Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 277 (2026) 1538–1547

Procedia Computer Science

www.elsevier.com/locate/procedia

7th International Conference on Industry of the Future and Smart Manufacturing
(former International Conference on Industry 4.0 and Smart Manufacturing)

LLMOps for End-to-End Automation in Facility Layout Planning: Methodology and Application

Da Ma[a,*], Sebastian Lang[a,b], Sanket Kute[a], Richard Reider[a], Marcel Müller[a]

[a] *Otto-von-Guericke University, Magdeburg, 39106, Germany*
[b] *Fraunhofer Institute for Factory Operation and Automation IFF, Magdeburg, 39106, Germany*

Abstract

Efficient facility layout planning and rapid solution development are crucial for enhancing internal logistics, productivity, and supply chain resilience. Despite extensive research using advanced heuristics or simulation tools, current approaches tend to emphasize single-task performance rather than workflow integration, limiting their industrial applicability. Large Language Model Operations (LLMOps) aims to orchestrate comprehensive end-to-end workflows among models and tools, making it suitable for addressing the interdependent complexities in planning tasks. This paper presents a novel integrated method for facility layout planning that combines the data processing capabilities of Large Language Models (LLMs) with external function calling. The approach creates multiple specialized agents responsible for interpreting technical drawings, implementing optimization algorithms, and summarizing solutions. These agents are orchestrated through an LLMOps platform into a unified workflow. We detail the design and function of each node in the proposed workflow and validate the approach with a conceptual example based on multi-row facility layout problem. This approach illustrates how agents, LLMs, and external tools can be orchestrated to unify fragmented tasks across teams, paving the way for more robust and reusable industrial automation frameworks.

© 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 7th International Conference on Industry of the Future and Smart Manufacturing (former International Conference on Industry 4.0 and Smart Manufacturing)

Keywords: Facility Layout Planning; Logistics in Production; LLMOps; LLM Agent Orchestration; Optimization Algorithms

1. Introduction

Facility Layout Planning (FLP), a fundamental challenge within logistics and production systems, refers to the spatial arrangement of all necessary resources within a production or service facility, such as machines, departments, work centers, or warehouses [1]. FLP seeks to improve material flow, maximize space utilization, and enhance production efficiency through the strategic placement of operations and equipment [2]. Tompkins and Reed [3] highlight that effective facility design requires the integrated consideration of material handling systems and spatial plant layout, since their interaction has a significant impact on total material handling costs. FLP problems can be broadly classified

* Corresponding author. Tel.: +49 391 67-57306
E-mail address: da.ma@ovgu.de

1877-0509 © 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 7th International Conference on Industry of the Future and Smart Manufacturing (former International Conference on Industry 4.0 and Smart Manufacturing)

into two categories: greenfield design and brownfield re-layout. Greenfield design involves the creation of layouts for entirely new facilities, while brownfield scenarios deal with the adaptation and optimization of existing layouts [4, 5]. In practice, efficient layout generation and rapid re-planning are crucial not only for improving internal logistics and productivity but also for enhancing supply chain resilience in dynamic commercial environments.

FLP is typically addressed through two main approaches: one focuses on constructing simulation-ready models based on engineering drawings and domain-specific constraints, while the other concentrates on computational methods for solving the optimization problems. In practice, the initial layout is often derived from requirements analysis of CAD drawings or existing planning guidelines [6], which are then manually transformed into models that can be evaluated using simulation software [7]. These candidate layouts are then used as input to the optimization's fitness function, where discrete-event simulation tools assess their performance under different operational conditions using data such as scheduling rules, transport routes, and resource allocation [7, 5]. On the optimization side, FLP is widely recognized as an NP-hard problem [8, 9], with a solution space that expands rapidly as the number of layout elements increases. To solve this, researchers have developed numerous heuristic and metaheuristic algorithms aimed at finding near-optimal configurations under complex constraints [10, 8, 9, 11].

Despite progress in computational methods, the upstream process of requirements analysis and modeling remain largely manual and disconnected from the optimization phase. This gap presents an obstacle to the practical deployment of FLP solutions, especially in dynamic production environments that require frequent reconfiguration and rapid adjustments. Academic efforts often focus on algorithmic refinements for specific subtasks under certain ideal scenarios [12, 13], while overlooking the integration of end-to-end pipelines and the development of general, deployable solutions for real-world industrial settings. As Burggraf et al. [14] point out, there are currently no algorithms capable of fully automating the evaluation and selection of facility layouts based on qualitative criteria without human involvement. As a result, layout generation remains largely dependent on manual judgment, which limits the accessibility and scalability of academic solutions in industrial applications. This limitation stems not only from the inherent difficulty of quantifying qualitative aspects, such as spatial logic or planner preferences, but also from the methodological challenges in building user-friendly decision support systems.

The advent of Generative Artificial Intelligence (GenAI) exemplified by Large Language Models (LLMs) has introduced new possibilities for addressing these challenges. LLMs demonstrate powerful problem-solving abilities, for instance in optimization tasks requiring complex reasoning and interaction with external computational tools [15]. Furthermore, their enhanced scalability, external tool invocation capabilities, and autonomous knowledge base construction enable LLMs to function as intelligent agents or even virtual employees.

In response to the complexity and multi-stage nature of FLP tasks, adopting an LLM-based operational approach provides a promising avenue. Building on this idea, we propose a workflow that leverages Large Language Model Operations (LLMOps) to unify data parsing, optimization, and result analysis into a cohesive pipeline, facilitating rapid and repeatable FLP with reduced implementation effort. The core research question of this study is: how can an integrated workflow powered by LLMs automate and connect data preprocessing, layout generation, and optimization to support facility layout reconfiguration?

