1st World Congress on Industrial Engineering and Operations Management
, October 9-11, 2024
Publisher: IEOM Society International, USA DOI: 10.46254/WC01.20240161
Published: October 09, 2024

Optimal Manufacturing Load Balancing in High-Mix Production Lines by Leader-Follower Hierarchical Game Decision Making

Mulang Song
Georgia Institute of Technology, Atlanta, USA
Msong84@gatech.edu

Jianxin (Roger) Jiao
Georgia Institute of Technology,
Atlanta, USA

Abstract

In manufacturing environments with a mix of upgraded and new production lines, balancing manufacturing loads becomes complex due to varying levels of digitalization and multiple evaluation criteria. This paper introduces a leader-follower hierarchical game decision-making model to optimize manufacturing load balancing in high-mix production lines. A bi-agent genetic algorithm (BAGA) is developed, employing a Stackelberg game framework where the leader agent focuses on high-level decisions and the follower agent handles low-level scheduling with engineering constraints. Max-plus algebra is utilized to estimate task completion times for conventional lines. A case study demonstrates the effectiveness of the proposed approach.

Keywords
Genetic Algorithm, Manufacturing Load Balancing, Game Theoretical Optimization.

1. Introduction
Ever since Industry 4.0 emerged, manufacturing industries have relentlessly adopted advanced technologies across various applications, resulting in a manufacturing environment that is increasingly dynamic and complex. During the digital transformation era, manufacturers face the challenge of the high costs associated with continuously adopting new production lines equipped with advanced technologies. These cutting-edge technologies require not only significant financial investment but also enhanced resources and sophisticated management strategies. As a result, manufacturers may not fully utilize the production capabilities of these new lines. To balance cost constraints and operational efficiency, many manufacturers employ a dual approach: they invest in new production lines while simultaneously upgrading existing ones by integrating new sensors or inspection cameras. This strategy leads to a production environment where products are manufactured using a mix of both upgraded and newly installed production lines.

In today's market, where mass customization and personalization are becoming more common, manufacturers are facing orders that are more complicated than ever before. Because of these changes, scheduling production tasks becomes increasingly difficult, as the usual methods might not work well with the new demands. Manufacturers are finding it harder to make decisions about which orders to accept and how to schedule the tasks needed to complete them. The challenge of balancing customer expectations with efficient production makes the process of accepting orders and scheduling tasks much more complex than it used to be.

Managing production environments that utilize production lines of varying ages with the complexities of order scheduling presents several significant challenges. First, constructing a decision-making model for manufacturing load balancing becomes complicated when multiple evaluation criteria are involved. Second, developing a solution algorithm capable of effectively solving this model is essential but not straightforward. Third, handling the complexity introduced by production lines with different levels of digitalization adds another layer of difficulty. For example, advanced lines equipped with digital twins and Discrete Event Simulation (DES) tools can readily provide predicted task completion times for use in solution algorithms. In contrast, conventional lines lack these capabilities, making it more challenging to integrate them into the overall solution.

To tackle these challenges, this paper introduces a leader-follower hierarchical game decision-making model aimed at manufacturing load balancing in environments with high-mix production lines of different digitalization levels. The paper also develops a bi-agent genetic algorithm (BAGA) to optimize this model, accommodating multiple evaluation criteria that reflect the complexities of modern production. The algorithm contains leader agent GA and follower agent GA. Based on Stackelberg game model, The leader agent focuses on high-level decision-making criteria and the follower agent deal with low-level scheduling problem with engineering constraints. Additionally max-plus algebra model is utilized here to estimate task completion times for conventional production lines, which is essential for the load balancing process when advanced predictive tools like digital twins or discrete event simulations are not available.

The paper is organized as follows. Section 2 reviews relevant literature on Digital Twin technology and max-plus algebra in manufacturing systems. Section 3 explains how conventional production lines are modeled using max-plus algebra to estimate task completion times. In Section 4, the paper presents the manufacturing load balancing model, including the leader-follower hierarchical decision-making problem and its optimization formulation. Section 5 introduces a game-theoretical solution algorithm—a bi-agent genetic algorithm—that solves the leader-follower hierarchical optimization problem by handling multiple evaluation criteria and different levels of production line digitalization. Section 6 shows a case study based on connector assembly shopfloor and section 7 gives the conclusion remarks.

2. Related Work

Digital twin (DT) technology has emerged as a revolutionary approach in the realm of manufacturing, enabling real-time synchronization between physical production systems and their digital counterparts. For advanced production lines, digital twin-based Discrete Event Simulation (DES) offers a powerful tool for modeling, monitoring, and optimizing manufacturing processes. Digital twins facilitate real-time monitoring of production lines, providing instantaneous feedback on system performance. This capability enables prompt detection of anomalies, reducing downtime and improving overall efficiency (Qi & Tao 2018). By leveraging real-time data, digital twins can predict future system states and potential failures. This predictive capability aids in proactive maintenance, reducing unexpected breakdowns and extending equipment life (Tao et al. 2019).

Digital twin-based DES supports informed decision-making by simulating various scenarios and their impacts on production. This functionality helps in optimizing scheduling, resource allocation, and process flows, ultimately enhancing productivity (Rosen et al. 2015). Additionally, digital twins seamlessly integrate with Internet of Things (IoT) devices and Artificial Intelligence (AI) algorithms, enabling advanced data analytics and automation. This integration enhances the accuracy of simulations and the effectiveness of decision-making processes (Negri et al. 2017). However, for conventional production line with lower digitalization level, building a digital twin-based DES could be time consuming.

The use of max-plus algebra to model manufacturing flow lines allows for quick generation of equations for systems with both infinite and finite buffer sizes, facilitating rapid analysis and reconfiguration for a conventional production line. Seleim and ElMaraghy (2014) presented a method that simplifies this process, making it ideal for dynamic manufacturing environments. Chen et al. (2020) employed discrete event-driven model predictive control based on max-plus algebra for real-time work-in-process optimization in serial production systems. This approach significantly improved production efficiency by dynamically adjusting to real-time data and operational conditions. Al Bermanei et al. (2024) applied max-plus algebra to handle the scheduling of production systems with multiple stages and units, addressing issues such as cleaning times and minimizing total production time.

