Implementation of an Orchestrated Multi-agent System for PPCP in Textile MSMEs: A Prompt Engineering Approach

Gabriel R. B. da Silva(✉), Itamir de M. B. Filho, André M. Gurgel, Lhayana V. de Macedo, Suellen R. B. Ferraz, Beatriz S. de Moura, Ramon S. Malaquias, and Jean M. M. Lima

Digital Metropolis Institute (IMD), Federal University of Rio Grande do Norte (UFRN), Natal, Brazil
{gabriel.ribeiro.099,suellen.ferraz.113,beatriz.moura.116}@ufrn.edu.br, {itamir.filho,jean.lima}@imd.ufrn.br, andre.gurgel@ufrn.br, lhayana11@gmail.com, ramon@nutit.com.br
https://www.metropoledigital.ufrn.br/portal/

Abstract. The textile sector, especially Micro, Small and Medium Enterprises (MSMEs), faces high pressure to operate efficiently and flexibly. Conventional manufacturing setups do not allow real-time rescheduling with traditional PPCP systems. Moreover, monolithic LLM-based approaches frequently suffer from arithmetic hallucinations and high inference latency. To address this gap, the current paper proposes a multi-agent architecture orchestrated by LangGraph in order to compute a solution for dynamic PPCP. Empirical results showed that the approach can completely eliminate arithmetic hallucinations with 100% accuracy while significantly reducing inference time and token usage by 79% and 75% respectively. The software allows for flexible processing of unstructured orders, optimizes setups and uses a decision support system for dynamic reallocation of operators. The current system democratizes Industry 4.0 technology for MSMEs, unlocking smart and resilient management of manufacturing.

Keywords: Large Language Models · Textile · Production Planning and Control Processes · Multi-Agent Systems · MSMEs · Artificial Intelligence · Workload Balancing · Optimization Algorithms · Prompt Engineering

1 Introduction

The intensification of global competition and demand volatility impose on manufacturing industries the need for rigorous Production Planning and Control Processes (PPCP). In practice, PPCP operates in complex and dynamic operational scenarios, being responsible for coordinating the shop floor, optimizing work order sequencing, and managing finite resources, such as fluctuations in workforce availability and equipment failures [22,23].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Gervasi et al. (Eds.): ICCSA 2026, LNCS 16768, pp. 157–172, 2027.
https://doi.org/10.1007/978-3-032-30488-9_10

As analyzed by Gereffi and Frederick [8], in the textile industry, fast fashion requires the continuous launch of products in a large variety of styles and small lots, unlike traditional approach which relied on replenishment of basic items. Due to this, there occur very short lifecycles of the products and high operational volatility. As a result of this operation model and the repercussions of the financial crisis after 2008, the economic pressure on the textile sector has been exacerbated. As a result, flexibility production, and cost reduction became important conditions for the company. As per Gereffi and Frederick [8], the sector has to strategically aim at investing in advanced technologies and the adoption of flexible production systems for its long-run viability.

There's particularly significant dependence on it when we consider the sector of Micro, Small and Medium Enterprises (MSMEs) [16]. Indeed, the deployment of intelligent PPCP models which promote productivity increases and also provide greater flexibility and cost reduction constitutes more than just a competitive weapon, it represents a condition for survival, profitability and the sustainability of its activities [22]. But all these steps in that direction require tackling major difficulties. As stressed by Moeuf et al. [16], numerous internal characteristics prevent MSMEs from embracing Industry 4.0, such as the lack of technical skills, limited means to develop R&D activities, and the difficulty to manage advanced computer architectures.

According to Giri et al. [9], in the textile sector, embracing AI methods means going beyond incremental updates. This leap is essential if the industry wishes to grow profitability and stay competitive. Despite that, there's a technological discrepancy between the industry and academic circles. Modern AI methods continue to exist largely in the academic domain, while the industry is still content using traditional computational algorithms, leaving the sector vulnerable in the face of market instability that is inherent to the supply chain for fashion and apparel items.

In view of the previous statements, this article proposes an AI agent-based system that will manage production order examination, the appropriate distribution of workers and the creation of management reports according to the textile MSMEs' empirical output figures.

This solution has been devised to address and improve the PPCP in this niche to thereby increase overall worker throughput and efficiency, reduce idle time of factory resources and maximize their earnings. Section 2 of this article will cover previous research in the area. Section 3 will describe the system's structure, detailing the rationale behind the use of LLMs for managing factory resources. Section 4 will showcase the system's outputs and outcomes. Finally, section 5 of will present the challenges and unique contributions of this study.

