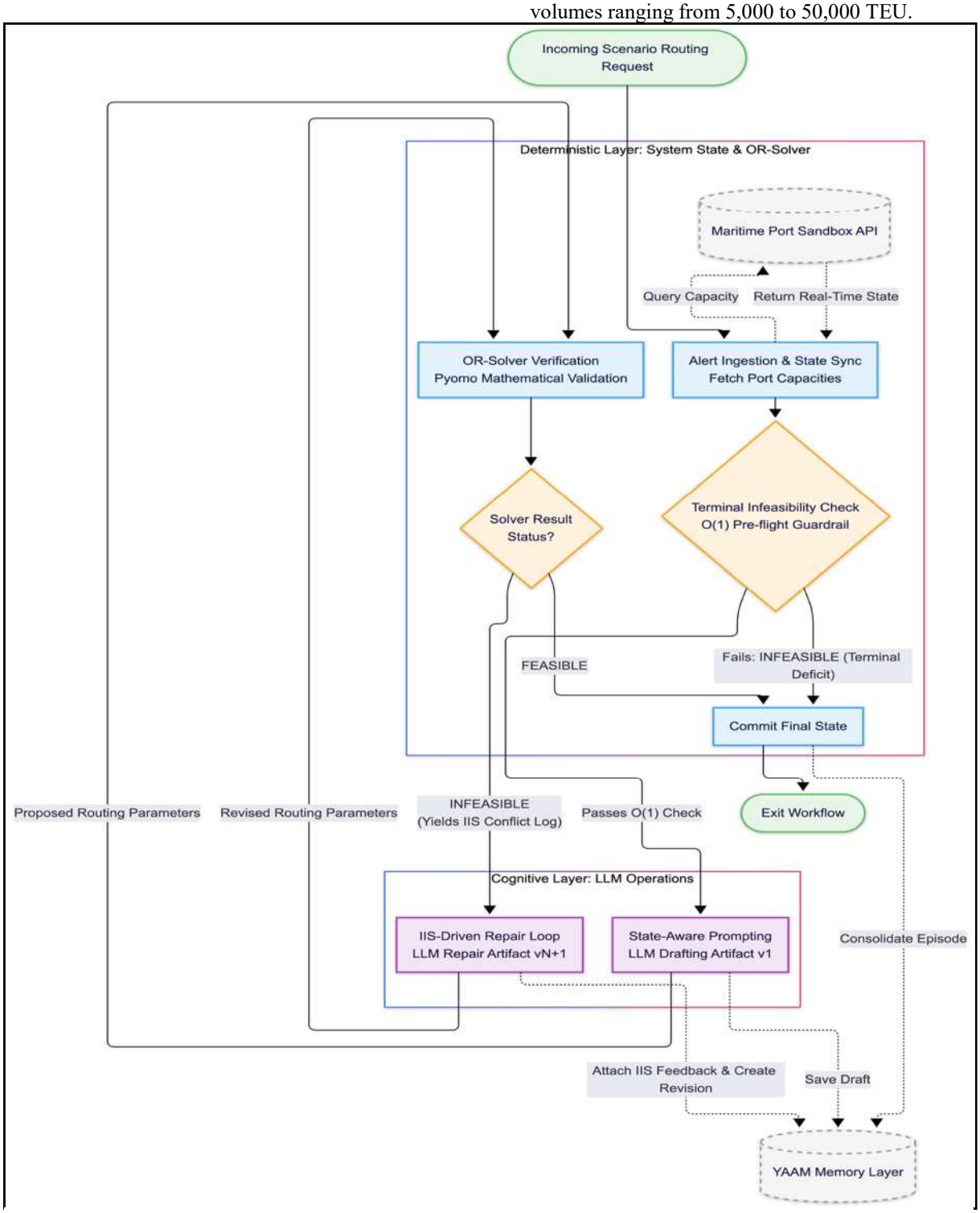

Figure 1: Methodology Flow Diagram

To simulate real-world Port Information Systems (PIS) and DCSA standards, the experimental dataset incorporated a model of gradient degradation of infrastructure throughput. Events were classified into three levels: Total Closure (0% available capacity), Severe Congestion (throughput limited by the solver to 20% to prevent network

2.3.2. Observability and Memory Isolation

A critical requirement for the architecture was to prevent data contamination between experimental runs. Since the system is integrated with the long-term YAAMemory and the short-term graph memory, a two-level namespacing isolation was applied. Each of the 78 runs was assigned a unique

thread ID in the Redis Stack, ensuring physical isolation of checkpoints within the graph. Additionally, data was partitioned at the PostgreSQL and Qdrant vector database levels using a single experimental tag, which isolated the test runs from production data.