Balancing the manufacturing load is crucial for optimizing production, especially when dealing with production lines of different ages. This challenge is closely tied to order scheduling. Recent research has made significant improvements in order and task scheduling by using hybrid algorithms that are faster and offer better optimization. Jiacheng and Lei (2020) introduced a hybrid genetic algorithm that incorporates information entropy and game theory. This method improves solution accuracy, optimization ability, and stability compared to traditional genetic algorithms. Similarly, Mahmoudinazlou et al. (2023) developed a hybrid approach combining the Imperialist Competitive Algorithm and Simulated Annealing. They applied it to Order Acceptance and Scheduling problems with time windows and sequence-dependent setup times, achieving better optimization results and faster solutions. Additionally, Wei and Ma created a unified framework that integrates product and process models within ERP systems for Order Acceptance and Scheduling. This framework supports customer-focused production and effectively addresses issues related to capacity and customization. Overall, these studies demonstrate that hybrid algorithms and integrated systems can simplify Order Acceptance and Scheduling processes in complex, real-world manufacturing environments.

3. Data-Driven Representation of High-Mixed Production Lines

Optimizing the manufacturing loads between high-mix production lines requires robust modeling tools to model the system behavior. Conventional production lines, characterized by their lack of advanced digitalization and absence of digital twins, face challenges in accurately predicting production status and optimizing performance. Production system, as one type of discrete even dynamics systems, has several modeling tools like Petri-nets, Markov-chains, queuing networks, and max-plus algebra (Cassandras & Lafortune 2008). Max-plus algebra excels in modeling the timing and synchronization of events in production systems which is particularly advantageous for production line modeling without a digital twin. It focuses on the timing relationships between events rather than continuous variables. This approach simplifies the representation of production workflows and dependencies, providing a computationally efficient way to analyze the performance of the production system. Modeling a discrete event system (DES) using max-plus algebra can be represented in the following equations (De Schutter, 1996):

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k),\ y(k) = C \otimes x(k)$$

Here, $x(k)$ represents the state vector at each event step k, $u(k)$ is the input vector, and $y(k)$ is the output vector. The basic operations of max-plus algebra are maximization and addition, which differ from the conventional algebraic operations of addition and multiplication. The operators $\otimes$ and $\oplus$ denote max-plus multiplication and addition, respectively. Specifically, for two scalars $a$and b, max-plus addition is defined as $a \oplus b = \max(a, b)$, and max-plus multiplication is defined as $a \otimes b = a + b$. In the context of matrices, assuming A and B are two matrices then the operations can be extended as follows:

$$(A \otimes B)_{ij} = \max_{k}\left(A_{ik} + B_{kj}\right),\ (A \oplus B)_{ij} = \max\left(A_{ij} + B_{ij}\right)$$

Two special matrices in max-plus algebra are the identity matrix and the epsilon matrix (De Schutter, 1996). The identity matrix I, in max-plus algebra is analogous to the identity matrix in conventional algebra. For a matrix I of dimension $n \times n$, the diagonal elements are zero, and all off-diagonal elements are $-\infty$. This matrix acts as the neutral element for max-plus multiplication where $I \otimes A = A \otimes I = A$ for any matrix A. The epsilon matrix, ϵ, is a matrix where all elements are $-\infty$. It acts as the neutral element for max-plus addition where $A \oplus \epsilon = A$ for any matrix A.

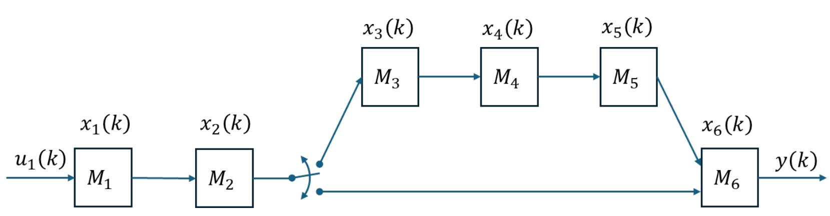

Figure 1. Production line with six Machines and two modes

Using max-plus algebra, a serial production line with six processes and two modes can be modeled, following the methodology presented by Bermanei et al. (2024). This serial production line will also be presented in case study with more details. As shown in figure 1, mode 1 includes three processes $M_1 \rightarrow M_2 \rightarrow M_6$, and mode 2 include six processes $M_1 \rightarrow M_2 \rightarrow M_3 \rightarrow M_4 \rightarrow M_5 \rightarrow M_6$. The intralogistics of the machine is a rotatory table where there are no buffers between processes. For mode 1, the production line can be represented as shown in equation 1, where p_i is the time used for i^{th} process, $x_i(k)$ represents the starting time of the i^{th} process for the k^{th} entered job, $u_1(k)$ represents the input time for the k^{th} job and $y(k)$ represents the time when the k^{th} product leaves the system.

$$X(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_4(k) \\ x_6(k) \end{bmatrix} = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon \\ p_1 & \epsilon & \epsilon & \epsilon \\ \epsilon & p_2 & \epsilon & \epsilon \\ \epsilon & \epsilon & p_4 & \epsilon \end{bmatrix} \otimes \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_4(k-1) \\ x_6(k-1) \end{bmatrix} \oplus \begin{bmatrix} e \\ \epsilon \\ \epsilon \\ \epsilon \end{bmatrix} \otimes u_1(k), \quad y(k) = x_6(k) + p_6 \tag{1}$$

Similarly, mode 2 of the production line can be represented in equation 2 as follows:

$$X(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ x_5(k) \\ x_6(k) \end{bmatrix} = \begin{bmatrix} \epsilon & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ p_1 & \epsilon & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & p_2 & \epsilon & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & p_3 & \epsilon & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & p_4 & \epsilon & \epsilon \\ \epsilon & \epsilon & \epsilon & \epsilon & p_5 & \epsilon \end{bmatrix} \otimes \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ x_4(k-1) \\ x_5(k-1) \\ x_6(k-1) \end{bmatrix} \oplus \begin{bmatrix} e \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \\ \epsilon \end{bmatrix} \otimes u_1(k), \quad y(k) = x_6(k) + p_6 \tag{2}$$