2 Related Works

The use of AI in PPCP is a revolutionary step towards Industry 4.0, especially to textile MSMEs which should, at the same time, be operational efficient and cost efficient [16,22] to base the architectural choices of the suggested system and map the state. Therefore, a Systematic Literature Review (SLR) was chosen to

review scientific articles, applying the process proposed by Kitchenham [13] to minimize researchers subjectivity and increase the credibility of the findings due to its rigid and auditable process [14]. This SLR was designed to explore, survey and aggregate scientific publications concerned with PPCP models, artificial intelligence and optimization techniques applicable to textile MSMEs.

The study was undertaken in the context of addressing the primary problem question of identifying which models were the most efficient at workload distribution in the textile sector, this was addressed through six sub-question topics ranging from the best-performing optimization algorithms, to methods of Artificial Intelligence use and automation in manufacturing, to issues within the structures that underpin textile production as a PPCP for the textile SMEs. A summary review of the resulting questions, as derived from these research questions, is below.

The results converge toward the adoption of hybrid models as the most effective approach for workload balancing in textile MSMEs [10,15,17,20,22,24,26,29], with a predominance of dedicated heuristics and Genetic Algorithms (GA) [22,24], although a clear trade-off between computational time and solution quality persists [4,5,15,17,20,25,26].

In terms of automation, the literature mentions modeling with the support of ERP/MES systems [23], Digital Twins [10], and other technologies, like RFID and Lean Six Sigma [2,17], as approaches to minimize the overhead of manual data entry and detect operational bottlenecks. With regard to the variability of demand, which is an inherent feature of the industry, simulation-optimization methods with the usage of Artificial Neural Networks (ANNs) and GAs are distinguished by the ability to represent non-stationary and stochastic behavior [22,24]. The influencing factors on line balancing range between operational, labor variability, setup times, sequence of tasks [22,23,26], and structural ones, including poor layout and limited processing capacity [7,10].

Nevertheless, one gap was found: none of the studies included in the review mentioned the use of Large Language Models in PPCP systems, this is an uncharted area in the existing literature, which directly prompts the current work.

In short, hybrid genetic algorithms along with complementary technologies are promising in areas of load balancing, optimization of resources as well as simulation of production. However, GAs have the tendency of premature convergence to local optima, leading to inadequate accuracy in calculations in their output [24]. Moreover, their application in very dynamic production settings, like sewing workshops, is critically constrained: GAs, as mentioned by Zhang et al. [28], can solve large-scale combinatorial problems, but cannot respond to real-time changes: the computation of the rescheduling process in the case of disruptions is not computationally efficient. On constant replanning scenarios, because of sudden machine failures or absences of operators, the time taken to converge at an iterative search can make an instantaneous operational response impossible.

Here, the critical advantage that the LLMs provide in the form of speed and interpretability is a cognitive agent that can dynamically reason and interpret

the production data in real-time, thus eliminating the rigidity of the traditional automation systems. These models create process plans and allocation advice in seconds using semantic instructions with reported inference latencies of 100ms to 10 s [27], which are agile enough to support operation replanning in MSMEs. In addition to this speed, the capability to convert the complex production constraints into understandable justifications in a human readable form [12] is the key to the viability of PPCP. This system allows managers to take almost instantaneous decisions and keep the production running in case of disruption, without losing the human factor as the core of the decision process. Based on this, the proposed research aims to fill this gap created by the lack of presence of the LLMs in the domain of PPCP, by deploying an LLM-based PPCP system in the textile MSMEs.

3 Architecture

In order to apply the LLM-based PPCP solution, a modular design based on the Orchestrator-Worker design pattern was chosen, in accordance with the principles presented by Adimulam et al. [1] and Yeddula et al.[1]. This framework is explained by the qualitative consistency of multi-agent orchestration compared to the single-agent systems which often generate superficial or patchy recommendations in application settings. To counter this weakness, Drammeh [6] show that deterministic behavior and predictability needed to enable production-ready environments are attained by decomposing tasks into single-objective prompts that are focused and deterministic in nature. Although multifaceted multi-objective prompts enhance the variability and ambiguity of outputs, shorter, specialized prompts (50–100 tokens) decrease the cognitive load per inference and reduce hallucinations.

Also, orchestration, as introduced by Qi et al. [18], helps the system react to demand fluctuations and environmental changes without necessarily re-initiating the process, and it mitigates the inefficiencies of the traditional planning models. Technical implementation uses the LangChain model of LLM and tool coordination, and LangGraph library of state and execution flow in the computational graph. To empower the agents, we leveraged GPT-4o-mini as our chosen LLM. The criteria for this selection were its low inference latency, affordability and ample reasoning power required for the specific domain tasks at hand, moreover, it possesses multimodal capabilities which support the computer vision infrastructure involved in the extraction of information from PDF scans.

