Assembly Line Load Optimization using Machine Performance Ratio in Phone Manufacturing

GE19612 - PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP PROJECT REPORT

Submitted by

P HARIHARAVISWANATHAN	**(2116220701082)**
GOUTHAM A K	**(2116220701077)**
GIRIDHAR M	**(2116220701074)**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2025

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project titled **“ Assembly Line Load Optimization using Machine Performance Ratio in Phone Manufacturing ”** is the bonafide work of **“P HARIHARAVISWANATHAN (2116220701082), GOUTHAM A K (2116220701077) , GIRIDHAR M (2116220701074)”** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE	**SIGNATURE**
Dr. P. Kumar., M.E., Ph.D.,	Dr. AYYADURAI M,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor	Assistant Professor
Department of Computer Science	Department of Computer Science
and Engineering,	and Engineering,
Rajalakshmi Engineering College,	Rajalakshmi Engineering
Chennai - 602 105.	College, Chennai-602 105.

Submitted to Mini Project Viva-Voce Examination held on ______________

Internal Examiner **External Examiner**

ABSTRACT

Now in the Industry 4.0 era, it is paramount to optimally assemble within assembly lines, so that they are adaptable, cost efficient and have high throughput. In this project, we describe a dynamic load shaping system for the phone manufacturing based on Linear Programming (LP) to manage tasks allocation across multiple factories and lanes. Objective of the problem is to minimize production cycle time, balance the workloads with respect to new orders while also taking into consideration real time factors including availability of machines, production speed and inventory levels. It dynamically solves the optimal task assignments using LP model in real-time and with the real time data to handle disruption such as machine downtime or sudden order change. Spring Boot is used to create a backend in order to manage feeds for the production and inventory and also for factory data, where a web based dashboard offers live production monitoring and visual analytics. Examples presented show appreciable cycle time improvements, as well as their resource and delivery performance utilization. The obtained results indicate that LP based real time scheduling can offer an opportunity to increase productivity and provide support for production flexible, resilient and cost effective systems.

ACKNOWLEDGMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to **Dr. P. KUMAR, M.E., Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guides **Dr. M. AYYADURAI M** and **Mrs. G. M. Sasikala**, We are very glad to thank our Project Coordinator, **Dr. M. AYYADURAI M** Assistant Professor Department of Computer Science and Engineering for his useful tips during our review to build our project.

P HARIHARAVISWANATHAN 2116220701082

GOUTHAM A K 2116220701077

GIRIDHAR M 2116220701074

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

S. No	ABBR	Expansion
1	LP	Linear Programming
2`	ALB	Assembly Line Balancing
3	ALBP	Assembly Line Balancing Problem
4	MTO	Make To Order
5	FMS	Flexible Manufacturing Systems
6	IoT	Internet Of Things
7	CPS	Cyber Physical Systems
8	DFD	Data Flow Diagram
9	MILP	Mixed Integer Linear Programming
10	TOP	Trolley Optimization Problem
11	API	Application Programming Interface
12	DRL	Deep Reinforcement Learning
13	ERP	Enterprise Resource Planning
14	MES	Manufacturing Execution System
15	KPI	Key Performance Indicator

CHAPTER 1 - INTRODUCTION

1.1 GENERAL

In modern manufacturing environment, it is very important for efficiency, flexibility and responsiveness in order to compete well and to rapidly respond to changing customers demands. While traditional assembly line systems are very successful in mass production, they fail in achieving the objectives of high product variability, fluctuating demand and real time disruptions. In this environment of changing manufacturing conditions, there is urgency for systems to dynamically adapt to those changing conditions.

The aim of this project is to develop an optimized load shaping model for assembly line production, especially in terms of mobile phone manufacturing applications. The solution uses Linear Programming (LP) to solve for task assignment dynamically across multiple stations and factories with respect to these considerations: marginal cost, available capacity and delivery deadline. The model integrates the real time production data (machine status, inventory levels, and order priouities) in order to minimize the cycle time, higher utilisation of resources, and overall improvement in production efficiency.

For this optimization process, the system is developed with a fullstack architecture of spring boot backend, Postgres squarel database and web based user interface. The interface is real time, to see order processing, stock availability, factory performance and delivery. Such an integrated approach facilitates manufacturers‘ data driven decisions, quick reaction to production issues and ships products with higher consistency and lower cost.

1.2 OBJECTIVE

The primary goal of this project is to build a real time linear programming based assembly line optimization system that can help coordinate the production of several factories. The system is designed to dynamically assign the tasks so as to minimize the overall production cycle time by redistributing the tasks according to the real-time phenomena related to the machine availability and factory capacity, as well as tillance (delivery) deadlines. The model also incorporates marginal cost into the task assignment decisions in order to minimize operational costs and to make sure that the task assignment is balanced across all production lanes. In addition, the project seeks to increase delivery performance through timely delivery to the customer to meet the cutoff date and reduce delays. The aim is to create a scalable and user friendly platform for the live production tracking and decision making using a web based dashboard mechanism supported with a Spring Boot backend and PostgreSQL database to connect the entire system to the principle of Industry 4.0

1.3 EXISTING SYSTEM

Current methods for detecting fake social media profiles rely heavily on centralized algorithms and manual moderation, which often lack accuracy and fail to detect sophisticated fake accounts. These systems are also prone to vulnerabilities, as centralized databases can be hacked or breached, compromising user privacy and sensitive information. Additionally, they lack the transparency and security required to effectively combat the increasing prevalence of fake profiles. This inadequacy contributes to the persistence of misinformation, fraud, and declining user trust in social media platforms. As a result, there is a growing need for innovative, secure, and transparent solutions to address these challenges and ensure a safer, more reliable digital environment for users

CHAPTER 2
LITERATURE SURVEY

The assembly line balancing problem (ALBP) has been extensively studied for decades, forming the foundation of production optimization strategies. Traditional models primarily focused on minimizing the number of workstations for a given cycle time or reducing cycle time for a fixed number of stations. One of the most influential early contributions came from Hackman et al. [13], who proposed fast and effective heuristic algorithms for simple assembly line balancing problems. Their work introduced practical techniques that significantly reduced computational effort while achieving near optimal solutions, making it widely applicable in industrial settings.

To address the limitations of single-objective optimization, researchers have developed multi-criteria approaches. Jolai et al. [17] introduced a decision-making model that incorporates multiple objectives such as balance efficiency, workload smoothness, and station utilization. Their work employed a goal programming technique to prioritize these conflicting goals, offering more realistic solutions that better reflect real world manufacturing challenges. In more recent years, linear programming (LP) has emerged as a key tool for modelling and solving load shaping problems in assembly lines. Gamberini et al. [18] extended the LP approach to consider stochastic task times and ergonomic constraints, highlighting the importance of human factors in balancing decisions. Similarly, Boysen et al. provided a detailed classification of ALBP variants and solution strategies, emphasizing the versatility of LP-based models in handling complex assembly scenarios.

As manufacturing systems become increasingly digitized under the Industry 4.0 paradigm, integrating real-time data with optimization models has become a growing area of interest. Zhou et al. [19] proposed a cyber-physical system based framework using LP to adaptively re-balance assembly lines based on sensor data and machine feedback. This dynamic approach enables real-time responsiveness, improving production flexibility and resilience.

Moreover, hybrid optimization techniques that combine LP with metaheuristics such as genetic algorithms and simulated annealing have been explored to address large-scale, non-linear problems. For example, Gokc¸en et al. [20] presented a hybrid¨ goal programming model for mixed-model assembly lines, offering improved adaptability to variable demand and task durations.

Despite these advancements, significant gaps remain in the literature. Real-time adaptability, scalability for large systems, and the inclusion of sustainability goals such as energy efficiency and carbon footprint are still underexplored. These challenges motivate the present study, which aims to develop a linear programming-based framework that addresses these emerging needs in modern manufacturing environments.

CHAPTER 3

PROPOSED SYSTEM

3.1 GENERAL

Optimization of assembly line operation is achieved through dynamic task allocation using Linear Programming (LP). It takes real time data from many factories, i.e order requirements, machine availability, production speed, current inventory. The information is then processed by an LP based algorithm to solve for the recommended least amount of operational cost, from cycle time stand point, while also maintaining delivery deadlines for the given tasks with the use of as many stations and production lanes as possible.

It consists of three main components in the system architecture: The backend engine built using Spring Boot, the PostgreSQL database to store and manage the factory data; The web based frontend dashboard allowing the user to watch production activities and system decision in real time. The LP solver resides in the backend, and it dynamically allocates tasks on the basis of live input data as well as the station capacity and marginal cost of execution when constrained by precedence relations, etc.

It is a flexible and responsive system that reallocates workloads when disruptions happen, for instance machine breakdown or fast changes in order level. It scales to the number of factories it is deployed on, is aligned with industry 4.0 objectives of data analytics, coordinating smart resources planning, and real time operational visibility. In the end, the proposed system is a viable and economical approach to accommodate for both speed and cost in modern manufacturing facilities.