With using the form of switching max-pus linear model (Van Den Boom et al., 2020), the general equation is shown in equation 3. The mode of the production line is represented as $\sigma(k)$ where $\sigma(k) \in \{1,2\}$.

$$X(k) = A\big(\sigma(k)\big) \otimes X(k-1) \oplus B\big(\sigma(k)\big) \otimes u(k) \tag{3}$$

Iteratively applying the max-plus algebra operations over the sequence of steps from 1 to k, $X(k)$ can be derived based on $X(1)$ as shown in equation 4.

$$X(k) = \otimes_{i=1}^{k} A\big(\sigma(i)\big) \otimes X(1) \oplus \sum_{j=2}^{k} \otimes_{i=j}^{k} A\big(\sigma(i)\big) \otimes B\big(\sigma(j-1)\big) \otimes u(j-1) \tag{4}$$

By using the max-plus model for production lines with limited digitalization level, once the scheduled job sequence is received, it becomes possible to calculate when each job will be finished. This information can be used for optimization in both leader and follower problems. The limited digitalization level helps by simplifying the data requirements and focusing on key timing relationships, which are critical for managing and optimizing production in less digitally advanced environments. Moreover, using a switching max-plus mode allows for the model to handle different types of events or jobs from 1 to k, providing flexibility in managing diverse production scenarios.

4. Manufacturing Load Balancing as A Leader-Follower Hierarchical Game Decision-Making Problem

In the context of balancing the manufacturing load, the challenge arises from balancing multiple objectives that have different units and scales. Top management aims to maximize revenue by completing as many orders as possible, while operational management focuses on minimizing engineering costs and work-in-progress (WIP) costs. To effectively manage these competing objectives, a leader-follower hierarchical decision-making evaluation mechanism is proposed in this section. This evaluation mechanism enables the separate handling of strategic and operational levels, providing a robust and flexible solution to the complex problem at hand.

Manufacturing systems inherently possess a hierarchical structure where strategic decisions made by top management (leader) influence and constrain the operational decisions made by the shop floor management (follower). This natural hierarchy is essential for maintaining coherence and alignment between high-level goals and ground-level operations (Xu et al. 2012). The leader makes decisions first, setting the context for the follower's decisions. This sequential nature ensures that strategic objectives are met while operational constraints are respected. The interdependence between the leader and follower decisions ensures that the overall system objectives are optimized in a coherent manner (Deb & Sinha, 2009). By separating the objectives, the leader-follower framework avoids the need for arbitrary weighting factors that would otherwise be required to combine different metrics like revenue, engineering cost, and WIP cost. This separation allows for a more accurate and unbiased optimization process, as each level focuses on its specific objectives using appropriate units and scales (Sinha et al. 2013). The framework allows for iterative feedback loops, where operational decisions (followers) can inform and refine strategic decisions (leader) in subsequent iterations.

One of the primary reasons for employing a leader-follower hierarchical game decision-making approach is the difference in units and scales for revenue, engineering cost, and WIP cost. These metrics cannot be directly compared or combined without introducing arbitrary weighting factors, which can lead to biased or suboptimal solutions. In a single optimization problem, the need to appropriately weigh these different metrics complicates the objective function and makes it difficult to achieve a balanced solution. Revenue is measured in monetary units, reflecting the financial performance of the shop floor. Revenue recognition follows principles that ensure systematic and standardized reporting across different industries (Xu et al. 2012). Engineering Costs include costs related to resource usage, labor, and equipment depreciation. These are also expressed in monetary units but often on a different scale compared to revenue. Engineering costs are crucial for maintaining production quality and operational efficiency. WIP Cost represents the cost associated with incomplete products, including storage and holding costs. WIP costs are calculated based on the value of raw materials, direct labor, and manufacturing overheads that have been incurred for products still in the production process (Xu et al. 2012). The leader-follower hierarchical decision-making mechanism separates those goals of maximizing revenues and minimizing engineering and operational costs. Before introducing the hierarchical decision-making mechanism, assumptions are made as follows:

1) The shop floor has no prior information about manufacturing tasks that arrive in real time.
2) Each manufacturing task requires producing only one type of product variant.
3) Certain high-priority manufacturing tasks must be assigned to the production line.
4) The shop floor can decide whether to assign normal manufacturing tasks to any production line.
5) Once a job of a manufacturing task has been processed, the entire manufacturing task must be completed.
6) If multiple jobs of one manufacturing task are assigned to a production line, those jobs need to be processed consecutively.
7) Each production line can produce several types of product variants.
8) There is a changeover time when a production line switches from one product variant to another.

Figure 2 illustrates the leader-follower hierarchical decision-making mechanism designed to optimize manufacturing load balancing and planning in a shop floor. It is structured into two distinct levels of decision-making: the leader level and the follower level, each with its specific objectives. The accompanying Table 1 provides a detailed list of parameters and decision variables relevant to this hierarchical decision-making framework. As shown in Figure 2, in leader-level decision-making, it focuses on strategic decisions, primarily determining whether to assign rejectable task loads T_j to any production line M_k. The primary objective at this level is to maximize revenue level R from fulfilling manufacturing tasks. The revenue is calculated as a function of several factors including the revenue r_j from completing any tasks, tardiness penalty coefficients w_{tj}, due dates of manufacturing tasks d_j, manufacturing task completion times C_j, and the binary decision variables o_j indicating the acceptance.

At the follower level, decisions involve determining when and where each job t_i^j of a manufacturing task j is processed across the production lines. The goal here is to minimize the overall cost OC associated with engineering operations and work-in-progress (WIP). This cost is calculated as a function of inventory cost coefficients w_{cj}, engineering operational cost rates m_k, task completion times C_j, jobs finished times f_{ijk}, and job process times p_{ijk}. In the decision-making process, the follower-level receives task acceptance decisions o_j from the leader level. After determining the optimal assignment of jobs in each accepted task, the manufacturing task completion time C_j is obtained based on the finish time of each task f_{ijk}. The finish time is either obtained through running simulation of the current production plan or calculated using Max-plus algebraic production line representation model.