The subsequent sections are organized as follows: section 2 reviews key background technologies and relevant prior work; section 3 details the proposed methodology for developing an integrated workflow; section 4 provides a case study demonstrating the implementation of the LLMOps framework. Finally, section 5 concludes the paper, summarizing key contributions and outlining directions for future research.

2. Technical Background and Related Work

This section outlines the technical foundations and related work. Given the multi-objective nature of the tasks, we divide the discussion into three parts. We begin with optimization algorithms for material flow, which address the core computational challenges of FLP. We then introduce the fundamental concepts of generative AI techniques and their applications in layout planning. Finally, we present the LLMOps framework, which enables multi-agent orchestration and integration with external tools.

2.1. Optimization Algorithms for Material Flow

Evaluating the performance of facility layouts typically involves one or more criteria, among which approximately 70% are directly related to material flow efficiency [2]. Reducing material handling costs has been widely acknowledged as a primary strategy for maximizing overall factory productivity, as it accounts for 20% to 50% of total operating expenses [10]. Thus, the central challenge of FLP is to find optimal or near-optimal solutions that balance multiple objectives that may conflict with one another.

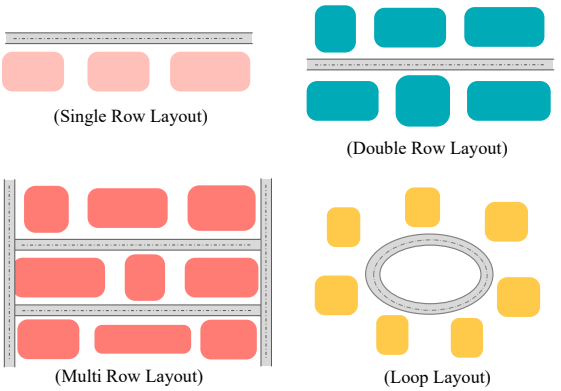

Fig. 1: Categories of facility layouts according to material handling (adapted from [1])

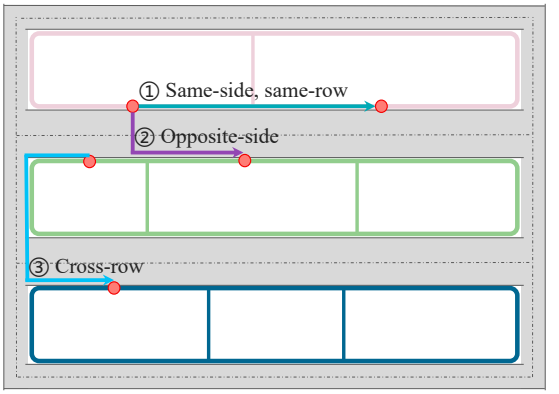

Fig. 2: Calculation of material handling distances considering the location of doors (red dots indicate door positions)

Depending on the type of material handling system, facility layout configurations are commonly categorized into single row, double-row, and multi-row layouts, as illustrated in Figure 1. Single row layouts are often favored for their simplicity, while multi-row layouts allow for more compact designs and higher spatial efficiency, especially when dealing with multiple handling paths. In this study, we will focus on the multi-row layout problem and incorporate the location of doors when planning paths and calculating handling distances.

In a typical multi-row layout scenario, we consider m production modules and n available rooms, where each module must be assigned to a room. To consider a specific layout arrangement, we define a revised material handling cost function. Let $T = t_{ij}$ be the transport demand matrix, where t_{ij} denotes the material transportation demand per day between modules i and j; and let $D = d_{rs}$ represent the distance matrix between room locations, where d_{rs} denotes the transportation distance between rooms r and s. Given a permutation function A encoding the assignment of modules to rooms, the cost becomes dependent on the transformed distances $d_{A(i)A(j)}$, yielding the following optimization goal:

$$\textit{minimize} \quad C = \sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} \cdot d_{A(i)A(j)} \tag{1}$$

To improve realism in distance calculation, we define the distance matrix $D = d_{rs}$ based on door-to-door paths rather than room centroids. As shown in Figure 2, doors are placed at the midpoint of one side, and internal room traversal is prohibited. We distinguish three cases of paths based on the spatial relationship between the doors: (1) Same-side, same-row: use direct horizontal distance; (2) Opposite-side, same-row: use Manhattan distance with vertical detour; (3) Cross-row: compute the shortest U-shaped path via corridors between rows.

Each room r_i has a length L_i and each production module m_j has a length ℓ_j. To ensure spatial feasibility, the constraint is that no module may occupy more than 80% of the assigned room's length. Formally, for a given assignment $A(j) = i$, yielding the following constraint:

$$\ell_j \leq 0.8 \cdot L_{A(j)} \tag{2}$$

Several optimization algorithms have been applied to meet the challenges of multi-row layout design. Heuristic-based methods such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA) are commonly employed to efficiently navigate complex, multi-dimensional search spaces. For example, GA operates

on populations of candidate solutions using iterative selection and recombination to approximate global optima [8]. PSO, as demonstrated by Hu et al. [16], exploits interactions among particles to converge rapidly toward high-quality solutions. Jiang et al. [9] applied a multi-objective SA to simultaneously minimize material handling costs. However, despite numerous available algorithms, the tasks such as defining fitness function, selecting objectives, and tuning hyper-parameters still need manual effort from domain experts. Therefore, there is a growing need to reduce the manual burden by introducing intelligent assistance. The next section introduces relevant applications of GenAI in this context.