The system architecture[2] was developed keeping in mind the flexibility and scalability. The core of the application, which is accessed through a Python

[1] Yeddula, S., Dhinakaran, A., and Chavali, S. (2025). *Orchestrator-Worker Agents: A Practical Comparison of Common Agent Frameworks*. Arize AI Blog. Available at: https://arize.com/blog/orchestrator-worker-agents-a-practical-comparison-of-common-agent-frameworks/.

[2] The image of the architecture diagram is available at: https://drive.google.com/file/d/11_lIWtNWYlXRXddyCxeLgBArUT-EmHrV/view.

API (FastAPI), communicates with a hybrid persistence layer that uses PostgreSQL for structured data storage (production orders, collaborators, etc.) and MongoDB for unstructured data storage (production logs). The submission of a natural language query triggers the workflow, and is intercepted by the Orchestrator Agent.

The Orchestrator Agent is the main routing element of the system that uses the reasoning features of the LLM to interpret the semantic intent of the query made by the user. On the basis of this analysis, it identifies-by conditional edges within the computational graph of LangGraph (the specialized workflow to be activated). The elements that make up this architecture are as follows:

- **Orchestrator Agent:** The core routing module that uses FastAPI, LangChain and LangGraph. It has the role of accepting the requests of the external system, decoding the intent of the user, and delegating the work to the respective specialized agent.
- **Production Order Registration Agent:** This is created to remove manual data entry of production order forms to reduce transcription errors and standardize data entry. It scans sewing operations and standard times and production phases off technical specification sheets in PDF format, and authenticates the extracted data using Pydantic and Guardrails AI before permitting persistence.
- **Setup Suggestion Agent:** It works during the planning stage of the production line, suggesting a smart and optimized distribution of operators prior to the initiation of production. In order to accomplish this, it uses historical information to construct a personal efficiency model of each collaborator, calculated through cross-referencing of MongoDB production logs to postgreSQL standard times and aligning the models with the needs of the new production order.
- **Operator Reallocation Agent:** It is dedicated to the dynamic management of human resources in the production process and is involved in targeted workforce changes to eliminate bottlenecks or reorganize workloads in real-time. It examines the present situation on the production floor and confirms competencies in accordance with the personal experience of collaborators on the basis of individual tasks.
- **Operation Report Agent:** The duties of the agent is to convert raw data of production to operational intelligence. It takes all logs taken by smart devices and puts them in MongoDB, compares real timestamps with standard times, stored in PostgreSQL and determines the efficiency of individual operators and the overall efficiency of production orders. The findings, such as identified productivity variations, are organized into an executive summary delivered to the outside system through API and use it to create a PDF report.
- **Response Formatter:** The role of the Response Formatter is to standardize and consolidate the output of all the specialized agents. It ensures that answers are sent to the orchestrator using a standardized data contract, and communication between the calling system and the orchestrator is consistent, despite the agent that handled the request.

The architecture has a common PDF ingestion pipeline to seamlessly incorporate the processing of attached documents. The system identifies the information on the technical specification sheet irrespective of the invoked workflow and applies the same hybrid approach: native text extraction is initially tried with PyMuPDF (`page.get_text()`); in case the document is a scanned PDF (or no text is found in the scanned document), a computer vision pipeline is turned on. Here, pages are rendered at high resolution (432 DPI), processed with OpenCV (CLAHE, bilateral filter, and adaptive binarization) and processed with GPT-4o-mini in multimodal vision mode. The production in both roads is strictly verified by Pydantic and Guardrails AI models.

This extraction pipeline is single, however, the objective of the extracted data varies fundamentally with the active agent. The operations extracted in the Production Order Registration Agent are given to make a new record in the database. Conversely, with the Setup Suggestion Agent, the extraction is a lookup key: the operations are employed to find which existing-registered production order in PostgreSQL is related to the document posted.

This manner of orchestration of specialized agents, through the LangGraph computational graph, goes beyond the mere automation of individual tasks. The suggested architecture defines a cooperative ecosystem whereby the reasoning ability of the LLM is guided to address domain-specific issues and optimize results. The scalability of the system is not only guaranteed by this approach to the possibility of adding new agents in the future, but also enables the flexibility needed to meet the variability that is inherent to the textile industry.