The engineering constraints in the leader-follower hierarchical decision-making mechanism are critical for ensuring the feasibility and coherence of the manufacturing load balancing and planning process. These constraints are designed to maintain consistency across both strategic and operational decision-making levels, which is crucial for the overall efficiency and effectiveness of the production system.

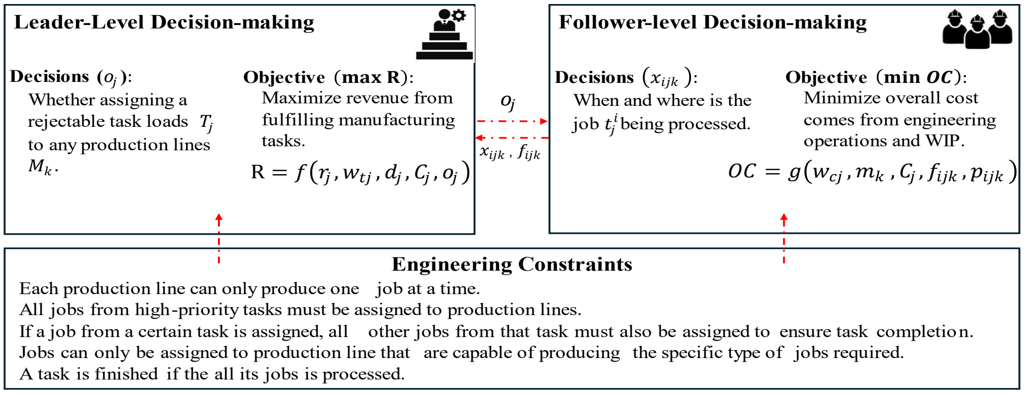

Figure 2. Leader-follower hierarchical game decision-making mechanism

Table 1. Parameters and decision variables

N	Number of manufacturing tasks.
m	Number of production lines.
J_i	Number of jobs in task j.
a_{ik}	If task j can be produced on production line k.
d_i	Due date for manufacturing task j
p_{iik}	Process time of job i in task j on production line k.
f_{iik}	Finished time of job k in task i on production line j
w_{ti}	Tardiness penalty coefficient for task j.
w_{ci}	Inventory cost coefficient for task j.
w_{ti}	Tardiness penalty coefficient for task j.
r_i	Revenue from completing task j.
x_{iik}	Binary variable indicating if job i of task j is assigned to line k
C_i	Completion time of task j.
o_i	Binary variable indicating if task j is accepted.
$s_{i' i k}$	Changeover time when switching from last job i' to job i on machine k.
$\delta_{i' i}$	Binary variable indicates if last job i' to job i on machine k are same type.
A_k	Production line k available time.
m_k	Engineering cost rate for operation machine k.
T	Number of manufacturing task.
L	Quantity of product types.
M_k	k^{th} machine in the shop floor.
OC	Overall cost due to engineering operations and WIP
R	Overall revenue obtained through fulfilling task loads.
$\mathbb{M}_1$	Group of conventional production lines.
$\mathbb{M}_2$	Group of production lines with limited digitalization level.
$\mathbb{N}_1$	Group of high priority manufacturing task.
$\mathbb{N}_2$	Group of normal manufacturing task.

The leader-follower hierarchical decision-making mechanism is crucial in managing the complexity of manufacturing load balancing and planning. By structuring decision-making into distinct strategic and operational levels, this mechanism ensures that both high-level revenue goals and detailed operational cost objectives are optimized coherently. The shared engineering constraints across both decision-making levels ensure consistency and feasibility in the production planning and execution processes.

5. Optimization Problem Formulation of Leader-Flower Hierarchical Game Decisions

The proposed leader-follower optimization model is formulated to address the dual objectives of revenue maximization and operational cost minimization in a shop floor characterized by multiple production lines and tasks. The model is structured into a leader problem that determines whether assigning task loads, and a follower problem that optimizes job assignment and planning. Leader objective function is defined in equation 5.1 which aims to maximize the total revenue R from fulfilling more manufacturing tasks and minimizing the penalties associated with tardiness for assigned tasks. According to table 1, the first summation term captures the revenue from assigned tasks within the set $\mathbb{N}_2$ considering potential tardiness penalties.

The second summation term accounts for the revenue and penalties for high priority tasks in the set $\mathbb{N}_1$ which must be assigned. In equation 5.1, o_j is a binary variable indicating whether assigning manufacturing task j, r_j represents the revenue from fulfilling task j, w_{tj} denotes the tardiness penalty coefficient for task j, C_j is the completion time of task j, and d_j indicates the due date for task j. The follower problem's objective function focuses on minimizing the overall operational cost (OC) as shown in equation 5.2. This equation comprises two main components: the inventory cost and the operational engineering cost. The first term represents the inventory cost incurred due to the difference between task completion times and individual job finish times. The second term captures the additional engineering costs related to the spread between the earliest start and latest finish times on each machine. In equation 5.2, w_{ci} is the inventory cost coefficient for task j, x_{ijk} is a binary variable indicating whether job i of task j is assigned to production line k, m_k represents the additional engineering cost rate for operating machine k, f_{ijk} is the finish time of job i in task j on machine k, and S_{ijk} is the start time of job.
i. e.,

$$\mathbf{min}\ R = -\left[\sum_{j=1, j\in\mathbb{N}_2}^{T} o_j\left(r_j - w_{tj}\, max\left(0, C_j - d_j\right)\right) + \sum_{i=1, i\in\mathbb{N}_1}^{T}\left(r_j - w_{tj}\, max\left(0, C_j - d_j\right)\right)\right] \tag{5.1}$$

$s.t.$

$$\mathbf{min}\, OC = \sum_{j=1}^{T}\left(w_{cj}\sum_{k=1,}^{m}\sum_{i=1}^{J_i} x_{ijk}\left(C_j - f_{ijk}\right)\right) + \sum_{k=1,}^{m} m_k\left(\max_{i,j} f_{ijk} - \min_{i,j} S_{ijk}\right) \tag{5.2}$$