Table 1: Experimental Setup Summary Table

Parameter	Configuration	Details
Total Scenarios	78	Number of procedurally generated routing requests in the current dataset
Target Topology	NLRTM, BEANR, DEHAM, DEBRV	The Northern European port nodes allowed for rerouting allocations.
Cargo Volume Range	5,000 – 50,000 TEU	The range of randomly sampled cargo volumes (TEU) that must be rerouted per scenario.
Degradation: Total Closure	0% Capacity Constraint	Simulates complete operational suspension
Degradation: Operational Restriction	50% Capacity Constraint	Simulates partial capacity reduction
Degradation: Severe Congestion	20% Capacity Constraint	Simulates extreme yard utilization (92%), capping throughput to prevent gridlock.

To collect metrics such as execution time, token usage, and recursion iterations, a distributed monitoring system based on OpenTelemetry was deployed. Traceparent header forwarding allowed asynchronous calls to the LLM, the math solver, and the memory subsystem to be combined into a single continuous trace within Arize Phoenix, ensuring full transparency of the decision-making process.

3. Results and Discussion

To empirically evaluate the proposed architecture, we developed a deterministic generator of supply chain failure scenarios and a digital twin of a port network (maritime-port-sandbox). Analysis of 78 controlled experiments revealed several critical phenomena in the interaction between neurosymbolic and mathematical systems.

3.1. Mitigating the "Stubborn Agent Syndrome"

Initial experiments without the use of preventive heuristics revealed a vulnerability we termed the "Stubborn Agent Syndrome." When confronted with unsolvable scenarios (e.g., a capacity shortage of 46,905 TEU with closed alternatives), the LLM agent entered deep recursion, making up to 10 unsuccessful reallocation attempts (average cycle execution time: 6.8–7.2 seconds).

Implementation of the Terminal Infeasibility mechanism made it possible to eliminate this class of errors. Such architecture was able to algorithmically intercept infeasible scenarios before launching the LLM, which reduced the average processing time for unsolvable tasks to ~1.52 seconds (a ~4.7x speedup) and reduced token consumption to zero in these scenarios.

3.2. Eradicating "Cognitive Escape"

The most significant research insight was the identification of the "Cognitive Escape" phenomenon. Analysis of the LLM router logs under conditions of critical capacity shortage revealed attempts by the agent to go beyond the boundaries of the specified topology. Faced with the mathematical impossibility of placing cargo in the permitted ports (NLRTM, BEANR, DEHAM, DEBRV), the agent autonomously “expanded” its world view, proposing to redirect the surplus to the port of Le Havre (FRLEH), which was absent from the digital twin’s configuration.

This behavior illustrates a fundamental limitation of modern LLMs: the prioritization of heuristic expediency (delivering the cargo) over the strict structural rules of a closed system. Without a deterministic OR solver, this destructive hallucination would have led to an integration collapse. The solver successfully blocked these attempts with an INFEASIBLE status, proving the critical necessity of a mathematical protection layer (Guardrail) in autonomous logistics systems.

3.3. The Architectural Limits of Direct Feedback and the Necessity of Deterministic Guardrails

The implementation of the State-Aware Semantic Drafting mechanism yielded a 20.5% zero-shot success rate (16 out of 78 scenarios), where the LLM correctly interpreted the constraints and generated a mathematically valid plan on its first attempt. However, empirical data revealed a critical vulnerability in the baseline system's iterative reasoning capabilities.

Analysis of the failed initial drafts showed a 0% success rate in the IIS-driven repair loop under the current configuration. When the LLM failed the initial allocation, feeding the raw mathematical solver logs (IIS) directly back into the prompt proved insufficient as a harness for self-correction. Instead of iteratively resolving the constraints, the agent inevitably descended into deep recursion, accounting for 43.6% of the runs (34 instances) terminating in an ERROR_RECURSION status.