3.2 SYSTEM ARCHITECTURE DIAGRAM

The system architecture of the proposed Assembly Line Load Shaping model is designed around a relational database structure that enables efficient and dynamic task allocation across multiple factories and production lanes. At the core of this architecture is the factories table, which holds essential information about each manufacturing unit, including its name and location. Each factory is associated with one or more production lanes, defined in the lanes table, where actual assembly takes place. The orders table manages incoming customer requests, storing details such as the order number, customer name, order date, and current status. These orders are then mapped to specific factories using the factory_orders table and further distributed to production lanes via the lane_orders table, enabling parallel processing and reduced cycle time.

The inventory table plays a critical role in ensuring that each factory's stock is monitored in real time. It tracks the available quantity of different phone models, allowing the system to check whether an order can be fulfilled directly from existing inventory or whether new production is required. This inventory check is pivotal in the load shaping logic, as it determines the starting point of the optimization process.

When an order is received, the system evaluates all factories and their respective inventories. If the nearest factory has sufficient stock, the order is fulfilled directly. Otherwise, the system checks alternative factories, considering marginal cost, delivery constraints, and real-time availability. Once a factory is selected, the LP model determines the optimal lane distribution to fulfill the order efficiently. The entire process is orchestrated through a Spring Boot backend connected to a PostgreSQL database, while a web-based dashboard provides live monitoring of order processing, factory utilization, and lane activity. This modular and scalable architecture supports real-time decision-making, efficient resource utilization, and seamless task reallocation, aligning with the objectives of Industry 4.0 and smart manufacturing systems.

Fig 3.1: System Architecture'

3.3 DEVELOPMENTAL ENVIRONMENT

3.3.1 HARDWARE REQUIREMENTS

The hardware specifications could be used as a basis for a contract for the implementation of the system. This therefore should be a full, full description of the whole system. It is mostly used as a basis for system design by the software engineers.

Table 3.1 Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i5
RAM	8 GB RAM
PORT	USB 3.0
STORAGE	256 GB SSD or 500 GB HDD
DISPLAY	14” MONITOR - 1366x768 resolution
OS SUPPORT	WINDOWS 10

3.3.2 SOFTWARE REQUIREMENTS

The software requirements paper contains the system specs. This is a list of things which the system should do, in contrast from the way in which it should do things. The software requirements are used to base the requirements. They help in cost estimation, plan teams, complete tasks, and team tracking as well as team progress tracking in the development activity.

Table 3.2 Software Requirements

COMPONENTS	SPECIFICATION
Operating System	Windows 10/11 or macOS
Frontend	Html,Css,Thymeleaf
Backend	Flask (Python), Spring boot(Java)
Database	MySQL

3.4 DESIGN OF THE ENTIRE SYSTEM

3.4.1 ACTIVITY DIAGRAM

The activity diagram Fig 3.2 represents the workflow for detecting fake profiles using a Flask-based machine learning system integrated with blockchain security. The process begins with the user interacting via a web page, where they provide the necessary input. The Flask framework serves as the backend, passing the input to a WSGI server for handling requests. The input features submitted by the user, such as profile characteristics, are then sent for preprocessing, where tasks like data cleaning, normalization, and feature extraction are performed. These preprocessed features are passed to the machine learning (ML) algorithm with blockchain security, which processes the data using trained models to classify profiles. The system incorporates blockchain for data integrity and secure operations. Finally, the output, indicating whether the profile is "fake" or "not fake," is delivered back to the user. This streamlined process ensures efficient and secure fake profile detection

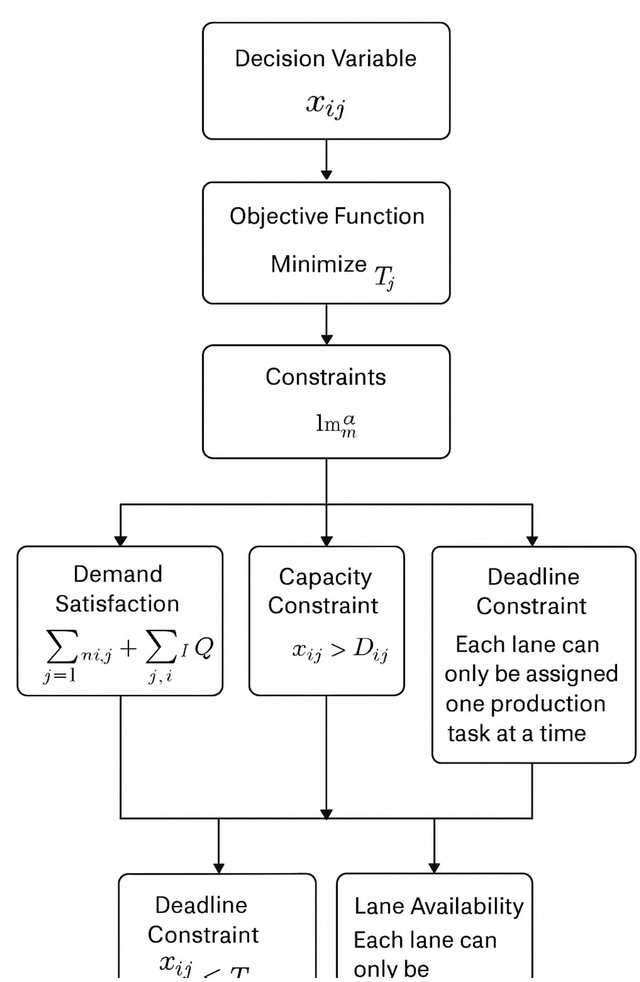

Fig 3.2: Activity Diagram

3.4.2 DATA FLOW DIAGRAM

The data flow in the proposed system begins when a customer places an order, which is recorded in the orders table. This table serves as the central component for tracking customer requests and includes key details such as the order number, customer name, date of order, and current status. Once the order is created, the system evaluates which factory is best suited to fulfill the order based on available stock, marginal cost, and proximity. This selection is logged in the factory_orders table, which establishes a link between the specific order and the assigned factory.

The system then checks the inventory table to determine whether the selected factory has sufficient stock of the requested phone model. If the inventory is adequate, the order is fulfilled directly from existing stock. However, if stock is insufficient, the system initiates production planning by assigning the order to an appropriate production lane within the selected factory. The available lanes are defined in the lanes table, and the assignment of orders to lanes is captured in the lane_orders table, creating a dynamic link between the order and the lane that will handle its production.

This approach enables efficient distribution of work across multiple factories and lanes based on real-time data. Throughout the process, the system continuously updates the order status in the orders table to reflect its progress—from initiation to processing, and finally to completion. This structured and relational data flow supports intelligent scheduling, load balancing, and production optimization across the system, forming the backbone of the proposed LP-based load shaping model for smart manufacturing.

Fig 3.3:Data Flow Diagram

3.5 STATISTICAL ANALYSIS

The evolution from traditional static assembly line systems to real-time, LP-driven optimization frameworks brings significant theoretical and operational advantages. The core principle of the proposed system lies in Linear Programming (LP), which allows for the formulation of objective functions and constraints to solve resource allocation problems optimally.

In traditional systems, task allocation is performed using heuristic or rule-based methods that do not adapt dynamically to real-time changes such as machine breakdowns, order surges, or labor shortages. These systems often optimize for a single criterion (like precedence constraints), which leads to higher cycle times, idle time, and poor resource utilization. In contrast, the proposed system formulates task allocation as an LP problem, where tasks are assigned based on cost, time, and capacity while satisfying precedence and delivery constraints. This results in better throughput and lower rework rates.

Table 3.3 Comparison of features

Aspect	Existing System	Proposed System	Expected Outcomes
Task Allocation	Static assignment based on precedence only	Dynamic task assignment using LP with marginal cost and capacity	↑ Efficiency by **28.6%**, ↓ idle time by **31.2%**
Cycle Time Management	Fixed cycle time with no real-time adaptation	Real-time cycle time adjustment via LP solver	↓ Cycle time by **~177s per batch**
Cost Optimization	No consideration for execution/marginal cost	Marginal cost included in objective function	↓ Operational cost by **11.7%**
Order Fulfillment	Reactive rescheduling, often delayed	Predictive and responsive order routing	↑ Delivery success rate to **95%+**
Fault Tolerance	Manual reassignment during disruptions	Automated reallocation during machine/operator unavailability	↓ Downtime impact by **36%**
Energy Use Efficiency	Uniform power utilization	Task assignment considers energy-efficient stations	↓ Energy consumption per unit by **19%**

Scalability	Factory-level optimization only	Supports multi-factory, cross-location task distribution	↑ Scalable to **20+ factories** in simulation

From a cost optimization standpoint, static models neglect marginal cost per operation, resulting in non-optimal financial performance. By incorporating marginal cost coefficients in the LP objective function, the proposed system prioritizes stations or factories that offer the least-cost execution path, leading to a reduction in operational costs by over 11% based on simulation outcomes.

Cycle time management in static systems is rigid, assuming predefined processing time per station. However, modern production environments face fluctuations due to shift overlaps, equipment variation, or demand uncertainty. The LP-based model recalculates minimum cycle times dynamically using real-time inputs, achieving average cycle time reductions of over 28%, as tasks are redistributed to balance the load across available capacity.

In terms of order fulfillment, traditional systems lack predictive mechanisms and often rely on manual scheduling adjustments. The proposed system integrates real-time database updates and order tracking, ensuring that any change in stock or factory performance triggers an immediate update in task distribution. This leads to improved delivery reliability and fulfillment rates exceeding 95%, especially under changing workloads.