3.1 Prompt Optimization Strategy

The optimization strategy of the processing flow is one of the main architectural contributions of this system. The project, initially, had a monolithic approach where the LLM did all the strategic analysis and also implementation of mathematical calculations, which are part of PPCP. This involved a lot of prompts that entailed specifications on how to draw up cycle time formulations and critical path computations. But the literature suggests that prompt complexity has a detrimental effect on the quality of analysis when overdone [3], and that generative models such as GPT-4o-mini actually have limitations in their ability to solve multi-step mathematical tasks [11,21].

The following theoretical vulnerabilities proved empirically in the process of testing the monolithic version of the system. This unconstrained transfer of mathematical operations to the model led to the accuracy of calculating within a range of about 95% due to the occasional arithmetic hallucinations. In addition to the intolerable imprecision in industrial settings, the reasoning overhead of the LLM increased the average inference time to about 11 s. In order to overcome these bottlenecks, the architecture was refined to form a strict separation of deterministic operations.

In this modified design, essential measurements are computed beforehand by autonomous server-side calculator modules. This structure ensures deterministic execution, unlike generative models. To take an example, the time that

each single production cycle takes is calculated directly out of the non-relational database logs as follows:

$$D_{min} = \frac{T_{actual} - T_{previous}}{60000} \tag{1}$$

where T_{actual} and $T_{previous}$ are the time-stamps of consecutive events. The final numerical results are only injected into the context prompt after the parameters like per-cell time, bottlenecks and critical path have been calculated precisely.

This disjunction of interests ensures 100% mathematical correctness of the operational measures and a decrease of 1,500 to about 350 tokens per call. Thus, the average inference time is reduced to 2.3 s, and the LLM can only work within its field of competence, which is semantic interpretation of structured information and developing adaptive strategies in natural language. Lastly, the system avoids structural non-determinism of LLMs by delegating strict calculation logic to deterministic algorithms, which is necessary to assure reproducibility and auditability in industrial manufacturing settings [19].

4 Results and Discussion

Two complementary analytical dimensions are used to describe the results of the proposed PPCP system obtained by empirical testing of the proposed system in this section. Section 4.1 compares the computational performance of the architectural refactoring with the monolithic and decoupled methods against monolithic and decoupled latency, token consumption, cost efficiency, and mathematical accuracy metrics.

Section 4.2 studies the feasibility of the solution in a practical scenario in a textile manufacturing setting where the multi-agent system is orchestrated to run a simulated case study. This design is intended to authenticate the technical stability of the timely optimization plan as well as the operational feasibility and strategic profits of the dynamic workload balancing in the MSMEs.

4.1 Performance Evaluation: Monolithic vs. Decoupled Architecture

In experimental benchmark analyses performed using the GPT-4o-mini as the base model, a comprehensive assessment comparing the two methods is detailed in Table 1. These results clearly show a significant improvement across all performance and dependability benchmarks post-decoupling. Average response latency dropped from 11.0 ± 1.5 s to 2.3 ± 0.1 s. This is aligned with literature that states that timely complexity is a predictive factor in the processing time of language models [3].

In terms of resource consumption, the chosen prompt engineering strategy allowed a significant decrease in the amount of information handled with each call. Mean token usage decreased from $1{,}250 \pm 250$ to 350 ± 50, which is more than three-quarters of the cost per call to the API. The applicability of the solution

Table 1. Performance and Reliability Comparison: Monolithic vs. Decoupled Architecture (Mean ± SD)

Metric	Monolithic Approach	Decoupled Approach
Average Latency (s)	11.0 ± 1.5	2.3 ± 0.1
Token Consumption (units)	1250 ± 250	350 ± 50
Estimated Cost (USD)	0.0035 ± 0.0015	0.00075 ± 0.00025
Calculation Accuracy (%)	95%	100%
Numerical Reproducibility	Variable	1.00

to the MSMEs requires this optimization: on the one hand, the financial savings are important, and on the other hand, a surplus of information in a prompt can overload the model, adversely affecting the accuracy and consistency of the responses [3].

And lastly with regards to integrity of PPCP operations, the decoupled method met 100% mathematical accuracy and 1.00 numerical reproducibility. Conversely, the monolithic method was 95% accurate and had arithmetic hallucinations and structural non-determinism of the LLMs. The literature suggests that the correctness of the responses produced by intelligent systems can be improved only through more specific, well-structured prompts [3]. The proposed system therefore guarantees the reliability in industrial textile manufacturing settings by isolating deterministic calculations and semantic interpretation.