$s.t.$

$$o_j \in \{0,1\}, j \in \mathbb{N}_2 \tag{5.3}$$

$$C_j = \max_{i,k} f_{ijk}, \forall i, k \tag{5.4}$$

$$x_{ijk} \le a_{jk} \tag{5.5}$$

$$\sum_{k=1}^{m}\sum_{i=1}^{J_i} x_{ijk} = o_j \cdot J_j, \forall i, k, j \in \mathbb{N}_2 \tag{5.6}$$

$$\sum_{k=1}^{m}\sum_{i=1}^{J_i} x_{ijk} = J_j, \forall i, k, j \in \mathbb{N}_1 \tag{5.7}$$

$$f_{ijk} = S_{ijk} + p_{ijk}, k \in \mathbb{M}_1 \tag{5.8}$$

s $$X\left(\sum_{j\in j'}\sum_i\sum_k x_{ijk}\right) = \bigotimes_{z=1}^{\sum_{j\in j'}\sum_i\sum_k x_{ijk}} A\left(\sigma(z)\right) \otimes X(1) \oplus \sum_{b=2}^{\sum_{j\in j'}\sum_i\sum_k x_{ijk}} \bigotimes_{z=b}^{\sum_{j\in j'}\sum_i\sum_k x_{ijk}} A\left(\sigma(z)\right) \otimes B\left(\sigma(b-1)\right) \otimes u(b-1),\ k \in \mathbb{M}_2 \tag{5.9}$$

$$f_{ijk} = X\left(\sum_{j\in j'}\sum_i\sum_k x_{ijk}\right) + t_6, k \in \mathbb{M}_2 \tag{5.10}$$

$$S_{ijk} \geq A_k \cdot x_{ijk}, \forall i, j, k \tag{5.11}$$
$$S_{ijk} \geq f_{i'jk} + s_{i',i,k} \cdot \delta_{i',i,k} \tag{5.12}$$
$$R \leq 0 \tag{5.13}$$
$$OC \geq 0 \tag{5.14}$$

Equation 5.3 to equation 5.14 are constraints shared by both leader and follower optimization problem. Equation 5.3 ensures that the decision of whether assigning task j is binary. The task completion time C_j for each task j is defined as the maximum finish time of all jobs within that task across all machines as shown in equation 5.4. Equation 5.5 ensures the task assignment constraint that a job i in task j can only be assgined to a machine k if the machine is capable of handling that task (a_{jk}). Equation 5.6 and 5.7 are two job planning constraints that ensure that each job in a task is assigned to exactly one machine. For tasks in $\mathbb{N}_2$ (normal priority tasks), the assignment is conditional on the task being accepted ($o_j = 1$).

For tasks in $\mathbb{N}_1$ (high priority tasks), all jobs must be assigned unconditionally. Equation 5.8 defines how the finish time of a job is obtained from digital-twin based DES where S_{ijk} and p_{ijk} are both obtained from the simulation runs of planned task on production line k that belongs to conventional production lines $\mathbb{M}_1$. To obtain the finish time of each assigned job on production lines with limited digitalization level, equation 5.9 and equation 5.10 are derived from equation 1 to 4. In equation 5.9, $\sum_{j \in j'} \sum_i \sum_k x_{ijk}$ represent the order of job i in task j be processed on machine k where j' are tasks be processed before and processing according to assumption 6 in chapter 2. Therefore, $X\left(\sum_{j \in j'} \sum_i \sum_k x_{ijk}\right)$ represents the starting time of each process of job i on machine k and the finish time (when job i is leaving the production system k) is defined in equation 5.10. t_6 is the processing time of the last process as mentioned in max-plus model representation. Equation 5.11 ensures that the start time of any jobs on machine k not begin before the machine available time A_k. Non-overlapping job execution constraints are ensured by equation 5.12. No two jobs are processed on the same machine at the same time, accounting for changeover times when switching between different product types. Here, $\delta_{i',i}$ is an indicator variable represents if job i and last job i' and $s_{i',i,k}$ is the changeover time from job i' to last job i on machine k. Equations 5.13 and 5.14 ensures that overall revenue R and operation cost OC remains reasonable numbers.

6. Bi-agent genetic algorithm for model solution of Leader-flower Hierarchical Optimization

The leader-follower problem for manufacturing task loads balancing and planning defined in previous section involves two interconnected decision-making levels, each with distinct objectives. Both level sharing constraints and certain decision variables. The leader problem focuses on maximizing overall revenue by deciding whether to assign each manufacturing task, considering tardiness penalties and due dates. The following problem aims to minimize operational costs by optimizing job assignments and scheduling on multiple production lines with considering inventory costs, engineering costs, and setup times. Due the hierarchical decision-making structure, a bi-agent genetic algorithm (BAGA) is proposed here. This solution approach leverages the Stackelberg game framework, where the leader agent makes the initial decision, subsequently guiding the follower agent. The leader agent's decisions set the stage for the follower agent's optimization process, reflecting a hierarchical decision-making structure. The proposed BAGA can effectively captures the interdependence between high-level task acceptance decisions and detailed operational scheduling. The Stackelberg game model encapsulates this hierarchical interaction, enabling the BAGA to robustly address the intertwined objectives of revenue maximization and cost minimization in a hyper-reconfiguration manufacturing system on a shop floor.

6.1 Bi-agent genetic algorithm architecture

The workflow of proposed BAGA solution algorithm is illustrated in figure 3. As shown in the figure, the algorithm starts with chromosomes populations generation in the leader level. Step 1 creates population of potential solutions for the leader problem, where each individual represents a vector of binary decision variables vector o which its element o_j indicating whether task j should be assigned to any production lines. In step 2, the feasibility of decision variables vector o is validated based on equation 5.3 where high priority tasks must be assigned. After the validation, feasible chromosomes are sent to follower optimization problem.