2.2. Generative AI in Facility Layout Planning

GenAI is a class of AI techniques that generate new content, such as text, images, or structured data [17]. Initial attempts have been made to apply GenAI to layout planning, with a focus on spatial reasoning in industrial contexts. Disselkamp et al. [18] describe a wide range of early-stage applications of ChatGPT in different phases of factory planning, such as identifying layout functions, proposing ideal logistics configurations, and visualizing simplified layouts using ASCII or code-based interfaces. Several systems also focus on using GenAI for spatial layout generation. For instance, GenFloor, introduced by Keshavarzi et al. [19], provides an interactive design assistance system that can generate and evaluate spatial layouts based on geometric and performance constraints. Similarly, Hu et al. [20] explore prompt engineering and structured semantic mapping to guide LLMs in generating valid shopfloor layouts.

While the studies mentioned above do show GenAI's capability to solve constraints and spatial logic, these efforts are generally limited to conversational capabilities and rarely integrate external knowledge or tools. For example, LLMs do not natively support spatial reasoning or visual layout generation. As shown by Disselkamp et al. [18], the text-to-image models are unable to generate a valid layout. It suggests the need to decouple tasks, assigning LLMs to reasoning and interpretation, while assigning tasks such as numerical computation and visualization to external functions. One notable case is the work of Kmiecik [21], which incorporates LLMs and optimization algorithms for warehouse material flow optimization, aiming to minimize handling distances. This example illustrates the importance of developing multiple agents and a unified toolchain for FLP tasks, a topic that will be further introduced in the next subsection within the context of LLMOps orchestration.

2.3. LLMOps for Multi-Agent Orchestration

The interconnected nature of the FLP problems requires the coordinated operation of multiple components. In this regard, LLMOps stands out as a hybrid discipline of software engineering and artificial intelligence, dedicated to the deployment, orchestration, and maintenance of LLM-based agents across the entire lifecycle [22, 23]. Like Machine Learning Operations (MLOps) and Development Operations (DevOps), LLMOps incorporates data management, model development, deployment, and monitoring, and further extends to prompt engineering and external tool integration.

Two emerging trends illustrate the ongoing evolution of LLMOps. On one hand, visual workflow builders lower the entry barrier by enabling process design through no-code or low-code interfaces for non-programming users. On the other hand, the Model Context Protocol (MCP), released by Anthropic in November 2024, establishes an open-standard specification independent of model vendors. With unified interfaces for context sharing, function calling, and data access, MCP supports overcoming fragmentation in industrial environments.

Although the concept of LLMOps has attracted growing attention since the release of ChatGPT-3.5, the field still lacks a universally accepted conceptual framework or formal standards [22]. Existing work emphasizes the concepts and frameworks of LLMOps, with limited contribution to practical use cases. In this work, we aim to address this gap by applying LLMOps to unify the FLP tasks in concrete industrial scenarios. Furthermore, by integrating external tools aligned with the MCP standard, our approach contributes to the realization of industrial deployment.

3. Development of the Proposed Workflow

This section outlines the proposed workflow for applying LLMOps to solve the FLP problem in a modular and systematic way. It begins with an overview of the entire process, followed by detailed explanations of node definitions and prompt design, and finally discusses the integration of external tools.

3.1. Overview of the Proposed Workflow

The overall methodological framework originates from the six phases of factory planning defined in the VDI 5200 guideline [24], which includes target planning, preliminary planning, rough planning, detailed planning, implementation planning, and realization. These phases serve not only as domain context but also guide the decomposition and structuring of system tasks. Agent definitions emphasize the extension of LLMs reasoning capabilities through structured interaction with external tools, enabling the system to complete domain-specific tasks. Specifically, tools are introduced to handle input parsing, optimization execution, and layout plotting, because these tasks exceed the native capabilities of language models.

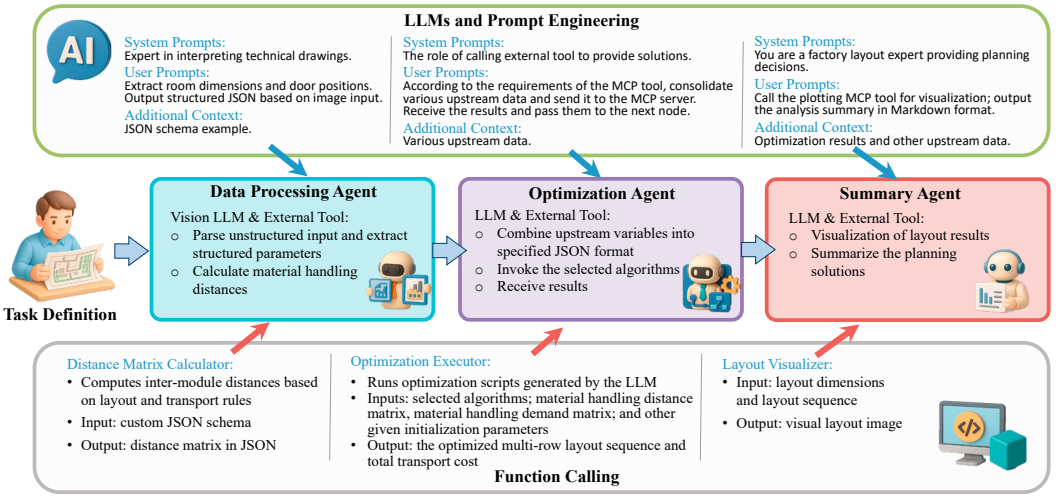

Fig. 3: Overview of the proposed LLM-based multi-agent system

We propose a conceptual workflow illustrated in Figure 3. The workflow consists of three LLM-powered agents, each responsible for a phase of the planning pipeline and coordinated through external function calling mechanisms.