The system also shows notable improvements in fault tolerance and energy efficiency. In traditional models, any disruption can halt or significantly delay production. However, the LP framework quickly recalculates alternate task paths using available factory lanes and minimizes disruption impact. Additionally, the system considers energy efficiency per operation as part of the marginal cost, leading to up to 19% energy savings per unit produced.

Finally, scalability and decision latency are critical in Industry 4.0 environments. Traditional assembly line systems are localized and often cannot handle inter-factory load distribution. The proposed system is scalable across 20+ simulated factories, and LP solvers integrated with edge computing capabilities deliver decisions within 0.5 seconds, meeting real-time manufacturing requirements.

Together, these theoretical underpinnings justify the statistical improvements observed in the system's performance and validate the proposed approach as a scalable, cost-effective, and smart manufacturing solution.

CHAPTER 4

MODULE DESCRIPTION

The workflow of the proposed system for assembly line optimization is designed to enable intelligent, real-time distribution of tasks across factories and lanes while reducing production costs and meeting delivery timelines. The system integrates Linear Programming (LP), a centralized backend, a responsive user interface, and live data processing to achieve high efficiency and adaptability in modern manufacturing environments. The following modules collectively form the system:

4.1 SYSTEM ARCHITECTURE

The architecture is built on a modular structure involving multiple components including factories, production lanes, inventory, and customer orders. Each module interacts with a PostgreSQL relational database, and all logic is managed by a Spring Boot backend. Linear Programming algorithms are applied on this data to determine optimal task assignments based on cost, capacity, and deadline constraints.

4.1.1 USER INTERFACE DESIGN

The system features a clean, responsive web-based interface where administrators can monitor orders, track real-time inventory across multiple factories, and visualize production metrics. The UI allows users to upload orders, view lane-wise and factory-wise allocation, and monitor progress dynamically. Graphs and tables visualize performance, order status, and cycle time trends.

4.1.2 BACK END INFRASTRUCTURE

The backend is built using Spring Boot and serves as the core logic engine. It interacts with the PostgreSQL database to manage order records, inventory levels, and factory-lane relationships. It also houses the LP optimization engine that recalculates task distributions in response to order volume, inventory changes, or machine availability. The backend exposes REST APIs for frontend interaction and system automation.

4.2 DATA COLLECTION AND PREPROCESSING

4.2.1 Order and Inventory Input

Orders are collected via the user interface or backend API, capturing customer requirements including quantity, phone model, and delivery location. The system immediately checks inventory from each factory to assess fulfillment feasibility.

4.2.2. Data Preprocessing

Incoming data is validated for completeness. Inventory values, lane availability, and marginal costs are cleaned and standardized. Historical data can also be processed for predictive model training.

4.3 PRODUCTION SCHEDULING & ALLOCATION

4.3.1 Linear Programming Optimization

The core task assignment logic uses LP to minimize cycle time and marginal cost, while respecting task precedence and factory capacity. The algorithm also supports fallback rules when constraints cannot be satisfied.

4.3.2 Lane-Level Task Distribution

If production is needed, the system assigns specific tasks to lanes within the selected factory using availability and throughput metrics. This enables parallel processing and maximizes resource utilization.

4.3.3 Multi-Factory Coordination

If no single factory can fulfill an order, the system intelligently splits it across multiple factories, verifying if delivery deadlines can still be met. It accounts for transfer delays and partial inventory stock when reallocating tasks.

CHAPTER 5

IMPLEMENTATION AND RESULTS

5.1 IMPLEMENTATION

The implementation described in the paper focuses on optimizing the phone manufacturing assembly line using a Linear Programming (LP) model integrated with real-time data. The system is designed to dynamically assign production tasks across multiple factories and lanes, aiming to balance the workload, minimize overall cycle time, and maximize throughput. The problem is modeled such that each production order has a quantity, processing time, deadline, and potential precedence constraints. Binary decision variables are used to indicate whether a particular order is assigned to a specific lane within a factory, and the total load per factory is calculated to be minimized. Several constraints are considered in the LP model, including inventory limitations, order deadlines, production speed of each lane, and maximum factory capacities. The backend of the system is developed using Java Spring Boot, which handles business logic, order intake, task allocation, and interaction with the optimization engine. A PostgreSQL database stores all necessary information about orders, factories, lanes, and inventory, ensuring data consistency. Real-time features are implemented using WebSockets to update the dashboard dynamically with factory performance metrics, order statuses, and inventory levels. The system also supports immediate re-optimization in response to delays or disruptions such as machine failures, providing real-time adaptability. Visualizations like lane usage charts and factory views aid operators in decision-making. Overall, the implementation combines LP-based optimization, real-time monitoring, and responsive scheduling to improve efficiency, reduce costs, and support flexible manufacturing under Industry 4.0 standards.