At the follower level, each individual decision variables vector o goes through step 3.1 to step 3.6. In step 3.1, initial populations of planning variables x and sequence variable q are generated based on one decision variables vector o. x is a 3D matrix containing of elements x_{ijk}. Sequence q is a 2D matrix with size of $m \times N$ which each of its row represents the task processing sequence of a production line. Based on assumption 6 introduced in chapter 2, Sequence

q is necessary since jobs from the same task are processed consecutively. Assume that there are 5 tasks in the planning time window and 2 productions, first row of sequence matrix $[1,2,3,4,5]$ represents that production line 1 will process all assigned jobs of task 1 first. Furthermore, constraint defined by equation 5.2 is followed while generating plan variables x. For each job in the manufacturing task, it will only be assigned to a capable production line. Each decision variables vector o will randomly generate an amount of population size potential solutions. In step 3.2, with the task load plan given in step 3.1, the finish time of each assigned job f_{ijk} is obtained through Max-plus model or DES model of production lines. Based on equation 5.2, fitness of the whole follower population is evaluated in step 3.3. Before reaching the limit iterations in the follower level, selection, crossover and mutation operations are applied to feasible chromosomes in step 3.5. To maintain the follower population size and diversity, infeasible chromosomes will be replaced by newly generated initial chromosomes or offsprings. In step 4, Infeasible chromosomes will be repaired by ensuring assigning all high priority tasks. In step 5, the leader level fitness of optimal set of variables (o, x, q) from the follower is evaluated based on equation 5.1. Before reaching the iteration limit in the leader level, selection, crossover, and mutation operations are applied to feasible chromosomes to explore the solution space and improve population fitness.

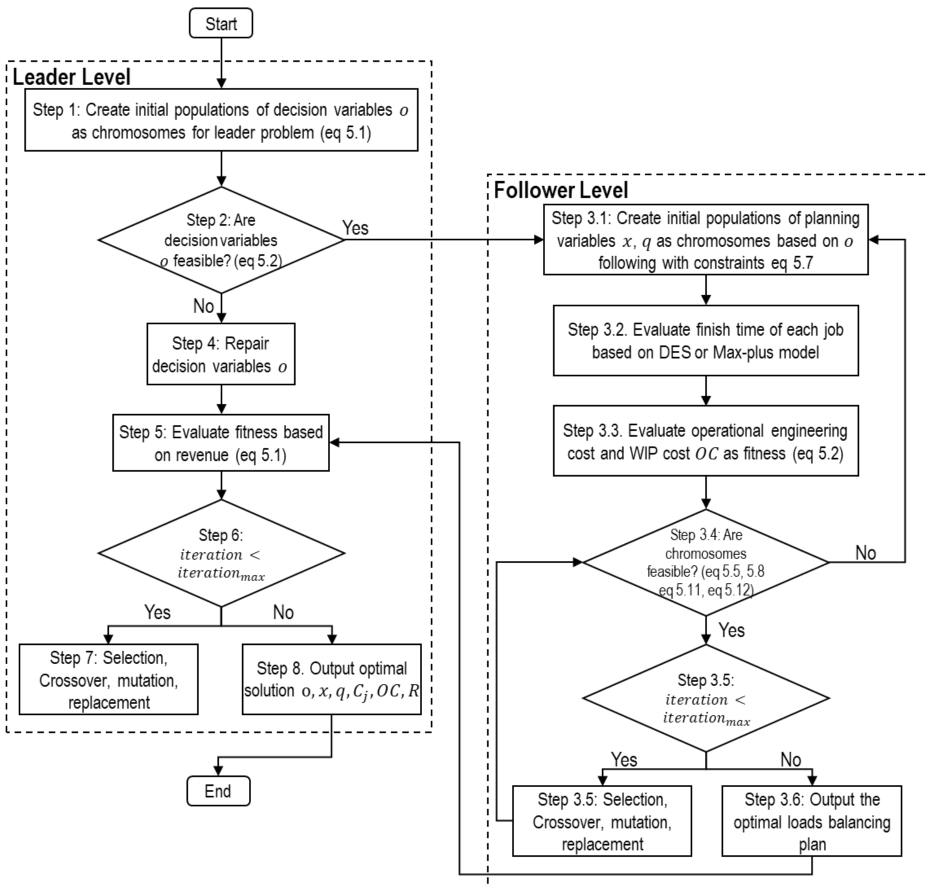

Figure 3. Overall flow chart of bi-agent generic algorithm

6.2 Encoding and operation

The proposed BAGA for the decision-making problem involves a robust encoding, crossover, and mutation scheme at both levels. As shown in figure 4, an encoding strategy without compression is employed for chromosome encoding in the leader level. Decision variable o is vector with length of number of manufacturing tasks N. Planning variable x generated in the follower level is a $\max(J_1, J_2, \dots, J_N) \times N \times m$ containing elements of x_{ijk}. For a feasible solution, if $o_j = 1$, then $\sum_i \sum_k x_{ijk} = J_j$. In the leader level, single point crossover method is used to maintain genetic diversity and explore the solution space effectively. As illustrated in figure 4a, suppose there are five manufacturing tasks and five production lines, a crossover point is randomly selected along the task dimension and applied to both x and o. The parent chromosomes are then split at this point, and their segments are exchanged to produce two offspring. After the crossover, $\sum_i \sum_k x_{ijk} = J_j$ still holds true. This method allows for more complex mixing of genetic material, increasing the likelihood of finding optimal solutions. Figure 4b shows the mutation operation of chromosomes in the leader level. Each element in o has a probability of flipping its value from 0 to 1 or vice versa. x will be updated based on o in the next follower-level optimization process. Figure 5 illustrates the crossover in the follower level. A crossover point is randomly selected along the task dimension and applied to both x and q. The parent chromosomes are then split at this point, and their segments are exchanged to produce two offspring. To avoid the duplication of elements in sequence q after crossover, duplicated elements will be changed.