1. **Data Processing Agent:** Parses unstructured layout input, extracts spatial parameters, and computes inter-module material handling distances.
2. **Optimization Agent:** Executes the optimization program to minimize transport cost while fulfilling spatial constraints, yielding optimized module sequences and transport cost values.
3. **Summary Agent:** Visualizes the optimized layout by a plotting library, and evaluates planning solution, offering interpretive feedback based on transport logic and spatial reasoning.

Interviews with our industrial partners revealed two key stakeholder needs: clients seek rapid and diverse layout solutions, while designers in multidisciplinary teams require coordinated workflows supported by seamless information exchange. To meet these demands, our workflow employs LLM-based agent orchestration guided by external function coordination, aligning with the principle that effective data and task orchestration is essential in multi-stage machine learning pipelines [25]. The following section provides detailed instructions on node definition and tool integration.

3.2. Node Definition and Prompt Design

In LLMOps-based workflows, functional nodes serve as the fundamental units of execution. Each node encapsulates a specific operation, and together they define the overall logic of a pipeline. This modular design enables dynamic task orchestration, especially when integrating LLM-based agents into multi-stage processes [26]. Table 1 describes node components used in our workflow, including inputs, outputs, and core functionalities.

The design of the prompts for the three agents is explained in detail in Figure 3, which consists of three coordinated layers: system prompt, user prompt, and additional context, this structure provides control over both task framing and output consistency. As highlighted in recent studies, structured prompts are not merely task inputs but serve as interfaces that encode the reasoning scope and behavior expectations of each LLM agent, particularly in

Table 1: Workflow Node Definitions and Functional Roles

Node	Input	Output	Functionality
Start	Predefined parameters, images, checkboxes	Global variables	Initializes task context and global inputs
LLM	Text, image, code, JSON schema	JSON output	Executes logic defined by LLM prompts
Agent	User queries, tool descriptions, upstream variables	Structured results	Selects and invokes external tools
Http request	API request payload	Remote result	Interfaces with external tools or services
End	Upstream variables	Final output	Delivers workflow results to the user

complex multi-agent planning environments [27, 28]. We categorize prompts into three types: the system prompt defines the agent's domain role, guiding its overall reasoning style; the user prompt specifies the target task and constraints; the additional context complements the prompt with JSON schemas or few-shot examples, helping to reduce hallucinations and improves consistency of output format.

3.3. Function Calling for LLM-Agents

Automated layout planning is achieved by integrating function calling into the LLM-agent workflow, allowing agents to systematically leverage external tools for their core tasks. We designed three external tools for each agent as shown in Figure 3. Each of these external tools extends the core abilities of the corresponding agent by handling tasks that are beyond the native capabilities of LLMs. The distance matrix calculator supplements the data processing agent by providing precise inter-module distance calculations, which the LLM cannot compute directly from image recognition alone. The optimization executor invokes selected optimization algorithms to obtain results, thereby overcoming the computational limitations of LLMs. Finally, the layout visualizer supports the summary agent by converting optimized module arrangements into visual outputs, enabling clear interpretation of layout solutions.

Each external tool is designed to be invoked by the LLM through MCP-aligned schemas, ensuring structured interaction and extending capabilities. In this way, agents can focus on their reasoning tasks and get support from external tools whenever complex computations or visualizations are required.

4. Case Study

To validate the effectiveness of the LLMOps workflow introduced in the previous section, this section presents a case study focused on a brownfield scenario involving multi-row facility layout planning. We illustrate the entire workflow from initial task definition, through optimization to the final conclusion, leveraging the LLMOps platform Dify [29] and customized external tools implemented under MCP standard, which provides standardized Streamable HTTP transport protocol. The source code and documentation are available at `https://github.com/0xMA1024/e2e-flp-llmops`.

4.1. Use Case Definition and Implementation

The brownfield FLP task aims to arrange the new facilities within the existing site. The original layout comprises 3 rows of 11 rooms, each with defined lengths and door positions. User input includes the modules lengths (see Table 2) and the layout image, both adapted from our industrial partner, and a transport demand matrix (see Table 3) generated by random assignment of material flow levels 0 to 5 in the ratio 10:5:2:2:1:1 (0 = no flow, 5 = highest). The planning task is minimizing the overall transport cost by assigning each of the 11 production modules (denoted by $M1$ to $M11$) to a room, under the restriction that the length of any module must not exceed 80% of the corresponding room.

Table 2: Length and name of production modules

Module	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
Length (m)	18	21	14	16	5	10	35	8	20	15	22

For the implementation phase, we employ the Dify platform, selected due to its comprehensive support for diverse node types, compatibility with various LLM models, and user-friendly visual workflow builder. Specifically, the MCP server is hosted on a local port, where Dify discovers the available MCP tools and makes them accessible for invocation by agent nodes. Dify's modularity and visualization features streamline the workflow design, debugging, and improvement processes, effectively enhancing operational efficiency.