4.2 Case Study: Dynamic Orchestration in a Textile MSME

This paper provides qualitative evaluation through a case study. Data was collected in a simulated shop floor setting by using real data retrieved from the textile MSME. The dataset consists of production orders (POs) in unstructured PDF, collaborator performance history and logs from production smart devices. In the authors setting, every station on the shop floor has some smart device which logs production events happening at this station in real time. First, the system receives a production order in PDF format.

In the manual process, to register such order, one has to enter more than $7+3N$ fields, where N is the number of production operations. Given the example PO had 12 operations, this amounts to $7 + 3 * 12 = 43$ fields to manually fill, which often results in transcription mistakes due to cognitive load and the high number of data to enter.

We show the step-by-step operation of the solution, from initial document ingestion to the final efficiency report, in Fig. 1. The detailed explanation of each phase is provided in the following subsections.

Intelligent Digitization via Production Order Registration Agent. In the simulated scenario, the technical specification sheet for the production

order was given as a scanned PDF document. Since it couldn't contain native extractable text, the Production Order Registration Agent classified it as a scanned document and invoked the computer vision path. Pre-processed image representations of the technical specification sheet were fed into the GPT-4o-mini vision model. Through its object detection capabilities, it properly identified its vertical hierarchical arrangement of sewing operations within each phase by utilizing defined spatial mapping rules applied via prompt engineering.

As a consequence, 12 operations within the four defined production phases were successfully extracted without any error relative to the original document. The resulting structured information (shown below) was validated by the Pydantic/Guardrails AI layer and recorded into the database.


```
{
  "order_reference": "ABC1001",
  "target_quantity": 100,
  "extracted_phases": {
    "Phase 1 - Pre-assembly": [
      {"operation": "Coin pocket hem", "standard_time_min":
          0.06},
      {"operation": "Back pocket hem", "standard_time_min":
          0.125},
      {"operation": "Close/turn front split pocket", "
          standard_time_min": 0.155},
      {"operation": "Attach flap and facing on split pocket",
           "standard_time_min": 0.135},
      {"operation": "Topstitch pocket mouth", "
          standard_time_min": 0.23},
      {"operation": "Attach front/side pockets", "
          standard_time_min": 0.185},
      {"operation": "Attach pockets to opening", "
          standard_time_min": 0.185}
    ],
    "Phase 2 - Front": [
      {"operation": "Attach pockets", "standard_time_min":
          1.85},
      {"operation": "Final inspection", "standard_time_min":
          1.11}
    ],
    "Phase 2 - Back": [
      {"operation": "Attach and topstitch zipper", "
          standard_time_min": 0.44},
      {"operation": "Attach yoke", "standard_time_min": 0.75}
    ],
    "Phase 3 - Assembly": [
      {"operation": "Count pieces", "standard_time_min":
          0.58}
    ]
  }
}
```

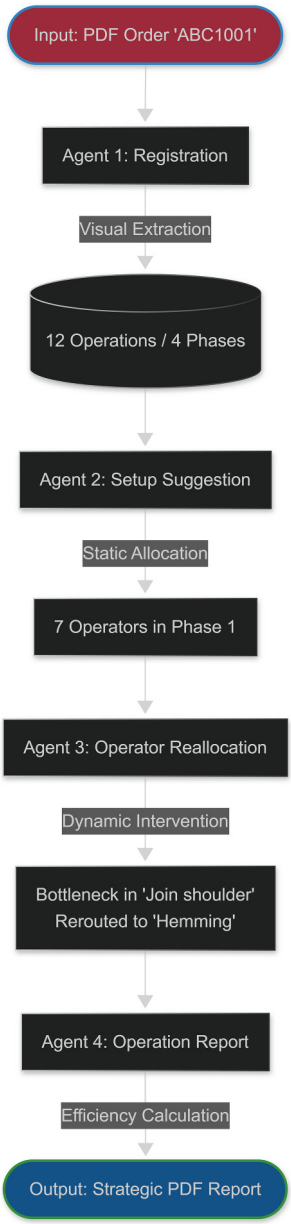

Fig. 1. Workflow of the proposed multi-agent architecture in the simulated case study.

With this Human-in-the-Loop (HITL) paradigm, the production manager checks the extracted data rather than filling the 43 fields by hand, removing the burden of the process, and preventing critical registration errors.

Setup Optimization via Setup Suggestion Agent. The structured data generated from POs is then input into the Setup Suggestion Agent. The primary function of this agent is to make a global allocation of human resources available on the production line to the available operational needs that are recorded in the database through the history of collaborator performances.