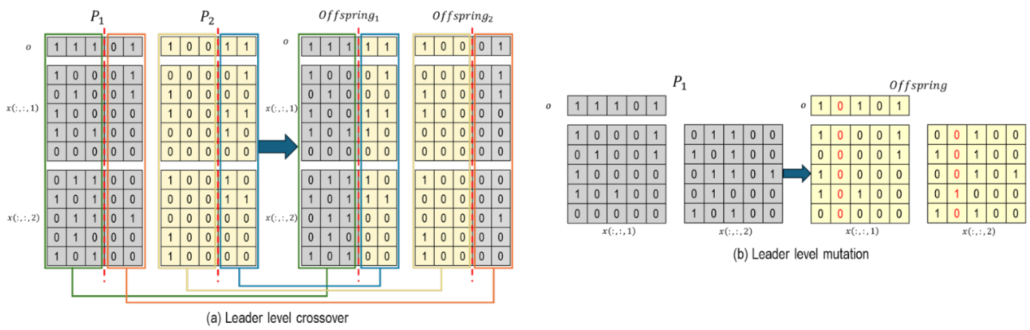

Figure 4. Leader level crossover and mutation operation (to save some space, you may put them in one row)

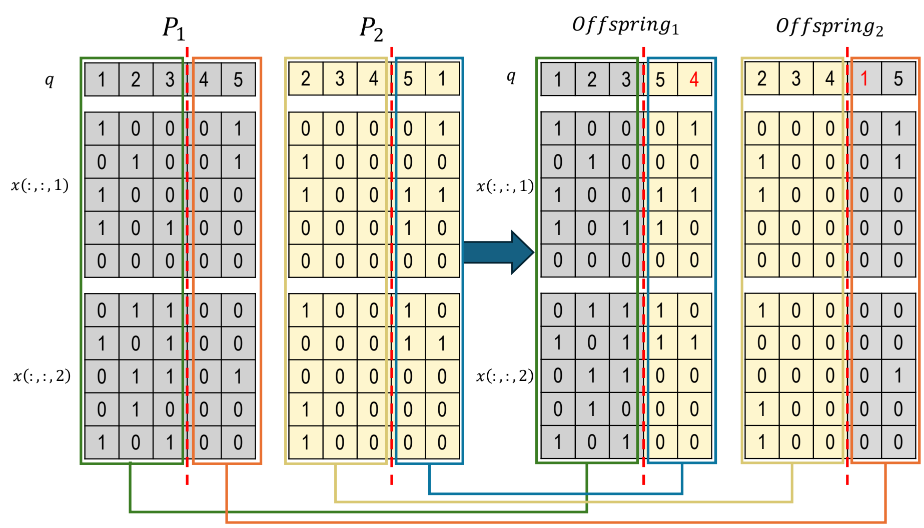

Figure 5. Follower level crossover operation

7. Case Study

This case study examines manufacturing load balancing and planning for assembling six types of connectors across ten production lines. Management aims to maximize revenue by assigning as much fulfillable load as possible, while the operations team seeks to minimize engineering costs by completing assignments promptly. The shop floor operates 24/7 in three shifts. The ten production lines vary in digitalization and capabilities. As shown in table 2, lines 1–7 are less automated and require more operators, while lines 8–10 use a DES model for finish time predictions; the others use a max-plus model. Engineering operational costs are measured in resource units, representing the percentage of time engineers or operators are needed. Machine available time indicates when a production line can start on its first assigned tasks within the planning window. In this case study, operational engineering cost m_k is defined as operator to machine $\text{ratio}_k \times$ engineering operational cost_k. Table 3 shows the unscheduled tasks in the planning window for the case study. The experiments are run using MATLAB R2024a on a Windows 11 system, 32 Gb ram, a processor AMD Ryzen 5800X3D. In the leader-level GA, the maximum number of iterations is 50 with the population size of 100. The crossover rate is set to be 0.5 and the mutation rate is set to be 0.2. In the follower-level GA, the maximum number of iterations is 50 with the population size of 50. The whole NGA involves 5000 chromosomes and 2500 iterations.

Table 2. Manufacturing Capabilities and Operational Metrics of Production Lines

Production Line	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10
DES								x	x	x
Max-Plus model	x	x	x	x	x	x	x			
Operator to Machine ratio	1	1	1	1	2	2	2	0.5	0.5	0.5
Engineering operational cost	20	20	20	20	20	20	20	35	35	35
Machine Available Time (min)	0	10	3	7	23	31	8	28	0	56

Table 3. Tasks in the planning window

	Job Type	Job quantity	Due date (min)	Priority	Price ($)	Tardiness Penalty ($/min)	Stock Cost
Task 1	4P CAP	50	2500	High	25000	500	4
Task 2	4P Tall CAP	30	2500	High	30000	800	8
Task 3	3P No Cap	20	1500	Normal	5000	200	3
Task 4	3P Tall Cap	20	1500	Normal	7500	200	6
Task 5	4P No Cap	15	750	Normal	4500	200	4
Task 6	4p Cap	40	1500	Normal	24000	400	4
Task 7	3p Cap	10	1000	Normal	2000	200	3
Task 8	4P Tall CAP	80	3500	High	48000	500	8
Task 9	4P Cap	30	1500	High	12000	800	4
Task 10	3p Cap	20	500	High	5000	800	3
Task 11	4P Tall CAP	50	4000	Normal	28000	200	8
Task 12	4P No Cap	33	2000	Normal	7000	200	4
Task 13	4P Cap	18	600	Normal	13000	500	4
Task 14	3P Tall Cap	5	100	Normal	2500	800	6
Task 15	4P Cap	9	130	Normal	8000	800	4

Figure 6 illustrates the convergence behavior of revenue and operational cost over 50 generations during the optimization process. The red dashed line represents revenue, which starts at a lower value and steadily increases until it stabilizes at around $146,000 after approximately 15 generations. The blue dashed line represents the operational

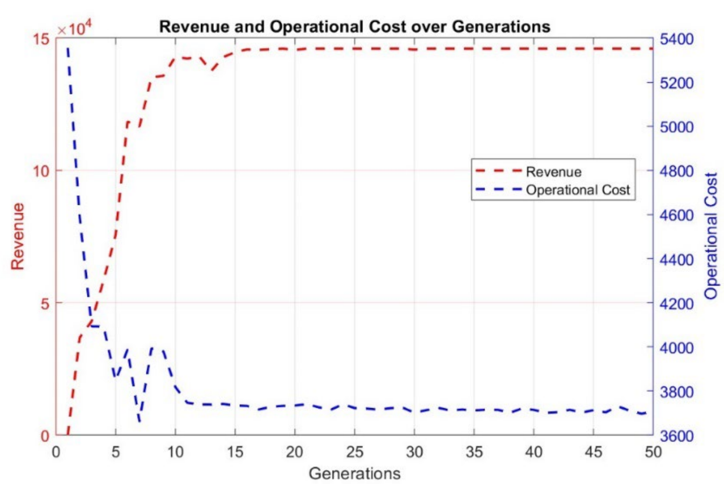

Figure 6. Convergence of Revenue and Operational Cost over Generations

cost, which initially fluctuates but quickly stabilizes around a value of 3,800 resource units after approximately 10 generations. The convergence patterns indicate the effectiveness of the nested genetic algorithm in balancing the trade-offs between maximizing revenue and minimizing engineering operational costs within a relatively short number of generations.