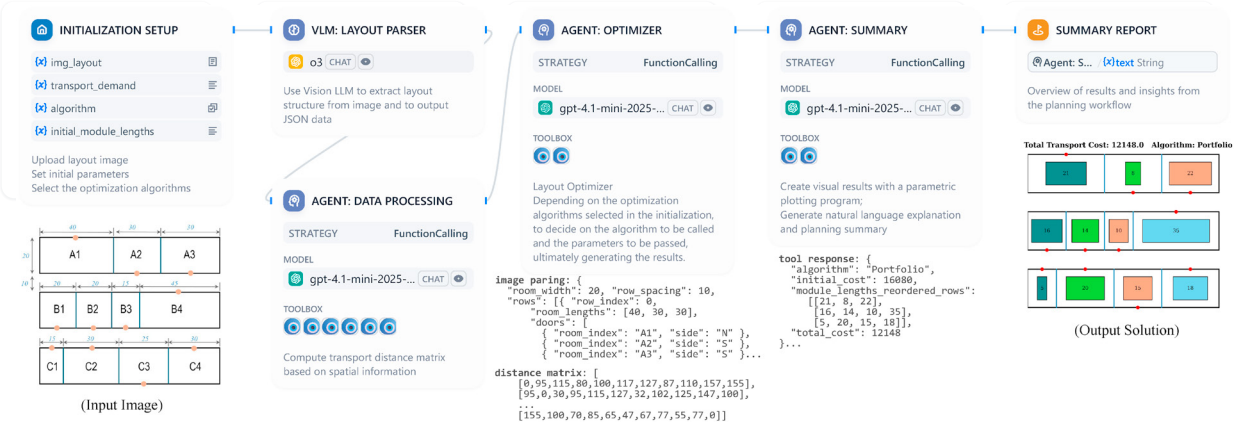

Fig. 4: Dify workflow illustration with input image, intermediate variable examples, and generated layout

For optimization, we employ the open-source python library Nevergrad [30], a derivative-free black-box optimizer known for its versatility and ease of use. It supports rapid deployment of diverse algorithms without manual hyperparameter tuning, making it ideal for real-world optimization tasks lacking explicit gradients. A custom fitness function is defined to minimize total transport cost $C = \sum_{i=1}^{m} \sum_{j=1}^{n} t_{ij} \cdot d_{A(i)A(j)}$. A penalty term is added for any length violation. We configure several algorithms from Nevergrad, using sorted continuous arrays to encode discrete facility arrangement. All algorithms are run for a fixed number of 50,000 evaluation iterations.

Table 3: Transport demand and distance matrices between production modules

(a) Transport demand matrix between production modules (unit/day)

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
M1	0	1	2	0	4	2	1	1	1	4	5
M2	1	0	0	2	3	3	0	0	3	0	1
M3	2	0	0	2	1	2	0	2	3	1	0
M4	0	2	2	0	4	0	2	1	1	3	1
M5	4	3	1	4	0	2	1	2	0	0	0
M6	2	3	2	0	2	0	1	1	4	0	0
M7	1	0	0	2	1	1	0	3	2	1	0
M8	1	0	2	1	2	1	3	0	0	1	3
M9	1	3	3	1	0	4	2	0	0	3	1
M10	4	0	1	3	0	0	1	1	3	0	1
M11	5	1	0	1	0	0	0	3	1	1	0

(b) Distance matrix of production modules (m)

	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11
M1	0	95	115	80	100	117	127	87	110	157	155
M2	95	0	30	95	115	127	32	102	125	147	100
M3	115	30	0	125	115	97	17	132	125	117	70
M4	80	95	125	0	20	37	107	12	30	97	85
M5	100	115	115	20	0	17	112	32	10	117	65
M6	117	127	97	37	17	0	95	50	27	125	47
M7	127	32	17	107	112	95	0	115	122	115	67
M8	87	102	132	12	32	50	115	0	22	85	77
M9	110	125	125	30	10	27	122	22	0	107	55
M10	157	147	117	97	117	125	115	85	107	0	77
M11	155	100	70	85	65	47	67	77	55	77	0

Figure 4 shows the full workflow containing all nodes. The process starts with reading four user inputs: a brownfield layout image, a transport demand matrix between production modules, the initial sequence of production modules, and the selected optimization algorithms. The image is parsed by a vision-LLM to extract spatial data, and the agent node for data processing invokes the MCP tool to compute the distance matrix. This tool computes the coordinates of all room endpoints and entry points, derives door-to-door distances, and returns a JSON-formatted matrix. Subsequently, the optimizer agent sends the selected algorithm and upstream data to the MCP server, which computes transportation costs using the input data, performs optimization iterations, and returns the results. Finally, the summary agent analyzes the optimization results combined with upstream data, explaining the optimal solution from the perspective of reducing transportation costs. The summary agent can call the plotting tool to create schematic diagram of the layout, ultimately generating a richly illustrated report in Markdown format. The vision-LLM employs `o3-2025-04-16` model, as it provides higher success rates in image processing. For the remaining nodes, we adopt the more compact

`gpt-4.1-mini-2025-04-14` model due to its MCP integration and lower cost. The next subsection presents the application and results of the workflow after deployment.

4.2. Results and Discussion

Once the Dify workflow is published and deployed, users can interact with it through a clean, simplified interface, as shown in Figure 5. After the necessary information is uploaded, the workflow starts executing on the backend and returns layout optimization results directly on the right-hand side, including visualizations, performance metrics, and exp[illegible]