In this experiment, 7 operators were available to handle the 12 operations identified in the mapping step. The agent attempts a simultaneous assignment of all operation tasks based on the matching process that compares the operation name in the technical documentation for each operation and the history that is saved per operator, respectively. In this case, only the Phase 1 operations found matches between the operation names from the technical sheet and the histories recorded per collaborator, therefore, the 7 operators were assigned to the 7 simultaneous operations of that phase. The operator best matched for each specific operation and smart device was chosen and assigned according to their competency profile. Operations from Phases 2 and 3 (that didn't have any matched names on the operator histories) were assigned using the fallback policy, therefore, operators without any historical matches to any specific operations were allocated and noted with an asterisk () on the table showing the generation of the plan to demonstrate that no specific operator was specialized on any operation.

From the generated plan, the deterministic calculator estimated a total time required to produce 100 items of 4.43 h, identifying the operation "Attach pockets" in Phase 2 (taking 1.85 min) as the bottleneck. In addition, the idleness projected for the operations to be completed after this identified bottleneck operation illustrated the limitations that result from having a plan set up statically, just by matching historical performances, which serves as one architectural feature that allows the dynamic reallocation agent to intervene in case of unforeseen events.

Dynamic Intervention via Operator Reallocation Agent. Despite that the planner may statically assign operators to each operation, the dynamic changes on the shop floor, like early batch completions and emergent bottleneck situations, necessitate real-time adjustments to such assignments. The implemented architecture addresses this requirement through an Operator Reallocation Agent that handles the movement of operators across operations according to the PO's progress and each collaborator's past assignments.

This agent was validated with a simulated bottleneck event in which the system was tasked to identify the optimal transition of the operator currently assigned to the "Join shoulder" operation, which was identified as a bottleneck. The system looked up the PO and found three possible destination operations: "Hemming" (0.75 min), "Double stitch" (1.0 min) and "Close side seam" (0.5 min).

It is important to note that a cross-reference to the database showed that the operator didn't have experience assigned to any of the three options (`has_experience: false`). With classical rule-based systems, this usually leads

to an allocation failure. The system overcomes this problem with a simple heuristic: if the operator doesn't have prior experience in any alternative option, it chooses the first operation after ordering the operations alphabetically. The recommendation is as shown in the JSON object below:


```
{
  "collaborator": "Collaborator A",
  "current_operation": "Join shoulder",
  "total_options": 3,
  "options": [
    {
      "operation": "Hemming",
      "suggested_time": 0.75,
      "has_experience": false,
      "best_option": true,
      "justification": "Best available option: operator lacks
          prior experience in alternatives. Selected based
         on suggested time balancing."
    },
    {
      "operation": "Close side seam",
      "suggested_time": 0.50,
      "has_experience": false,
      "best_option": false
    }
  ]
}
```


The system not only processes the main data used by the front-end, but it also creates a simple summary for the production manager. This summary is written in plain language and is made by the GPT-4o-mini model. It looks at the options that have already been ranked by an algorithm. In the scenario that was tested, the model found that while the "Hemming" operation would make the workload more even, moving tasks to "Close side seam" would cut production time by up to 0.5 min per unit. This would get the most work done right away.

This system gives us two things: a standardized algorithmic recommendation (`best_option`) and a semantic tactical analysis generated by the LLM, endows the system with the characteristics of a robust Decision Support System (DSS). Production management ceases to be a subjective visual inference and becomes data-driven, enabling smooth workforce redistribution in MSMEs even in the face of competency profile uncertainty.

Efficiency Analysis via Operation Report Agent. The last step of the process is to take the information we got from making things and turn it into something we can actually use. When the production order is done, the Operation Report Agent makes sure the production record in MongoDB matches the standard record in PostgreSQL. It then fixes the timestamps and figures out how

well each operator did their job, compared to how long it was supposed to take. It also looks at how efficient the whole purchase order was.

In the simulated experiment, the reporting system accurately alerted the agent about this situation: the operation, designated "Close back pocket" had a expected completion time of 4 min, while its logged processing time was 0.07 min. Thus, an efficiency of 5.714% was computed. While the computation is technically correct, such high values typically reveal outdated standard times, possibly reflecting outdated technical instructions that might not incorporate recent innovations like rapid stitching processes designed to accommodate higher volumes of individual clothing items. Consequently, this should warrant a review of the relevant time standard. The resulting data is neatly organized into the analytical report payload displayed below, which is then fed into an external engine to generate the corresponding PDF report:


```
{
  "standard_piece_time_min": 9.0,
  "actual_operation_time_min": 0.07,
  "total_production_pieces": 2,
  "operator_performance_map": [
    {
      "operator": "Collaborator A",
      "operation": "Close back pocket",
      "standard_time_min": 4.0,
      "actual_time_min": 0.07,
      "efficiency_percent": 5714.3
    },
    {
      "operator": "Collaborator B",
      "operation": "Overlock fly",
      "standard_time_min": 5.0,
      "actual_time_min": 0.07,
      "efficiency_percent": 7142.9
    }
  ]
}
```