8. Concluding Remarks

Balancing manufacturing loads in environments that blend advanced and conventional production lines is a complex task, especially with the growing demand for mass customization and personalized orders. Traditional scheduling methods often struggle under these dynamic conditions. To address this, the paper leader-follower decision-making model that separates strategic order acceptance from operational task scheduling in high-mix production environments. The leader level focuses on maximizing revenue by deciding which orders to accept, considering due dates and potential tardiness penalties. The follower level handles the efficient scheduling of tasks and resource management across different production lines. It also developed a bi-agent genetic algorithm to optimize this hierarchical game decision-making model, allowing us to consider multiple evaluation criteria without resorting to arbitrary weighting. By leveraging digital twins and discrete event simulation tools for advanced lines, and a max-plus algebra model for conventional lines, we integrated production lines with varying technological capabilities into a unified scheduling process.

References

Al Bermanei, H., Böling, J. M., and Högnäs, G., *Modeling and scheduling of production systems by using max-plus algebra*, Flexible Services and Manufacturing Journal, vol. 36, no. 1, pp. 129-150, 2024.

Cassandras, C. G. and Lafortune, S., *Introduction to discrete event systems*, Springer eBooks, 2008.

Chen, W., Gong, X., Rahman, H. F., Liu, H., and Qi, E., *Real-time order acceptance and scheduling for data-enabled permutation flow shops: bilevel interactive optimization with nonlinear integer programming*, Omega, vol. 105, pp. 102499, 2021.

De Schutter, B., *Max-Algebraic System Theory for Discrete Event Systems*, PhD Thesis, Faculty of Applied Sciences, K.U.Leuven, ISBN 90-5682-016-8, pp. 331, 1996.

Deb, K. and Sinha, A., *Solving bilevel multi-objective optimization problems using evolutionary algorithms*, Evolutionary Multi-Criterion Optimization, pp. 110-124, 2009.

Jiacheng, L., and Lei, L., "A hybrid genetic algorithm based on information entropy and game theory," *IEEE Access*, vol. 8, pp. 36602–36611, 2020.

Mahmoudinazlou, S., Alizadeh, A., Noble, J., and Eslamdoust, S., "An improved hybrid ICA-SA metaheuristic for order acceptance and scheduling with time windows and sequence-dependent setup times," *Neural Computing and Applications*, vol. 36, no. 2, pp. 599–617, 2023.

Negri, E., Fumagalli, L., and Macchi, M., A review of the roles of digital twin in CPS-based production systems, Procedia Manufacturing, vol. 11, pp. 939–948, 2017.

Qi, Q., and Tao, F., Digital twin and big data towards smart manufacturing and industry 4.0: 360 degree comparison, IEEE Access, vol. 6, no. xx, pp. 3585–3593, 2018.

Rosen, R., Von Wichert, G., Lo, G., and Bettenhausen, K. D., About the importance of autonomy and digital twins for the future of manufacturing, IFAC-PapersOnLine, vol. 48, no. 3, pp. 567–572, 2015.

Seleim, A. and ElMaraghy, H., *Max-Plus modeling of manufacturing flow lines*, Procedia CIRP, vol. 17, pp. 71-75, 2014.

Sinha, A., Malo, P., and Deb, K., *Efficient evolutionary algorithm for single-objective bilevel optimization*, arXiv (Cornell University), 2013.

Van Den Boom, T. J. J., Van Den Muijsenberg, M., and De Schutter, B., *Model predictive scheduling of semi-cyclic discrete-event systems using switching max-plus linear models and dynamic graphs*, Discrete Event Dynamic Systems, vol. 30, no. 4, pp. 635-669, 2020.

Wei, J., and Ma, Y., "Design of an order acceptance and scheduling module in a unified framework with product and process features," *IEEE International Conference on Automation Science and Engineering (CASE)*, vol. 8, pp. 968–973, 2012.

Xu, Y., Elgh, F., Erkoyuncu, J. A., Bankole, O., Goh, Y., Cheung, W. M., Baguley, P., Wang, Q., Arundachawat, P., Shehab, E., Newnes, L., and Roy, R., *Cost engineering for manufacturing: current and future research*, International Journal of Computer Integrated Manufacturing, vol. 25, no. 4-5, pp. 300-314, 2012.

Biographies

Mulang Song is a graduate student in the School of Mechanical Engineering, Georgia Institute of Technology (USA). His research interest is mainly related to Engineering Design, Blockchain, and the associated applications in Crowdsourced Manufacturing.

Dr. Jiao is the editor-in-chief of Journal of Engineering Design and an associate professor of mechanical and industrial engineering at Georgia Tech, USA. Prior to joining the School of Mechanical Engineering at Georgia Tech in December 2008, he was an Assistant Professor and then Associate Professor in the School of Mechanical and Aerospace Engineering at Nanyang Technological University, Singapore. Before his career in Singapore, he was a Visiting Scholar in the Department of Industrial Engineering and Engineering Management at Hong Kong University of Science and Technology from 1998 to 1999. From 1993 to 1994, he was a Lecturer of Industrial Engineering in the School of Management at Tianjin University, China, and from 1988 to 1990, he worked as an Associate Lecturer in the Department of Industrial Design at Tianjin University of Science and Technology, China. More info about his research: https://scholar.google.com/citations?user=9yikEHAAAAAJ&hl=en&oi=ao.