Table 2: Final Batch Run Results

Metric Category	Metric	Value
Overall Distribution	Total Scenarios Run	78
	Status: FEASIBLE (Zero-shot)	16
	Status: INFEASIBLE	28
	Status: ERROR_RECURSION	34
Performance	Average Execution Time	4.73s
	Average Repair Revisions	1.74
Behavioral Pathologies	Terminal Guardrail Intercepts	0
	Unknown Port Hallucinations	7

Rather than indicating a fundamental incapability of the LLM itself, this finding highlights a critical architectural limitation: a naive ReAct loop is an inadequate framework for complex operations research tasks. While this validates the "self-correction blind spot" observed by Ao et al. (2026a) in basic setups, it suggests the bottleneck lies in the integration methodology. Consequently, within this baseline "Cognitive Sandwich," the OR-solver acts strictly as a hard validation barrier - blocking hallucinations and preventing catastrophic execution - rather than an effective interactive tutor. Overcoming this limitation requires a paradigm shift toward advanced orchestration harnesses (e.g., utilizing LLM-as-a-Judge evaluating simulated states), which informs the trajectory of our ongoing research.

4. Limitations and Conclusion

4.1. Limitations of Current Study

Despite the empirical confirmation of the "Cognitive Sandwich" as an effective validation barrier, this study has several limitations that define the boundaries of its current applicability.

- Static Snapshot Modeling: The current experimental environment utilizes a "closed-world" snapshot approach. While effective for validating routing logic under immediate disruptions, it does not account for the temporal dynamics of maritime logistics, such as vessel arrival sequences, fluctuating port service times, or cumulative yard congestion.
- Naive Feedback Loop Efficiency: As documented in the results, the baseline ReAct-based repair loop proved insufficient for complex mathematical self-correction. The current harness relies on raw IIS-log injection, which lacks the sophisticated orchestration needed to guide LLMs through high-dimensional optimization spaces.
- Topological Scale: The experiments were limited to a four-node port network. While this provided a controlled environment for classifying behavioral pathologies like "Cognitive Escape," further research is required to evaluate the architecture's performance and token efficiency in global-scale networks with hundreds of potential nodes.
- Model Dependency: The findings are based on a specific class of models (Gemini 3.1 Flash Lite). Behavioral patterns and the "self-correction blind spot" may vary across different model families or parameter scales.

4.2. Conclusion and Future Work

This article presents and experimentally verifies the "Cognitive Sandwich" architecture - a neurosymbolic framework for the autonomous orchestration of logistics chains. Our research has shown that integrating the semantic capabilities of large language models with the deterministic rigor of mathematical OR-solvers enables the creation of reliable decision-making systems capable of operating in mission-critical industrial environments.

The main conclusions of the study are as follows:

- The Need for Deterministic Barriers (Addressing RQ1): It has been empirically proven that without external mathematical control, LLM agents are prone to "cognitive escape" - attempts to resolve logistical conflicts by unauthorized expansion of the operational topology. Using a solver as a "guardrail" eliminates the risk of such hallucinations.
- Effectiveness of preventive analysis (Addressing RQ1 and RQ2): The implementation of the Terminal Infeasibility Check mechanism reduced the processing time for mathematically unsolvable scenarios by a factor of 4.7, eliminating the "stubborn agent syndrome" and preventing useless recursive iterations.
- The Architectural Limits of Direct Feedback (Addressing RQ3): Empirical data revealed that feeding raw mathematical IIS logs back to the agent resulted in a 0% self-correction success rate, leading unoptimized drafts to collapse into infinite recursion. This highlights that naive feedback loops are insufficient harnesses for OR tasks. It shifts the baseline solver's role from an interactive tutor to an absolute feasibility filter,

necessitating the development of more advanced orchestration frameworks for future systems.

Results of our work confirm that LLM agents can become reliable decision-support tools in SCM if their reasoning is "grounded" in the physical constraints of the network through formal mathematical models.

Future research will focus on the transition from zero-shot reasoning to long-term learning in context. The next step will be the full integration of the "Cognitive Sandwich" architecture with the four-layer hierarchical memory model YAAM that we previously proposed (Ilin & Pavlyuk, 2025a). We plan to investigate how the accumulation of episodic experience (L3 Memory) and the distillation of semantic knowledge (L4 Memory) will enable the agent to use In-Context Learning to optimize cargo redistribution strategies, reducing the number of calls to the OR solver and increasing the overall intelligence of the system. This work will serve as the foundation for our next study on the cognitive evolution of logistics agents.