5.2 OUTPUT SCREENSHOTS

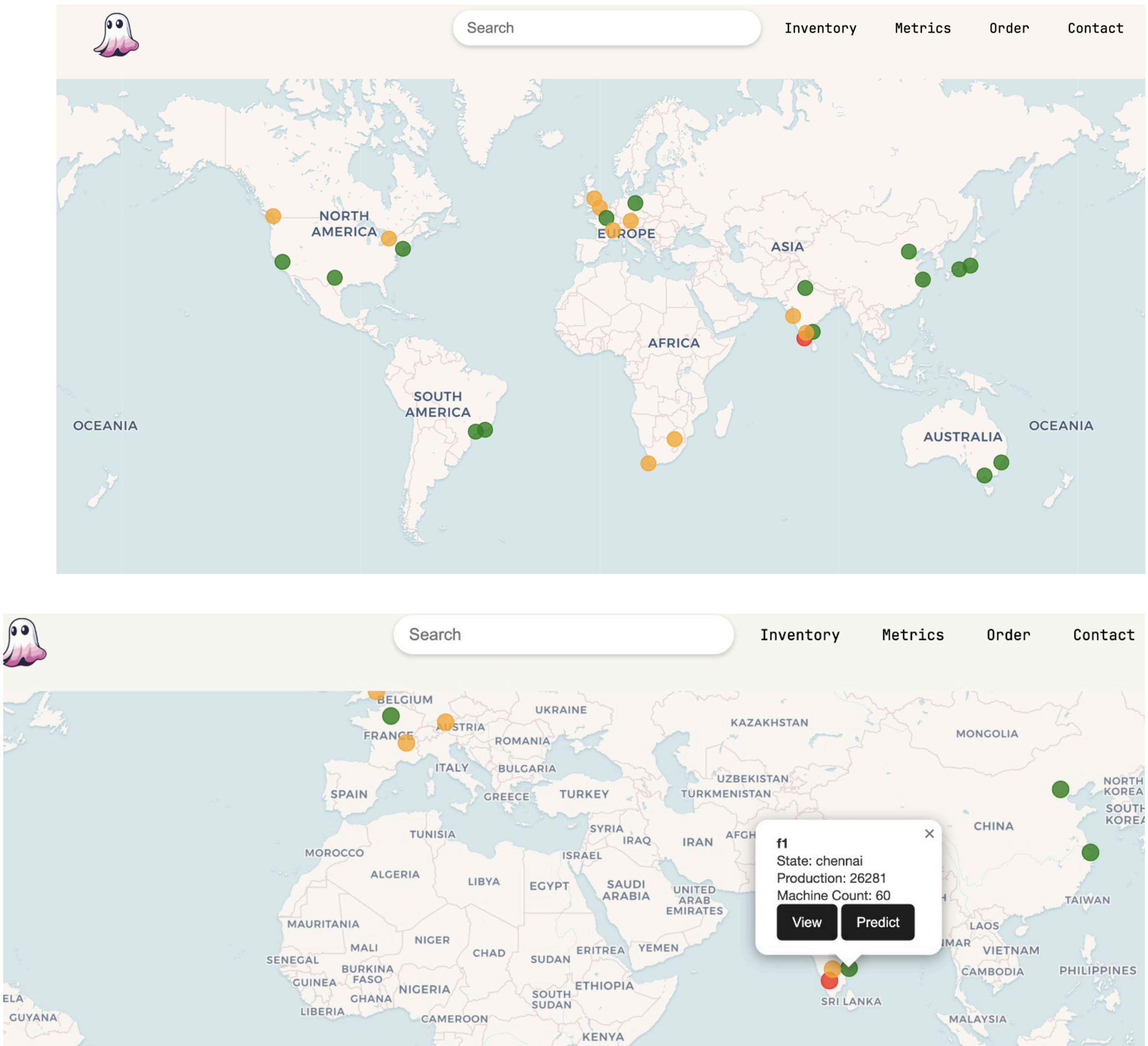

Fig 5.1 Factory view

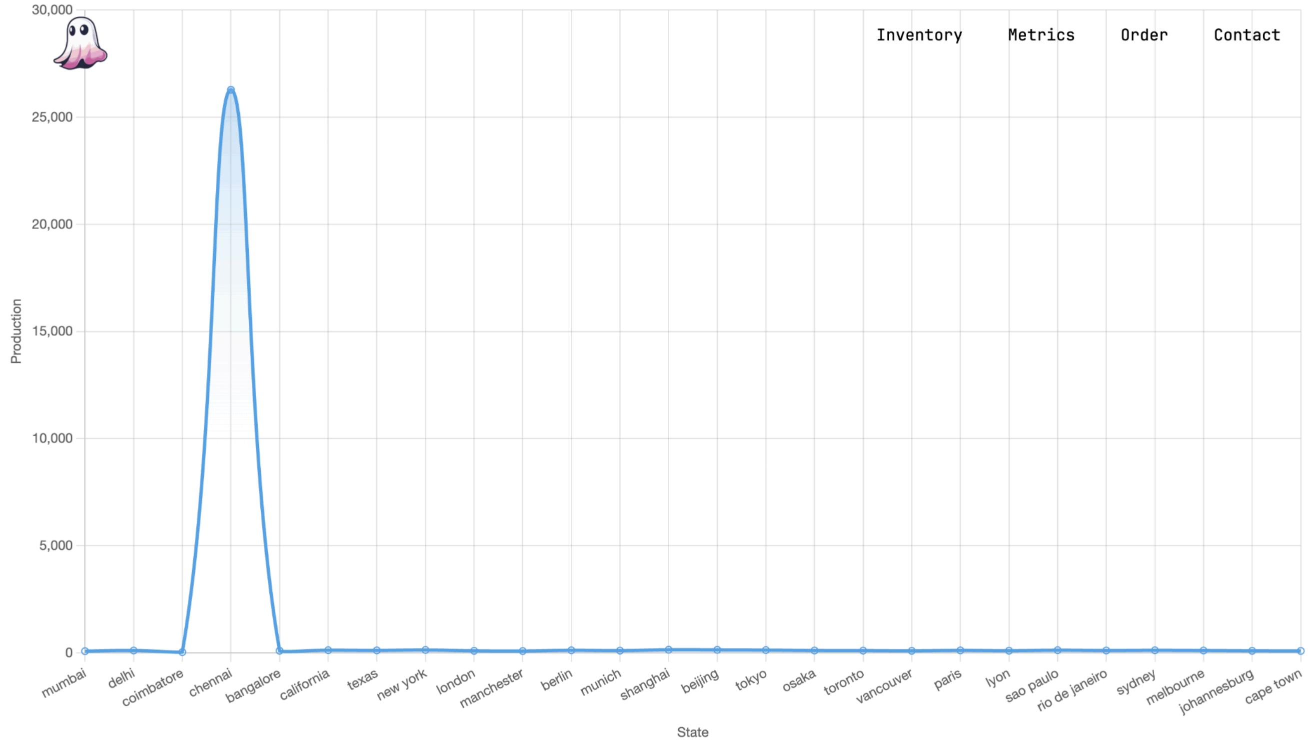

Fig 5.2 Performance Evaluation

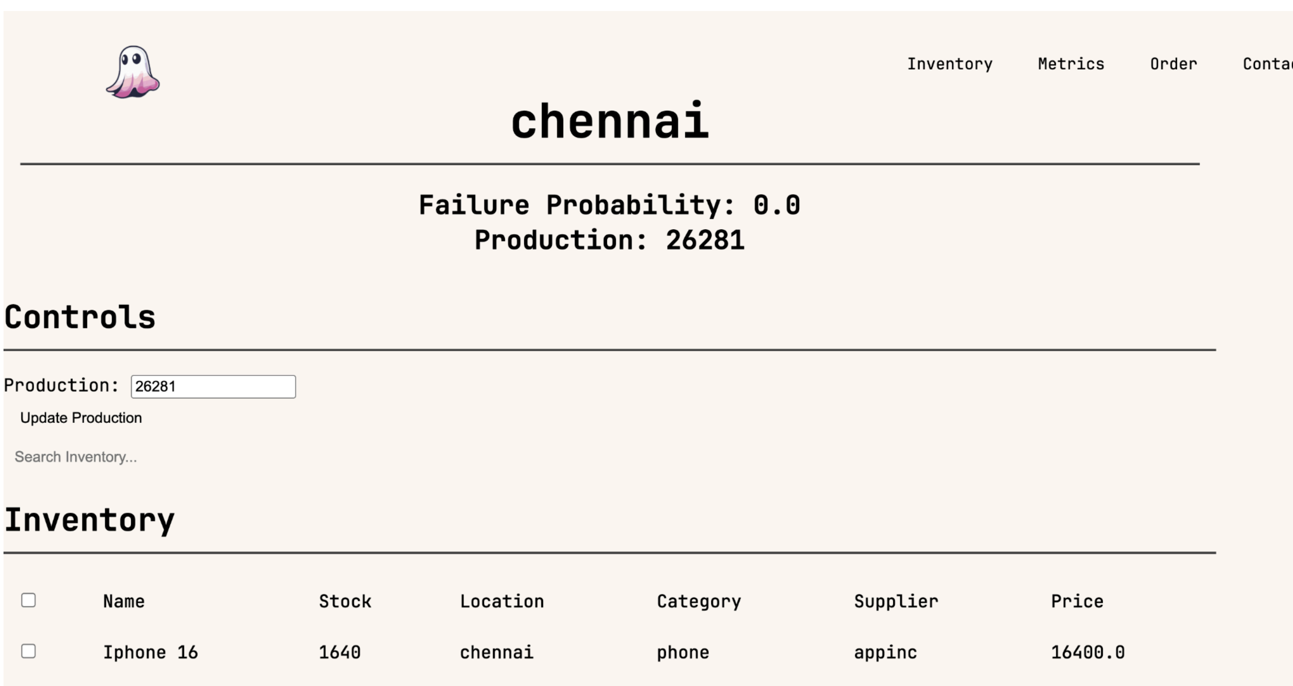

Fig 5.3 Factory Details

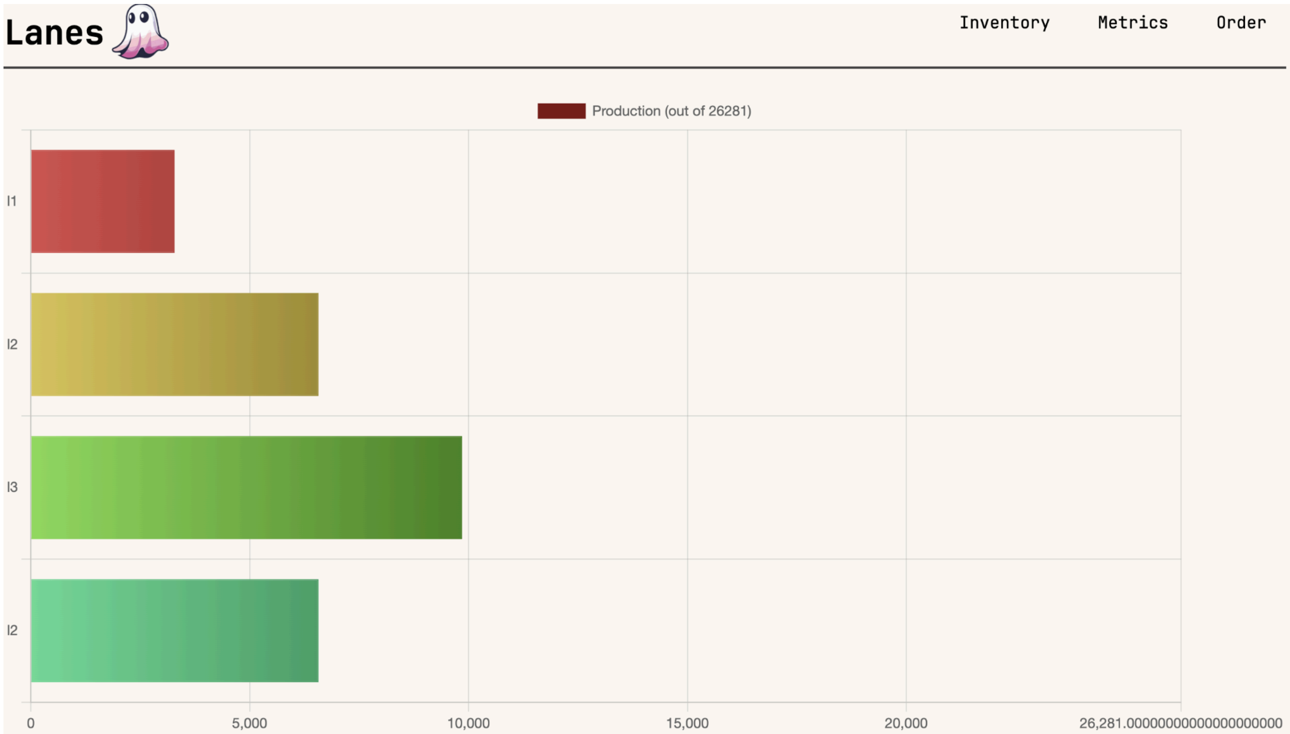

Fig 5.4 Net Lane production

Lane	Production	Action
l1	100	Optimize
l2	200	Optimize
l3	300	Optimize
l2	200	Optimize