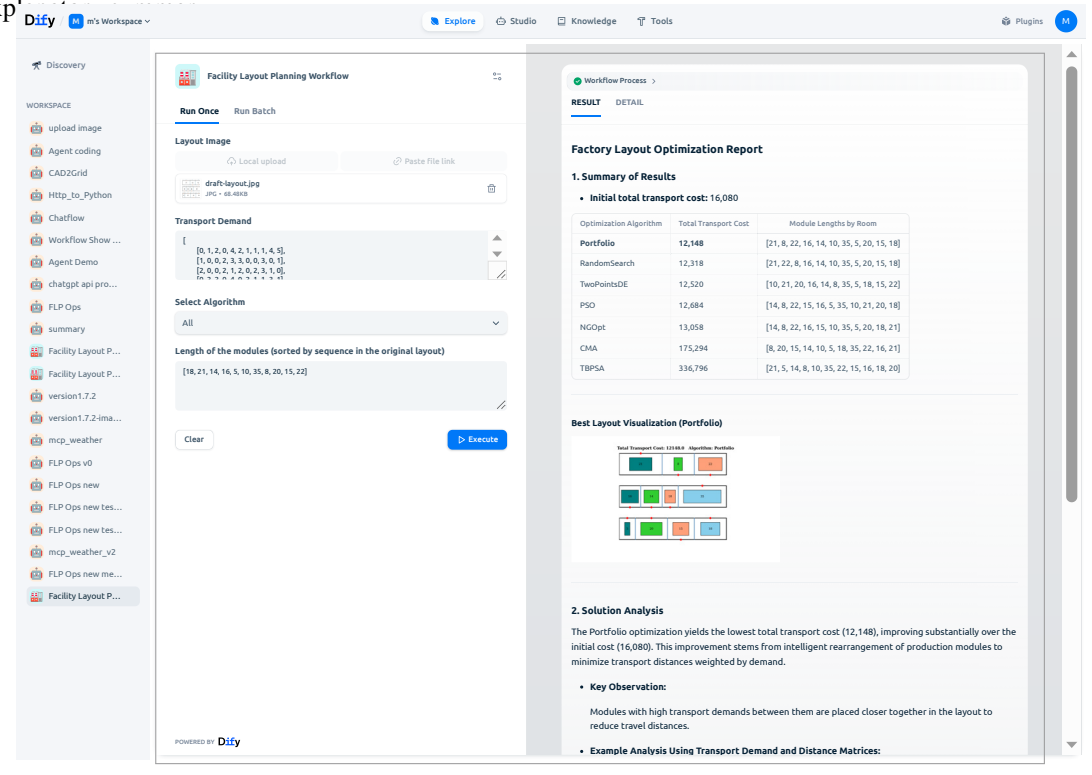

Fig. 5: Published Dify workflow: user interface with input parameters and generated solution

To provide a more comprehensive solution, we have executed all available optimization algorithms for comparative analysis. The summary agent successfully identified and articulated key layout features by reasoning over both numerical outputs and spatial visualizations. Among all algorithms, it correctly identified the optimizer with the lowest transport cost at 12,148. In addition, it highlighted that module pairs with high transport demand (e.g., $M4$: 16 m, $M5$: 5 m) were placed closer in the optimized layout, thereby reducing total transport cost.

The results address our core research question by confirming that an LLM-powered workflow can connect fragmented and separate tasks within a single data pipeline, and custom LLM agents enable lossless information exchange across all workflow stages. Each modular node is designed to handle its own unique function, which not only improves clarity but also makes these components easier to reuse or extend in future projects. This approach benefits organizations by supporting consistent data exchange, simplifying cross-team collaboration, and promoting the development of reusable solutions, thereby enhancing the overall efficiency and competitiveness.

Several limitations must be acknowledged. The present case does not cover all scenarios of FLP. More domain-specific tools are required to extend the toolchain to broader scenarios, such as greenfield planning and discrete event simulation. Moreover, MCP indeed standardizes tool integration and JSON schema definition, yet data exchange still depends on the LLM's instruction-following capability. Under large payload or deeply nested JSON, the model may pass incomplete data to downstream tools. Equally important, unlike the massive open-source libraries available for human developers, MCP tools for LLMs still require manual construction, a plug-and-play MCP ecosystem is still in

its early stages. Unlocking the benefits of LLMOps calls for continued progress in both domain-specific technologies and ecosystem maturity.

5. Conclusion and Outlook

This study presents a proof-of-concept implementation of an automated workflow for multi-row facility layout planning, orchestrated through an LLMOps framework. Rather than focusing on improving the performance of specific optimization algorithms, our main contribution is to verify how LLM-orchestration can unify fragmented tasks, from data interpretation and optimization to solution summarization, within a single cohesive pipeline. By equipping the LLMs with custom external tools for matrix computation, optimization execution, and plotting, the workflow enables standardized, lossless data exchange and validates the feasibility of LLM-based agent in industrial scenarios.

To extend our workflow beyond brownfield planning scenarios and broaden its applicability, future research should address three key directions: (1) Integrating commercial material flow simulation software into external tool ecosystems; (2) Fine-tuning vision LLMs to interpret more input formats such as CAD files; (3) Adopting MCP to enhance the standardization of resource integration, thereby facilitating more mature industrial deployment; and (4) Advancing agent autonomy by exploring self-evolving strategies [31], thereby enabling agents to autonomously solve problems through self-prompting.

From a broader perspective, small and medium-sized enterprises face barriers to accessing GenAI technologies, due to the lack of accessible and user-friendly tools, which has hindered the widespread adoption and scalability of LLM-based solutions [32]. LLMOps-driven end-to-end automation and low-code development provide a promising pathway for democratization of technology and technological sovereignty in the industrial sector.

Acknowledgements and Funding

This research is funded by the German Federal Ministry of Research, Technology and Space (BMFTR) under the project GenAI4FFD. We sincerely appreciate the support provided by the German Aerospace Center (DLR), the Fraunhofer IFF, and our industrial partners for their cooperation.