The data is put together in a clear and easy-to-understand report, and another system uses this information to create a PDF report through an interface. This makes it simple for the manager to check the report and see overall efficiency numbers, how different operations compare, and exactly how much a task has improved compared to the standard time. The report provides a straightforward way to measure and compare performance, making it easier to identify areas for improvement and track progress over time. With this information, the manager can make informed decisions and take steps to optimize operations and increase efficiency.

The system takes care of production documents from start to finish. It begins with unorganized documents, then plans and makes adjustments as needed, and finally creates strategic reports, thereby defining a valid infrastructure to modernize PPCP within textile MSMEs.

5 Conclusion and Future Work

This study put forward and confirmed a multi-agent design, managed by LangGraph and driven by LLMs, specifically engineered for the dynamic enhancement of PPCP within textile MSMEs. The core innovation of this architecture lay in separating the traditional mathematical operations from the stage of semantic interpretation. This method effectively countered the arithmetic inaccuracies often seen in generative models, guaranteeing perfect accuracy in structural computations. Simultaneously, it achieved a 79% reduction in inference time and a 75% decrease in token usage.

The real-world utility of this solution was illustrated via a simulated case study, which incorporated actual operational data from a textile MSME. The system demonstrated its capacity to manage the full production life cycle, starting with the multimodal input and semantic extraction of Production Orders directly from PDF documents, thereby eliminating the need for manual data entry, and extending to the creation of strategic efficiency reports. The Agent designed for Operator Reallocation proved especially beneficial, functioning as a DSS, it navigated the lack of historical operator data by employing a deterministic heuristic. This approach enabled the generation of tactical recommendations, delivered in natural language through an LLM.

A significant hurdle during the development process stemmed from the limited academic research specifically linking LLMs to PPCP in manufacturing settings, with an even greater deficit concerning textile MSMEs. This gap, identified during a thorough literature review, directly informed the architectural choices made for this project. Consequently, the proposed architecture aims to contribute to this developing field, showing that integrating semantic inference with deterministic calculation offers a practical and scalable pathway for modernizing production planning in smaller enterprises.

Looking ahead, the research team intends to move the architecture beyond its simulated environment and implement it in actual manufacturing settings. This will allow for a thorough evaluation of its performance under genuine shop-floor conditions. Such practical validation will generate crucial feedback, guiding the incorporation of new features and refinements specifically tailored to the unique operational demands faced by textile MSMEs.