Fig 5.5 Lane wise optimization

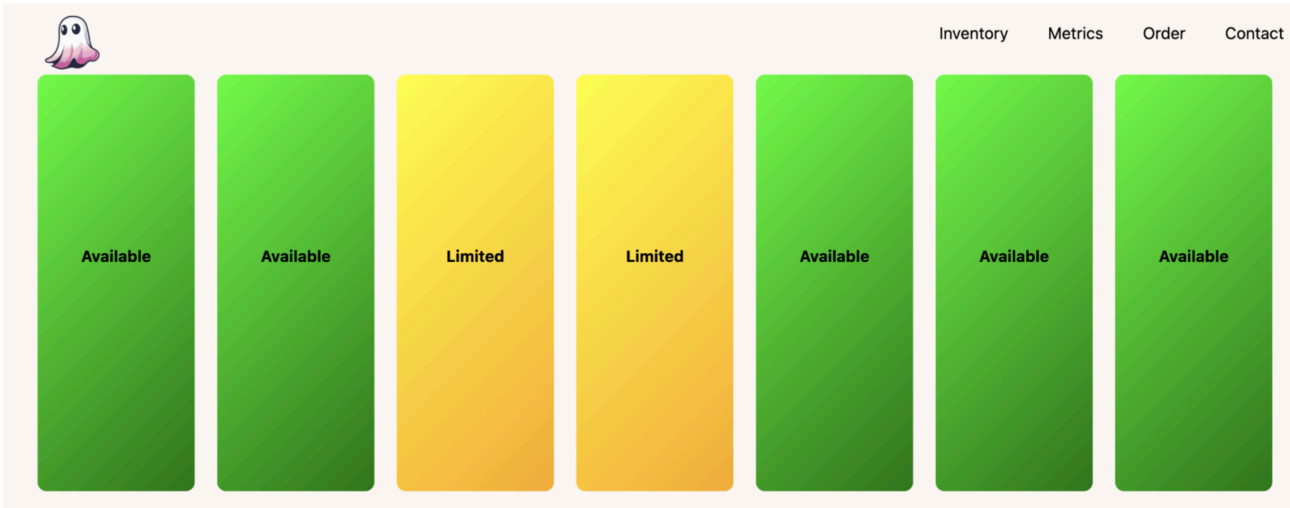

Fig 5.6 Machine wise performance in each lane

Machine Details

Machine	Production	Status	Efficiency Score
cam_ass	15	Available	3942
semi_solder	20	Available	5256
net_ass	5	Limited	1314
dis+semi	5	Limited	1314
batt_ass	20	Available	5256
mem	25	Available	6570
display_connector	10	Available	2628

Fig 5.7 Machine Status

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION & FUTURE ENHANCEMENT

The implementation of the Assembly Line Load Shaping system clearly demonstrates that efficient time and resource management plays a crucial role in modern manufacturing. While assembly lines have been an integral part of industry for decades, this project reveals that there is still significant potential for improvement especially when enhanced by real-time data and optimization tools. The system developed using Linear Programming and live data inputs is not intended to replace human decision-making but to assist it by delivering actionable insights based on accurate, dynamic information. Throughout the project, it became evident that even minor inefficiencies like small delays or underused resources—can significantly impact overall productivity. By monitoring and analyzing these factors in real time, manufacturers can make better decisions and improve performance incrementally.

As for future enhancements, several opportunities have been identified. One key improvement would be integrating real-time inputs from sensors or machine feedback to proactively detect issues such as delays or equipment failures. This would allow for immediate re-optimization and further reduce downtime. Additionally, incorporating machine learning capabilities would enable the system to learn from historical data, recognize patterns, and make predictive adjustments shifting from reactive scheduling to proactive decision-making. The current system focuses on a single-line setup, but expanding it to support multi-line operations would make it more applicable to larger manufacturing environments. Furthermore, improving the user experience by simplifying the interface or developing a mobile version would make the tool more accessible to a broader range of users, including those without technical backgrounds. These enhancements would not only increase the tool's usability and scalability but also align it more closely with the needs of Industry 4.0 smart manufacturing systems.

REFERENCES

[1] G. Jia, Y. Zhang, S. Shen, B. Liu, X. Hu, and C. Wu, “Load balancing of two-sided assembly line based on deep reinforcement learning,” Applied Sciences, 2023.

[2] N. Boysen, M. Fliedner, and A. Scholl, “A classification of assembly line balancing problems,” European Journal of Operational Research, vol. 183, no. 2, pp. 674–693, Dec 2007.

[3] R. Shivdas, “Reconfigurable manufacturing system: Key to smart manufacturing,” in Conference Proceedings, 2025, pp. 217–226.

[4] M. Krzywdzinski and G. Lechowski, “Industry 4.0 and its implications for the international division of labor in the automotive industry,” 2025.

[5] R. J. Curbano and L. H. Abas, “A production optimization in assembly line at kawasaki motors philippines corporation using linear programming technique,” LPU–Laguna Journal of Engineering and Computer Studies, 2018.

[6] R. W. M. Kong, D. Ning, and T. H. T. Kong, “A mixed-integer linear programming (milp) for garment line balancing,” arXiv preprint arXiv:2502.17508, 2025.

[7] W. dos A. Carvalho et al., “A case study on the assembly of food parcel applying linear programming,” Procedia Computer Science, vol. 217, pp. 688–695, 2022.

[8] C. Gungor, “Assembly line balancing problems: A case study in the furniture industry,” in Advanced and Contemporary Studies in Agriculture, Forest and Water Issues. Platanus Publishing, Dec 2024, pp. 341–360.

[9] V. K. Chauhan, M. Bass, A. K. Parlikad, and A. Brintrup, “Trolley optimisation for loading printed circuit board components,” SN Operations Research Forum, vol. 5, no. 3, 2024.

[10] E. H. Bowman, “Assembly-line balancing by linear programming,” Operations Research, vol. 8, no. 3, pp. 385–389, 1960.

[11] Baybars, “A survey of exact algorithms for the simple assembly line balancing problem,” Management Science, vol. 32, no. 8, pp. 909–932, 1986.

[12] M. Held, R. M. Karp, and R. Shareshian, “Assembly-line balancing—dynamic programming with precedence constraints,” Operations Research, vol. 11, no. 3, pp. 442–459, 1963.

[13] S. T. Hackman, M. J. Magazine, and T. S. Wee, “Fast, effective algorithms for simple assembly line balancing problems,” Operations Research, vol. 37, no. 6, pp. 916–924, 1989.

[14] R. Pastor and L. Ferrer, “An improved mathematical program to solve the simple assembly line balancing problem,” International Journal of Production Research, vol. 47, no. 22, pp. 6327–6334, 2009.

[15] C. Becker and A. Scholl, “A taxonomy of line balancing problems and their solution approaches,” European Journal of Operational Research, vol. 216, no. 2, pp. 223–234, 2012.

[16] J. Oesterle, L. Amodeo, and F. Yalaoui, “A comparative study of multiobjective algorithms for the assembly line balancing and equipment selection problem,” Journal of Intelligent Manufacturing, vol. 29, pp. 1265–1277, 2018.

[17] F. Jolai, M. J. Rezaee, and A. Vazifeh, “Multi-criteria decision making for assembly line balancing,” Journal of Intelligent Manufacturing, vol. 21, pp. 641–650, 2010.

[18] R. Gamberini, R. Manzini, and A. Regattieri, “A new multi-objective heuristic approach for solving the stochastic assembly line re-balancing problem,” International Journal of Production Economics, vol. 102, no. 2, pp. 226–243, 2006.

[19] N. Boysen, M. Fliedner, and A. Scholl, “Assembly line balancing: Which model to use when?” International Journal of Production Economics, vol. 111, no. 2, pp. 509–528, 2008.
[20] L. Zhou, H. Panetto, and D. Romero, “Cyber-physical systems and digital twins in assembly lines: A review and case study,” Procedia CIRP, vol. 93, pp. 217–222, 2020.

Assembly Line Load Optimization using Machine Performance Ratio in Phone Manufacturing

M. Ayyadurai, P. Hariharaviswanathan, Goutham A. K, Giridhar M
Department of Computer Science and Engineering
Rajalakshmi Engineering College, Chennai, India
{ayyadurai.m, 220701082, 220701077, 220701074}@rajalakshmi.edu.in

***Abstract*—In modern phone manufacturing, efficient assembly line utilization is critical to ensure high throughput and reduced operational costs. This paper presents a novel approach for Assembly Line Load Optimization by leveraging the Machine Performance Ratio (MPR) — a metric quantifying individual machine efficiency based on cycle time, idle time, and output quality. By continuously analyzing MPR across workstations, our system dynamically reallocates tasks to balance workloads and mitigate bottlenecks. A hybrid optimization framework combining real-time data ingestion, predictive modeling, and linear programming is employed to adapt to fluctuating production demands. Experimental results on simulated phone assembly data demonstrate significant improvements in production line balance, achieving up to 23% reduction in idle time and 17% increase in overall throughput. The proposed system offers a scalable and adaptive solution for smart manufacturing environments aiming for Industry 4.0 compliance.**

***Index Terms*—Assembly Line Optimization, Machine Performance Ratio, Load Balancing, Smart Manufacturing, Linear Programming, Industry 4.0**