References

Ao, R., Simchi-Levi, D., & Wang, X. (2026). Optirepair: Closed-loop diagnosis and repair of supply chain optimization models with LLM agents. DOI: 10.48550/arXiv.2602.19439.

Ao, R., Simchi-Levi, D., & Wang, X. (2026). Solver-in-the-Loop: MDP-Based Benchmarks for Self-Correction and Behavioral Rationality in Operations Research. DOI: 10.48550/arXiv.2601.21008.

Din, M. U., Akram, W., Bakht, A. B., Dong, Y., & Hussain, I. (2025, April). Maritime mission planning for unmanned surface vessel using large language model. In 2025 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR) (pp. 1-6). IEEE. DOI: 10.1109/SIMPAR62925.2025.10979114

Ilin, M., & Pavlyuk, D. (2025a). A Hybrid, Multi-Layered Memory Architecture for Collaborative Reasoning in Multi-Agent Systems. In Proceedings of the 2025 3rd International Conference on Foundation and Large Language Models (FLLM), Vienna, Austria, pp. 1281–1283. DOI: 10.1109/FLLM67465.2025.11391226.

Ilin, M., & Pavlyuk, D. (2025b). Adapting Efficient LLMs for Container Logistics: A Case Study on Few-Shot Prompt Engineering. In: Prentkovskis, O., Yatskiv (Jackiva), I., Karpenko, M., Taujanskaitė, K., Skačkauskas, P., Stosiak, M. (eds) TRANSBALTICA XVI: Transportation Science and Technology. TRANSBALTICA 2025. Lecture Notes in Intelligent Transportation and Infrastructure. Springer, Cham. DOI: 10.1007/978-3-032-18771-0_38.

Ilin, M., & Pavlyuk, D. (2025c). Enhancing Virtual Assistants' Reliability in Logistics with Statistical Tools via Model Context Protocol (MCP). In: Kabashkin, I., Yatskiv, I., Prentkovskis, O. (eds) Reliability and Statistics in Transportation and Communication: Ecosystems for Smart Connectivity and Intelligent Mobility. RelStat 2025. Lecture Notes in Networks and Systems, vol 1822. Springer, Cham. DOI: 10.1007/978-3-032-19245-5_3.

Keesara, V. R. (2025). AI Agents with MCV Architecture in Supply Chain Management: Toward Autonomous and Collaborative Networks. Journal of Information Systems Engineering and Management, 10(63s), 336-350. DOI: 10.52783/jisem.v10i63s.13865.

Mazyavkina, N., Sviridov, S., Ivanov, S., & Burnaev, E. (2021). Reinforcement Learning for Combinatorial Optimization: A Survey. Computers & Operations Research, 134, 105400. DOI: 10.1016/j.cor.2021.105400.

Ocker, F., Tanneberg, D., Eggert, J., & Gienger, M. (2024). Tulip agent: Enabling LLM-based agents to solve tasks using large tool libraries. DOI: 10.48550/arXiv.2407.21778.

Taiwo, A. I., Isi, L. R., Okereke, M., Sofoluwe, O., Olugbemi, G. I. T., & Essien, N. A. (2025). Developing Climate-Adaptive Digital Twin Architectures for Predictive Supply Chain Disruption Management. International Journal of Scientific Research in Science and Technology, 12(3), 931-947. DOI: 10.32628/IJSRST25123101.

Tsui, K. (2025). Self-correction bench: Uncovering and addressing the self-correction blind spot in large language models. DOI: 10.48550/arXiv.2507.02778.

Wang, Z., Chang, Q., Patel, H., Biju, S., Wu, C. E., Liu, Q., ... Siow, E. (2025). MCP-bench: Benchmarking tool - Using LLM agents with complex real-world tasks via MCP servers. DOI: 10.48550/arXiv.2508.20453.

Xu, H., Sun, Y., Tupayachi, J., Omitaomu, O., Zlatanova, S., & Li, X. (2025). Towards the autonomous optimization of urban logistics: Training generative AI with scientific tools via agentic digital twins and model context protocol. DOI: 10.48550/arXiv.2506.13068.

Zhao, Z., Swami, A., & Segarra, S. (2023). Graph-Based Deterministic Policy Gradient for Repetitive Combinatorial Optimization Problems. In International Conference on Learning Representations (ICLR 2023). Available at https://github.com/XzrTGMu/twin-nphard