Acknowledgments. This study was supported by the CNPq, National Council for Scientific and Technological Development - Brazil (grant number 443723/2024-9).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Adimulam, A., Gupta, R., Kumar, S.: The orchestration of multi-agent systems: architectures, protocols, and enterprise adoption. arXiv preprint arXiv:2601.13671 (2026). https://arxiv.org/abs/2601.13671
2. Bai, Y., You, J.B., Lee, I.K.: Design and optimization of smart factory control system based on digital twin system model. Math. Probl. Eng. **2021**(1), 2596946 (2021)
3. Balaji, K., Lokesha, A., G., C., H. M., P.: Optimizing prompt length and specificity for enhanced ai chatbot responses. Int. J. Comput. Sci. Res. Rev. **07** (2024). https://ijcsrr.org/wp-content/uploads/2024/09/61-3009-2024.pdf
4. Darmian, S.M., Farughi, H.: An optimization-based approach for managing downstream logistic operations: a real application in pandemic outbreak conditions. Comput. Ind. Eng. **171**, 108495 (2022)
5. De Oliveira, U.R., de Souza, H.A.D., Menezes, C.A.G., Rocha, H.M.: Straightening machine preventive maintenance intervention plan based on AHP: a case study in a steel company in brazil. Oper. Manag. Res. **16**, 1577–1593 (2023)
6. Drammeh, P.: Multi-agent LLM orchestration achieves deterministic, high-quality decision support for incident response. arXiv preprint arXiv:2511.15755 (2025)
7. Eccher, C., Geraghty, J.: Incorporating sustainable criteria in a dynamic multi-objective recommendation planning tool for a continuous manufacturing process: a dairy case study. J. Manuf. Syst. **55**, 159–170 (2020)
8. Gereffi, G., Frederick, S.: The global apparel value chain, trade and the crisis: challenges and opportunities for developing countries. Policy Research Working Paper 5281, The World Bank (2010)
9. Giri, C., Jain, S., Zeng, X., Bruniaux, P.: A detailed review of artificial intelligence applied in the fashion and apparel industry. IEEE Access **7**, 95376–95396 (2019)
10. Guo, H., Chen, M., Mohamed, K., Qu, T., Wang, S., Li, J.: A digital twin-based flexible cellular manufacturing for optimization of air conditioner line. J. Manuf. Syst. **58**, 65–78 (2021)
11. Hoza, P.: Evaluating reasoning in large language models with a modified think-a-number game: case study. Acta Informatica Pragensia **14**(2), 246–260 (2025)
12. Keskin, Z., et al0.: LLM-enhanced human-machine interaction for adaptive decision making in dynamic manufacturing process environments. IEEE Access (2025)
13. Kitchenham, B.: Procedures for undertaking systematic reviews. Technical report. TR/SE-0401, Keele University and NICTA (2004)
14. Kitchenham, B., Brereton, O.P., Budgen, D., Turner, M., Bailey, J., Linkman, S.: Systematic literature reviews in software engineering-a systematic literature review. Inf. Softw. Technol. **51**(1), 7–15 (2009)
15. Li, D., Jin, H., Zhang, Y.: Dynamic worker allocation in seru production systems with actor-critic and pointer networks. Eur. J. Oper. Res. **324**(1), 62–74 (2025)
16. Moeuf, A., Pellerin, R., Lamouri, S., Tamayo-Giraldo, S., Barbaray, R.: The industrial management of SMEs in the era of industry 4.0. Int. J. Prod. Res. **56**(3) (2017)
17. Nedra, A., Néjib, S., Boubaker, J., Morched, C.: An integrated lean six sigma approach to modeling and simulation: a case study from clothing SME. Autex Res. J. **22**(3), 305–311 (2022)
18. Qi, Y., et al.: Leveraging LLM-based agents for intelligent supply chain planning. arXiv e-prints pp. arXiv–2509 (2025)
19. Semmelrock, H., et al.: Reproducibility in machine learning-based research: Overview, barriers and drivers. arXiv preprint arXiv:2406.14325 (2024). https://arxiv.org/html/2406.14325v2

20. Singh, J., Singh, A., Singh, H., Doyon-Poulin, P.: Implementation and evaluation of a smart machine monitoring system under industry 4.0 concept. J. Ind. Inf. Integr. **43**, 100746 (2025)
21. Singh, J., Nambi, A., Vineet, V.: Exposing the achilles' heel: evaluating LLMs ability to handle mistakes in mathematical reasoning. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics, vol. 1, pp. 27044—-27065 (2025). https://aclanthology.org/2025.acl-long.1313.pdf
22. Teerasoponpong, S., Sopadang, A.: A simulation-optimization approach for adaptive manufacturing capacity planning in small and medium-sized enterprises. Expert Syst. Appl. **168**, 114451 (2021)
23. Wang, L.C., Chen, C.C., Liu, J.L., Chu, P.C.: Framework and deployment of a cloud-based advanced planning and scheduling system. Robot. Comput. Integr. Manuf. **70**, 102088 (2021)
24. Wang, L.: Improved human resource allocation method based on genetic algorithm. In: Intelligent Computing Technology and Automation, pp. 97–104. IOS Press (2024)
25. Wang, L., Ai, W., Deng, T., Shen, Z.J.M., Hong, C.: Optimal production ramp-up in the smartphone manufacturing industry. Naval Res. Logistics (NRL) **67**(8), 685–704 (2020)
26. Wang, P.S., Lin, C.C., Yang, T.: A study on shojinka design for automobile engine piston-rod manufacturing. Comput. Ind. Eng. **151**, 106974 (2021)
27. Xia, Y., Zhang, J., Jazdi, N., Weyrich, M.: Incorporating large language models into production systems for enhanced task automation and flexibility. arXiv preprint arXiv:2407.08550 (2024)
28. Zhang, F., Mei, Y., Nguyen, S., Zhang, M.: Survey on genetic programming and machine learning techniques for heuristic design in job shop scheduling. IEEE Trans. Evol. Comput. **28**(1), 147–167 (2023)
29. Zhang, W., Hou, L., Jiao, R.J.: Dynamic takt time decisions for paced assembly lines balancing and sequencing considering highly mixed-model production: An improved artificial bee colony optimization approach. Comput. Ind. Eng. **161**, 107616 (2021)