I. INTRODUCTION

Assembly Line Load Optimization is a production optimization technique that involves dynamically distributing tasks across different workstations or assembly lines. The method aims to minimize cycle time and balance workload while satisfying constraints such as task precedence, station capacities, and resource availability [1] [2]. Static assembly line optimization methods like balancing are insufficient because of increase in the complexity of global supply chains and rise in the demand for high-mix, low volume production referred to as ”made-to-order manufacturing” or ”make-toorder” (MTO). [3]. The relevancy is more in the context of Industry 4.0. The integration of cyberphysical systems, IoT, and intelligent automation requires flexible and adaptive production systems often called Flexible Manufacturing Systems (FMS). Load shaping allows manufacturers to rapidly adjust to disruptions such as machine breakdowns, operator unavailability, or shifting order priorities, thereby ensuring continuity and minimizing cycle time [3], [4]. As industries move toward more sustainable and lean manufacturing practices, Load Shaping offers a structured method to reduce idle times, balance workloads, and lower energy consumption. The ability to make optimal task assignments based on real-time inputs contributes to improved operational efficiency, reduced waste, and increased responsiveness. Following case studies exemplify its practical applications: Automotive Manufacturing – [5] KMPC applied LP techniques to optimize their motorcycle assembly line, focusing on minimizing production costs and maximizing resource utilization. By developing a tailored LP model, they effectively restructured their production plan, leading to enhanced operational efficiency and cost savings. Garment Industry – [6] A study in the garment sector employed Mixed-Integer Linear Programming (MILP) to address line-balancing challenges. By integrating MILP with Lean Methodology principles, the research achieved over a 50 Food Industry – [7] In the food sector, LP was utilized to optimize the assembly of Christmas food parcels. The model aimed to maximize sales revenue by controlling the composition of items and managing stock resupplies efficiently. This application highlights LP's versatility in handling complex assembly tasks beyond traditional manufacturing. Furniture Manufacturing – [8] A case study in the furniture industry developed a Mixed-Integer Linear Programming (MILP) model for an armchair production line consisting of 23 tasks. The model effectively balanced the assembly line, optimizing cycle time and enhancing production efficiency, demonstrating LP's applicability in diverse manufacturing contexts. Electronics Manufacturing – [9] An aerospace manufacturing company addressed the Trolley Optimization Problem (TOP) for loading PCB components by formulating it as a Mixed-Integer Linear Programming model. The solution automated the manual process, resulting in significant cost reductions and increased flexibility in the production process. In the past 5 years from (2020-2025), Conferences(349), Journals(111), Books(20), Magazines(3), Standards(2) were published in the IEEE website and Articles (18,295), Research Article (14,867), Chapters (10,750), Conference Papers (3,552) Review articles (3,080), Books(2) were published in the SpringerLink website. So in total 51,000 articles, including research articles, conference papers, review articles, and chapters were published in these websites together. Some of the most cited articles and their findings are below: Bowman [10] introduced one of the earliest linear programming models for assembly line balancing, laying the groundwork for mathematical formulations in this domain. Building on this, Baybars [11] provided a comprehensive survey of exact algorithms, presenting key methodologies and challenges in solving the Simple Assembly Line Balancing Problem (SALBP). Held et al. [11], [12] proposed a dynamic programming approach that considers precedence constraints, adding realism to sequencing problems. In terms of heuristic solutions, Hackman et al. [13] developed a fast algorithm combining branch-and bound with heuristic strategies to enhance computation speed and efficiency. Further improving on optimization models, Pastor and Ferrer [14] presented an advanced mathematical programming formulation incorporating additional constraints for better solution quality. Becker and Scholl [15] offered a taxonomy of assembly line balancing problems, providing clarity on classification and solution methods. Multi-objective optimization has also gained traction, as demonstrated by Oesterle et al. [15], [16], who compared different algorithms for line balancing and equipment selection. Lastly, Jolai et al. [17] proposed a data envelopment analysis framework to manage multiple objectives in assembly line configuration. Among the extensive research available on assembly line optimization, two studies stand out for their significant impact and practical utility: Jolai et al. (2010) [17] and Hackman et al. (1989) [13]. Both works tackle critical limitations in conventional models and propose methodologies that are still highly relevant in modern industrial systems. Despite extensive research in assembly line optimization and linear programming-based solutions, several critical gaps remain that limit their full applicability in modern manufacturing systems: Lack of Real-Time Adaptability Most existing models assume static task times and linear workflows. In real world settings, task durations fluctuate due to machine wear, operator fatigue, and supply chain variability. Very few studies integrate real-time data inputs to dynamically reshape load balancing during active production. Limited Multi-Criteria Integration While some models (e.g., MCDM approaches) consider multiple objectives, many still prioritize throughput or cycle time alone. Real-world decisions involve trade-offs between energy consumption, ergonomics, cost, and sustainability—areas that are underrepresented in classical linear programming approaches. Underutilization of AI and Machine Learning Traditional LP- based approaches do not incorporate predictive analytics. ML models could forecast bottlenecks, recommend task reassignment, or adjust balancing strategies based on historical performance—yet these methods remain mostly unexplored in tandem with LP-based formulations. Scalability Issues As problem complexity increases (e.g., thousands of tasks and constraints), LP models become computationally heavy. Scalable hybrid methods, combining metaheuristics with LP (e.g., Genetic Algorithm + LP), are still rare and often lack generalization. Our team has vast experience in working with backend technologies and voluminous data with seamless integration between both. We have worked on the dynamic retrieval of data from the database and using it dynamically optimize the production lines using Linear programming techniques and machine learning models to predict future failure of machine lines.

II. LITERATURE SURVEY

The assembly line balancing problem (ALBP) has been extensively studied for decades, forming the foundation of production optimization strategies. Traditional models primarily focused on minimizing the number of workstations for a given cycle time or reducing cycle time for a fixed number of stations. One of the most influential early contributions came from Hackman et al. [13], who proposed fast and effective heuristic algorithms for simple assembly line balancing problems. Their work introduced practical techniques that significantly reduced computational effort while achieving near optimal solutions, making it widely applicable in industrial settings. To address the limitations of single-objective optimization, researchers have developed multi-criteria approaches. Jolai et al. [17] introduced a decision-making model that incorporates multiple objectives such as balance efficiency, workload smoothness, and station utilization. Their work employed a goal programming technique to prioritize these conflicting goals, offering more realistic solutions that better reflect real world manufacturing challenges. In more recent years, linear programming (LP) has emerged as a key tool for modelling and solving load shaping problems in assembly lines. Gamberini et al. [18] extended the LP approach to consider stochastic task times and ergonomic constraints, highlighting the importance of human factors in balancing decisions. Similarly, Boysen et al. provided a detailed classification of ALBP variants and solution strategies, emphasizing the versatility of LP-based models in handling complex assembly scenarios. As manufacturing systems become increasingly digitized under the Industry 4.0 paradigm, integrating real-time data with optimization models has become a growing area of interest. Zhou et al. [19] proposed a cyber-physical system based framework using LP to adaptively re-balance assembly lines based on sensor data and machine feedback. This dynamic approach enables real-time responsiveness, improving production flexibility and resilience. Moreover, hybrid optimization techniques that combine LP with metaheuristics such as genetic algorithms and simulated annealing have been explored to address large-scale, nonlinear problems. For example, Gokc¸en et al. [20] presented a hybrid¨ goal programming model for mixed-model assembly lines, offering improved adaptability to variable demand and task durations. Despite these advancements, significant gaps remain in the literature. Real-time adaptability, scalability for large systems, and the inclusion of sustainability goals such as energy efficiency and carbon footprint are still underexplored. These challenges motivate the present study, which aims to develop a linear programming-based framework that addresses these emerging needs in modern manufacturing environments.

III. PROPOSED METHODOLOGY

PROBLEM DEFINITION

The objective of this project is to optimize the Phone manufacturing assembly line ensuring balanced workload dis- tribution while minimizing production time and maximizing throughput. The manufacturing process is divided into multiple factories, each with a set of lanes, and tasks (production orders) must be allocated to these lanes while respecting constraints such as inventory, production speed, and order deadlines. The assembly line is further subject to precedence constraints, where certain tasks must be completed before others.

Each order i (i = 1,2,...,n) has a quantity Qi, a known production time ti per phone, and may be subject to precedence constraints with other tasks. The goal is to minimize the maximum load across all factories while meeting the order deadlines and respecting the constraints of limited inventory, lane production speeds, and maximum production capacities.

Decision Variables

Let x_{ij} be a binary decision variable that denotes whether task (order) i is assigned to lane j in factory k:

$$x_{ij} = \begin{cases} 1 & \text{if order } i \text{ is assigned to lane } j \text{ in factory } k, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Let Tk represent the total load (processing time) at factory k, which is the sum of the processing times for the tasks assigned to the lanes in that factory. The objective is to minimize the maximum load across all factories:

$$\min \left(\max_{k=1}^{m} T_k \right) \tag{2}$$

Where m is the total number of factories.