References

[1] A. Drira, H. Pierreval, S. Hajri-Gabouj, Facility layout problems: A survey, Annual Reviews in Control 31 (2007) 255–267. URL: https://linkinghub.elsevier.com/retrieve/pii/S1367578807000417. doi:10.1016/j.arcontrol.2007.04.001.

[2] S. A. Ali Naqvi, F. , Muhammad, A. , Muhammad, Z. , Muhammad, , M. M. Shehzad, Productivity improvement of a manufacturing facility using systematic layout planning, Cogent Engineering 3 (2016) 1207296. URL: https://doi.org/10.1080/23311916.2016.1207296. doi:10.1080/23311916.2016.1207296, publisher: Cogent OA _eprint: https://doi.org/10.1080/23311916.2016.1207296.

[3] J. A. TOMPKINS, , R. REED Jr, An applied model for the facilities design problem, International Journal of Production Research 14 (1976) 583–595. URL: https://doi.org/10.1080/00207547608956377. doi:10.1080/00207547608956377, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207547608956377.

[4] P. Pérez-Gosende, M. , Josefa, , M. Díaz-Madroñero, Facility layout planning. An extended literature review, International Journal of Production Research 59 (2021) 3777–3816. URL: https://doi.org/10.1080/00207543.2021.1897176. doi:10.1080/00207543.2021.1897176, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00207543.2021.1897176.

[5] M. Klar, P. Schworm, X. Wu, P. Simon, M. Glatt, B. Ravani, J. C. Aurich, Transferable multi-objective factory layout planning using simulation-based deep reinforcement learning, Journal of Manufacturing Systems 74 (2024) 487–511. URL: https://linkinghub.elsevier.com/retrieve/pii/S0278612524000736. doi:10.1016/j.jmsy.2024.04.007.

[6] V. Naranje, P. V. Reddy, B. K. Sharma, Optimization of Factory Layout Design Using Simulation Tool, in: 2019 IEEE 6th International Conference on Industrial Engineering and Applications (ICIEA), 2019, pp. 193–197. URL: https://ieeexplore.ieee.org/abstract/document/8715162. doi:10.1109/IEA.2019.8715162.

[7] S.-H. Nam, S.-H. Oh, H.-C. Yoon, Y.-I. Cho, K.-Y. Cho, D.-H. Kwak, J. H. Woo, Development of Des Application for Factory Material Flow Simulation With Simpy, in: 2022 Winter Simulation Conference (WSC), 2022, pp. 1545–1556. URL: https://ieeexplore.ieee.org/document/10015508. doi:10.1109/WSC57314.2022.10015508, iSSN: 1558-4305.

[8] S. P. Singh, R. R. K. Sharma, A review of different approaches to the facility layout problems, The International Journal of Advanced Manufacturing Technology 30 (2006) 425–433. URL: https://doi.org/10.1007/s00170-005-0087-9. doi:10.1007/s00170-005-0087-9.

[9] F. Jiang, L. Li, Y. Tang, H. Zhang, X. Liu, A Facility Layout Algorithm for Logistics Scenarios Driven by Transport Lines, Applied Sciences 13 (2023) 7215. URL: https://www.mdpi.com/2076-3417/13/12/7215. doi:10.3390/app13127215, number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