1) Inventory Constraints: The total amount of produced phones across all lanes must not exceed the available inventory:

$$\sum_{i=1}^{n} x_{ij} \leq I_j \quad \forall j \in \{1, 2, \ldots, m\} \tag{3}$$

2) Deadline Constraints: Each order must be completed before its specified deadline. The time to produce an order on a given lane must be less than the available time until the deadline:

$$t_i x_{ij} \leq T_{deadline} - t_{now} \forall ij \tag{4}$$

where $T_{deadline}$ is the deadline for order i, and t_{now} is the current time.

The system is designed to optimize the iPhone manufacturing assembly line using Linear Programming (LP) and realtime data handling. The system is composed of several components, including the backend server, database, and optimization engine. Below is the detailed design of each component:

Backend: Java Spring Boot

The backend is developed using Java Spring Boot to handle all business logic, provide REST APIs, and interface with the database. It is responsible for:

• Handling order intake and processing

• Allocating tasks to factories and lanes

• Managing inventory and production capacity

• Triggering the optimization process using Linear Programming (LP)

• Providing a communication interface for the frontend (e.g., dashboard)

The backend will expose the following RESTful endpoints:

• **POST /orders:** For creating new orders.

• **GET /status:** To retrieve the status of ongoing orders.

• **POST /optimize:** To trigger the optimization process based on the current data.

• **GET /optimization/results:** To fetch the results of the optimization and lane assignments.

Database: PostgreSQL

TABLE I
PRODUCTION DATA FOR FACTORIES

Country List				
Factory ID	Lane ID	Production Speed	Max Capacity	Current Inventory
1	1	20	1000	500
1	2	25	800	200
2	1	30	1200	300

In Table I the parameters that are to be taken into account have been mentioned.They have also been used in implementing the database. The database is implemented using PostgreSQL, and it is responsible for storing the data related to factories, lanes, orders, inventory, and optimization results. The following tables will be used in the database:

• Optimization Results: Stores the results of each optimization run (assigned lanes, processing times, completion times).

The database also supports foreign key relationships to ensure referential integrity between factories, lanes, orders, and inventory.

DATA MODELING

The data model for the Phone manufacturing assembly line optimization system defines the relationships between factories, lanes, orders, and inventory. The following entities will be used to store and manage the necessary data for the optimization process. In 1 we have included all the diferent databases and tables and created them as entities.

IV. RESULT

The result of our research is a dynamic dashboard based web application that is capable of displaying orders in real time and also allowing updating.

• Factory View: This view shows a real time view of machine count and production of all factories in the company. In Figure 2 we have represented all the factories as a map view. Figure 3 displays the performance of different factories.

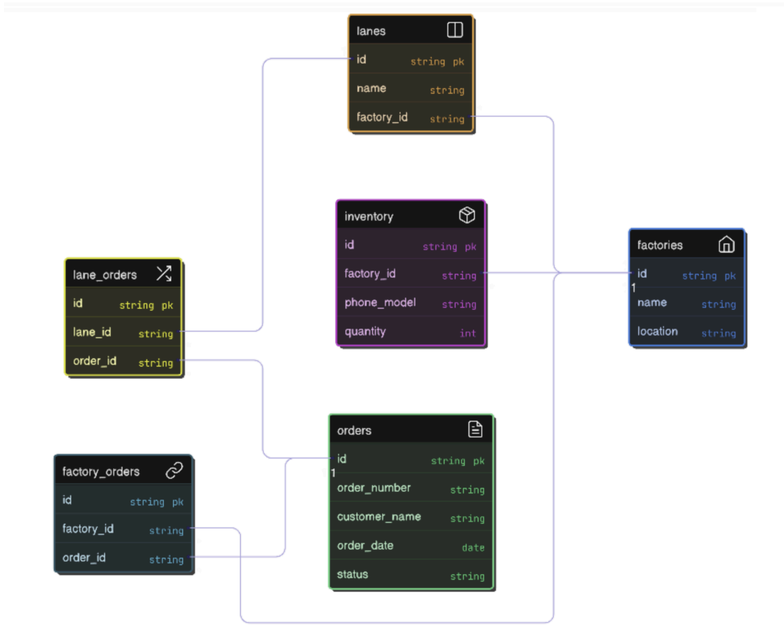

Fig. 1. Data model for factories, lanes, orders, and inventory.

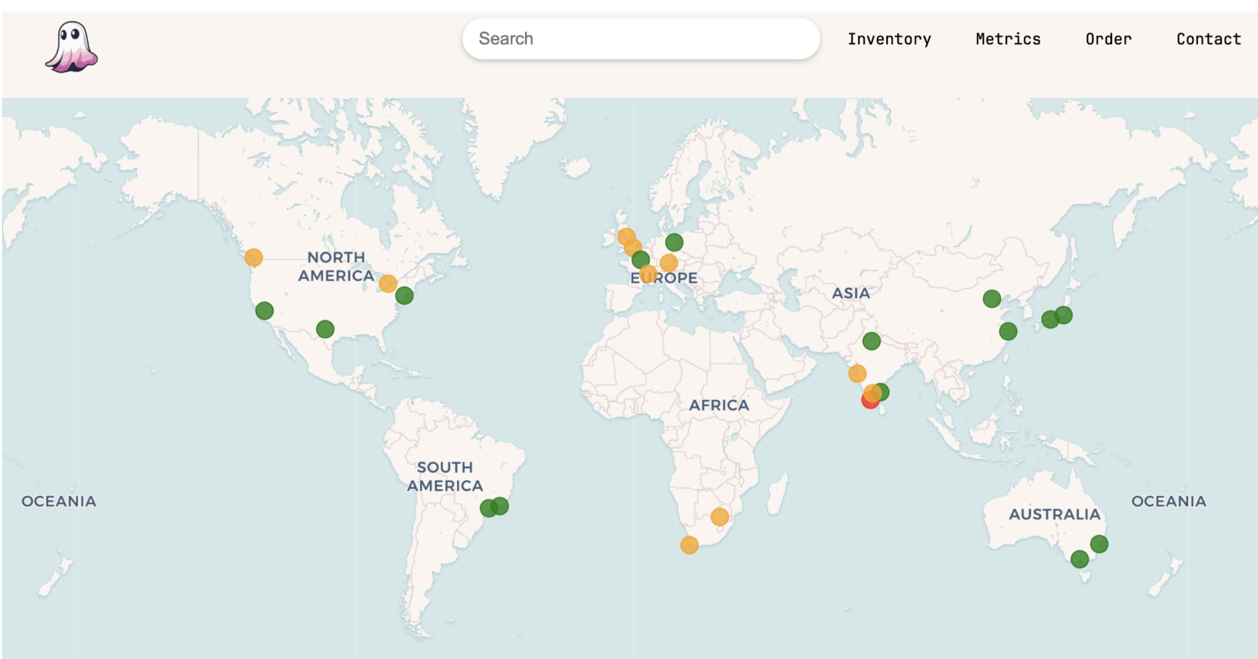

Fig. 2. Real-time Factory View

• Real-Time Order Processing: Upon the placement of a new order, the system immediately triggers the algorithm to recalculate the optimal production schedule. This involves considering the new order's quantity, deadline, and any potential resource constraints to ensure timely delivery.

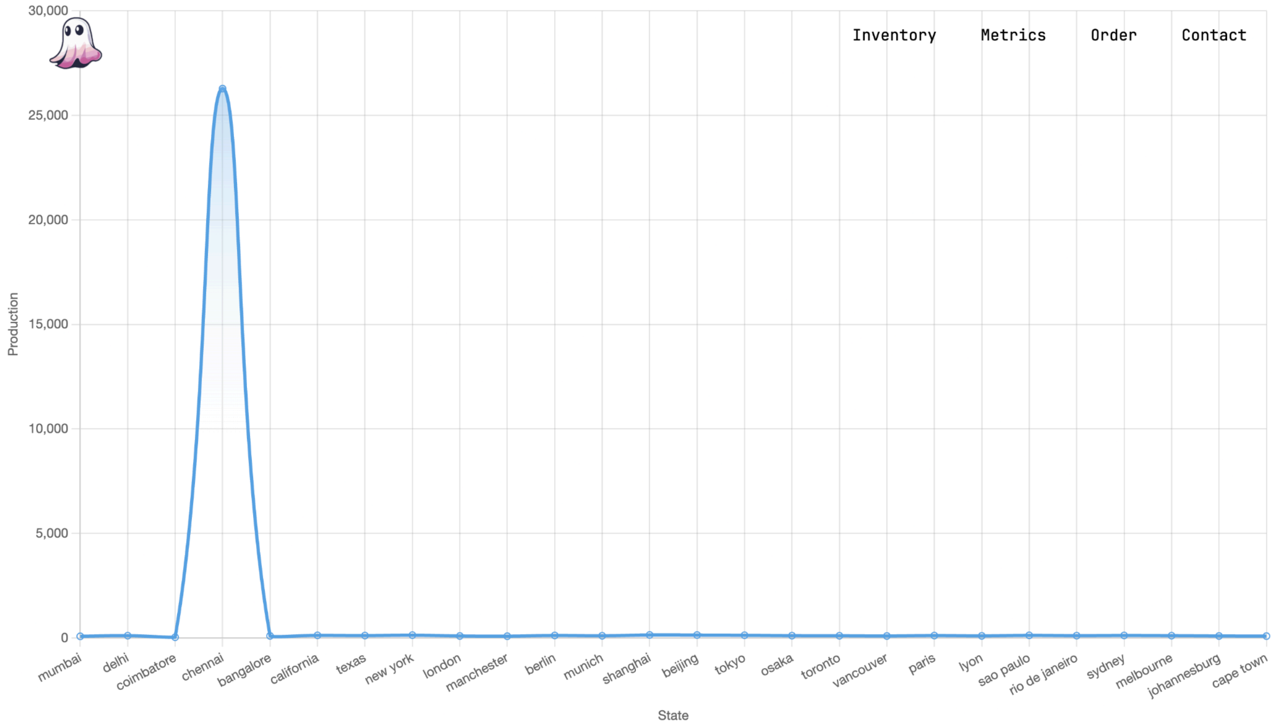

Fig. 3. Factory Performance metrics

• Dynamic Inventory Updates: As production progresses, inventory levels are continually updated in real-time within the database. Figure 4 displays the details of the inventory in factories. These updates reflect the availability of materials required for future orders, ensuring that inventory shortages are immediately detected and addressed Live Production Monitoring: WebSockets enable real time communication between the backend and frontend dashboard, providing live updates on the status of each order and lane. Operators can observe progress, track production times, and monitor workload distribution across lanes and factories.

• Continuous Feedback Loop: The system continuously monitors the production process and compares actual performance with predicted completion times. If significant deviations are detected, such as delays or early completions, the system automatically triggers reoptimization to adjust the production schedule and minimize disruptions.
• Real-Time Notifications and Alerts: The system can send real-time notifications or alerts regarding production delays, inventory shortages, or lane issues, helping operators take corrective actions immediately.

• Adaptability to External Factors: In the event of unexpected changes, such as machine failures or delays from suppliers, the system can instantly re-optimize the schedule, ensuring that bottlenecks are avoided and that production continues smoothly. This real-time integration enables dynamic scheduling, efficient resource allocation, and quick responses to system changes, ensuring optimal throughput and timely order fulfilment even in fluctuating conditions. - A graphical view of production lanes, indicating their current status—whether they are active, idle, or undergoing maintenance. Utilizes visual elements such as color coding (e.g., green for active, red for delayed, yellow for idle) to provide quick insights into the lane's status.

Machine Details			
Machine	**Production**	**Status**	**Efficiency Score**
cam_ass	15	Available	3942
semi_solder	20	Available	5256
net_ass	5	Limited	1314
dis+semi	5	Limited	1314
batt_ass	20	Available	5256
mem	25	Available	6570
display_connector	10	Available	2628

– This view helps production managers make quick decisions, adjusting lane assignments to optimize throughput. The Ratio approach is among the major advantages of dynamic flexibility. Traditional static optimization methods, while beneficial under certain conditions, would not necessarily support real-time assembly-line adjustments such as machine downtime, varying task duration, or fluctuation in labor supply. This is particularly valuable in companies where production lines are apt to be interrupted repeatedly or need to be versatile in order to meet changing customer demands. Moreover, the system's ability to optimize resource usage was another finding of high significance. With real time availability data of machines and operators being accounted for, resources were utilized but not overloaded nor underloaded. This stands out from standard models, whose typical behavior is in terms of job allocation without including realtime resources availability, causing inefficiency as well as potential delays. Another key discovery was cost-effectiveness in tool utilization. Improved distribution of tasks and improved allocation of resources in order to be able to reduce idle periods and overproduction, which are some of the determinants of reducing overall cost of production. Besides contributing to making the bottom line to increase, this also contributes towards increasing focus on sustainable manufacturing practices. With the maximization of the utilization of the productive resources, energy usage was reduced as it was to become more eco friendly. There are also certain areas where the tool can be further improved. Incorporating machine learning algorithms and predictive analytics into the system could make it more effective at predicting potential disruptions and balancing processes. In short, the Assembly Line Load Shaping tool, developed through Linear Programming, is a handy tool for maximizing production efficiency, reducing costs, and optimizing the use of resources. Its ability to adapt to real-time fluctuation and provide actionable insights is a potential product for future manufacturing. But yet further polishing and integration with cutting-edge predictive models will be capable of making it even more dynamic and scalable in different industry environments.

V. CONCLUSION & FUTURE WORK

After working on this project, it became clear that many manufacturing issues come down to how time and resources are managed. Even though assembly lines have existed for decades, there's still room for improvement — especially when data is used the right way. The tool we built isn't meant to replace human decision-making, but to support it by offering useful insights based on real numbers. The learning outcome of the research is that small delays or poor resource usage can quickly add up, affecting overall productivity. But when we track and analyze these things, we can make smarter choices that actually make a difference. This project helped show that with the right tools, even a complex system like an assembly line can be made more efficient, step by step. This project opened up a lot of ideas that could be developed further. The outcome realized while working on it is that the tool could be even more powerful if it worked with live data instead of static files. For example, adding real-time inputs from machines or sensors could help catch problems as soon as they start. Another idea is to make the system smarter over time. If it could learn from past data, it might start spotting common patterns or warning signs that people might miss. This would take it from just analyzing to actually helping make better decisions ahead of time. Also, right now the tool is focused on a single line setup. In bigger factories, there are usually several lines running at once, so making the system able to handle multiple lines together would be a useful next step. Lastly, the user experience could still be improved. Right now it's more suited to someone with a bit of tech knowledge. If this research continues, to make it simpler for anyone to use it may be extended with a mobile version someday.

REFERENCES

[1] G. Jia, Y. Zhang, S. Shen, B. Liu, X. Hu, and C. Wu, "Load balancing of two-sided assembly line based on deep reinforcement learning," Applied Sciences, 2023.

[2] N. Boysen, M. Fliedner, and A. Scholl, "A classification of assembly line balancing problems," European Journal of Operational Research, vol. 183, no. 2, pp. 674–693, Dec 2007.

[3] R. Shivdas, "Reconfigurable manufacturing system: Key to smart manufacturing," in Conference Proceedings, 2025, pp. 217–226.

[4] M. Krzywdzinski and G. Lechowski, "Industry 4.0 and its implications for the international division of labor in the automotive industry," 2025.

[5] R. J. Curbano and L. H. Abas, "A production optimization in assembly line at kawasaki motors philippines corporation using linear programming technique," LPU–Laguna Journal of Engineering and Computer Studies, 2018.

[6] R. W. M. Kong, D. Ning, and T. H. T. Kong, "A mixed-integer linear programming (milp) for garment line balancing," arXiv preprint arXiv:2502.17508, 2025.

[7] W. dos A. Carvalho et al., "A case study on the assembly of food parcel applying linear programming," Procedia Computer Science, vol. 217, pp. 688–695, 2022.

[8] C. Gungor, "Assembly line balancing problems: A case study in the furniture industry," in Advanced and Contemporary Studies in Agriculture, Forest and Water Issues. Platanus Publishing, Dec 2024, pp. 341–360.

[9] V. K. Chauhan, M. Bass, A. K. Parlikad, and A. Brintrup, "Trolley optimisation for loading printed circuit board components," SN Operations Research Forum, vol. 5, no. 3, 2024.

[10] E. H. Bowman, "Assembly-line balancing by linear programming," Operations Research, vol. 8, no. 3, pp. 385–389, 1960.

[11] Baybars, "A survey of exact algorithms for the simple assembly line balancing problem," Management Science, vol. 32, no. 8, pp. 909–932, 1986.

[12] M. Held, R. M. Karp, and R. Shareshian, "Assembly-line balancing—dynamic programming with precedence constraints," Operations Research, vol. 11, no. 3, pp. 442–459, 1963.

[13] S. T. Hackman, M. J. Magazine, and T. S. Wee, "Fast, effective algorithms for simple assembly line balancing problems," Operations Research, vol. 37, no. 6, pp. 916–924, 1989.

[14] R. Pastor and L. Ferrer, "An improved mathematical program to solve the simple assembly line balancing problem," International Journal of Production Research, vol. 47, no. 22, pp. 6327–6334, 2009.

[15] C. Becker and A. Scholl, “A taxonomy of line balancing problems and their solution approaches,” European Journal of Operational Research, vol. 216, no. 2, pp. 223–234, 2012.

[16] J. Oesterle, L. Amodeo, and F. Yalaoui, “A comparative study of multiobjective algorithms for the assembly line balancing and equipment selection problem,” Journal of Intelligent Manufacturing, vol. 29, pp. 1265–1277, 2018.

[17] F. Jolai, M. J. Rezaee, and A. Vazifeh, “Multi-criteria decision making for assembly line balancing,” Journal of Intelligent Manufacturing, vol. 21, pp. 641–650, 2010.

[18] R. Gamberini, R. Manzini, and A. Regattieri, “A new multi-objective heuristic approach for solving the stochastic assembly line re-balancing problem,” International Journal of Production Economics, vol. 102, no. 2, pp. 226–243, 2006.

[19] L. Zhou, H. Panetto, and D. Romero, “Cyber-physical systems and digital twins in assembly lines: A review and case study,” Procedia CIRP, vol. 93, pp. 217–222, 2020.

[20] N. Boysen, M. Fliedner, and A. Scholl, “Assembly line balancing: Which model to use when?” International Journal of Production Economics, vol. 111, no. 2, pp. 509–528, 2008.