[10] F. Jiang, L. Li, Y. Tang, H. Zhang, X. Liu, A facility layout algorithm for logistics scenarios driven by transport lines, Applied Sciences 13 (2023). URL: https://www.mdpi.com/2076-3417/13/12/7215. doi:10.3390/app13127215.
[11] S. Safarzadeh, H. Koosha, Solving an extended multi-row facility layout problem with fuzzy clearances using GA, Applied Soft Computing 61 (2017) 819–831. URL: https://linkinghub.elsevier.com/retrieve/pii/S1568494617305422. doi:10.1016/j.asoc.2017.09.003.
[12] K. Bouramtane, S. Kharraja, J. Riffi, O. El Beqqali, A. Chraibi, A comprehensive review of static and dynamic facility layout problems, Annual Reviews in Control 58 (2024) 100970. URL: https://linkinghub.elsevier.com/retrieve/pii/S1367578824000397. doi:10.1016/j.arcontrol.2024.100970, publisher: Elsevier BV.
[13] M. Xu, Z. Mei, S. Luo, Y. Tan, Optimization algorithms for construction site layout planning: a systematic literature review, Engineering, Construction and Architectural Management 27 (2020) 1913–1938. URL: https://www.emerald.com/insight/content/doi/10.1108/ecam-08-2019-0457/full/html. doi:10.1108/ECAM-08-2019-0457, publisher: Emerald Publishing Limited.
[14] P. Burggräf, T. Adlon, V. Hahn, T. Schulz-Isenbeck, Fields of action towards automated facility layout design and optimization in factory planning – A systematic literature review, CIRP Journal of Manufacturing Science and Technology 35 (2021) 864–871. URL: https://linkinghub.elsevier.com/retrieve/pii/S1755581721001656. doi:10.1016/j.cirpj.2021.09.013.
[15] Z. Yang, Y. Wang, Y. Huang, Z. Guo, W. Shi, X. Han, L. Feng, L. Song, X. Liang, J. Tang, OptiBench Meets ReSocratic: Measure and Improve LLMs for Optimization Modeling, 2025. URL: http://arxiv.org/abs/2407.09887. doi:10.48550/arXiv.2407.09887, arXiv:2407.09887 [cs].
[16] B. Hu, B. Yang, A particle swarm optimization algorithm for multi-row facility layout problem in semiconductor fabrication, Journal of Ambient Intelligence and Humanized Computing 10 (2019) 3201–3210. URL: https://doi.org/10.1007/s12652-018-1037-3. doi:10.1007/s12652-018-1037-3.
[17] S. Feuerriegel, J. Hartmann, C. Janiesch, P. Zschech, Generative AI, Business & Information Systems Engineering 66 (2024) 111–126. URL: https://doi.org/10.1007/s12599-023-00834-7. doi:10.1007/s12599-023-00834-7.
[18] J.-P. Disselkamp, D. Kurpick, B. Schutte, A. Hovemann, R. Dumitrescu, Use Cases of Generative AI in Factory Planning: Potential and Challenges, in: DS 130: Proceedings of NordDesign 2024, Reykjavik, Iceland, 12th - 14th August 2024, 2024, pp. 196–205. URL: https://www.designsociety.org/publication/47614/Use+Cases+of+Generative+AI+in+Factory+Planning%3A+Potential+and+Challenges. doi:10.35199/NORDDESIGN2024.22.
[19] M. Keshavarzi, M. Rahmani-Asl, GenFloor: Interactive generative space layout system via encoded tree graphs, Frontiers of Architectural Research 10 (2021) 771–786. URL: https://linkinghub.elsevier.com/retrieve/pii/S2095263521000443. doi:10.1016/j.foar.2021.07.003.
[20] Y. Hu, S. , Yicheng, W. , Xiaojian, W. , Sen, , J. Bao, Shopfloor layout generation method based on large language models, International Journal of Computer Integrated Manufacturing 0 (????) 1–19. URL: https://doi.org/10.1080/0951192X.2025.2482258. doi:10.1080/0951192X.2025.2482258, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/0951192X.2025.2482258.
[21] M. Kmiecik, Creating a genetic algorithm for third-party logistics' warehouse delivery scheduling via a large language model, Journal of Modelling in Management 20 (2025) 1138–1162. URL: https://www.emerald.com/insight/content/doi/10.1108/jm2-06-2024-0192/full/html. doi:10.1108/JM2-06-2024-0192, publisher: Emerald Publishing Limited.
[22] M. Chernigovskaya, D. Walia, K. Neumann, A. Hardt, A. Nahhas, K. Turowski, Towards a Standardized Business Process Model for LLMOps:, in: Proceedings of the 27th International Conference on Enterprise Information Systems, SCITEPRESS - Science and Technology Publications, Porto, Portugal, 2025, pp. 856–866. URL: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0013377700003929. doi:10.5220/0013377700003929.
[23] C. Shi, P. Liang, Y. Wu, T. Zhan, Z. Jin, Maximizing User Experience with LLMOps-Driven Personalized Recommendation Systems, 2024. URL: http://arxiv.org/abs/2404.00903. doi:10.48550/arXiv.2404.00903, arXiv:2404.00903 [cs].
[24] The Association of German Engineers (VDI), VDI 5200 Blatt 1: Factory planning - Planning procedures, ????
[25] J. Diaz-de Arcaya, A. I. Torre-Bastida, G. Zárate, R. Miñón, A. Almeida, A Joint Study of the Challenges, Opportunities, and Roadmap of MLOps and AIOps: A Systematic Survey, ACM Computing Surveys 56 (2024) 1–30. URL: https://dl.acm.org/doi/10.1145/3625289. doi:10.1145/3625289.
[26] X. Li, S. Wang, S. Zeng, Y. Wu, Y. Yang, A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges, Vicinagearth 1 (2024) 9. URL: https://doi.org/10.1007/s44336-024-00009-2. doi:10.1007/s44336-024-00009-2.
[27] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Wang, S. Jin, E. Zhou, R. Zheng, X. Fan, X. Wang, L. Xiong, Y. Zhou, W. Wang, C. Jiang, Y. Zou, X. Liu, Z. Yin, S. Dou, R. Weng, W. Cheng, Q. Zhang, W. Qin, Y. Zheng, X. Qiu, X. Huang, T. Gui, The Rise and Potential of Large Language Model Based Agents: A Survey, 2023. URL: http://arxiv.org/abs/2309.07864. doi:10.48550/arXiv.2309.07864, arXiv:2309.07864 [cs].
[28] Y. Cheng, C. Zhang, Z. Zhang, X. Meng, S. Hong, W. Li, Z. Wang, Z. Wang, F. Yin, J. Zhao, X. He, Exploring Large Language Model based Intelligent Agents: Definitions, Methods, and Prospects, 2024. URL: http://arxiv.org/abs/2401.03428. doi:10.48550/arXiv.2401.03428, arXiv:2401.03428 [cs].
[29] LangGenius, Dify: An open-source llm app development platform, https://github.com/langgenius/dify, 2025. Accessed: 2025-06-21.
[30] P. Bennet, C. Doerr, A. Moreau, J. Rapin, F. Teytaud, O. Teytaud, Nevergrad: black-box optimization platform, ACM SIGEVOlution 14 (2021) 8–15. URL: https://dl.acm.org/doi/10.1145/3460310.3460312. doi:10.1145/3460310.3460312, publisher: Association for Computing Machinery (ACM).
[31] N. Belle, D. Barnes, A. Amayuelas, I. Bercovich, X. E. Wang, W. Wang, Agents of Change: Self-Evolving LLM Agents for Strategic Planning, 2025. URL: http://arxiv.org/abs/2506.04651. doi:10.48550/arXiv.2506.04651, arXiv:2506.04651 [cs].
[32] L. Dibiaggio, L. Nesta, S. Vannuccini, European Sovereignty in Artificial Intelligence: A Competence-Based Perspective, 2024. URL: https://www.ssrn.com/abstract=5061172. doi:10.2139/ssrn.5061172.