IJAMMP

Cooperative Multi-Agent Scheduling to Improve Resource Utilisation in Large-Scale Manufacturing

Cheng Qian* and Yuchun Xu†

RESEARCH ARTICLE

ARTICLE INFO

Keywords:

Cooperative Game Theory;
Discrete Manufacturing;
Online Scheduling;
Resource Optimisation;
Sustainable Manufacturing

Article History

*Received:*24 October 2025
*Revised:*19 February 2026
*Accepted:*28 February 2026
Published:

ABSTRACT

Scheduling production in large-scale manufacturing is a complex challenge that requires dynamically allocating tasks to machines while balancing throughput and resource utilization. Game-theoretic approaches have been widely applied, with traditional models treating either all machines as players competing for tasks or all tasks competing for access to machines. Such formulations overlook the joint perspective of machines and tasks, often leading to schedule fragmentation, i.e. small, unusable gaps that severely reduce machine utilisation. This paper introduces a novel cooperative framework in which both machines and tasks act as players, negotiating directly to determine feasible start times when tasks are assigned to machines. Manufacturing systems are modelled as multi-agent networks, with a classifier selecting scheduling strategies based on the relative importance of machines and tasks. Machine importance is dynamically assessed using PageRank centrality, while task importance is determined by urgency and complexity. The negotiation process is formulated as a cooperative game that maximizes a collective utility function incorporating processing costs, penalties for early or late completion, importance of machines and tasks, and a new measure of opportunity loss caused by fragmented schedules. This cooperative game model can result a schedule with minimised gaps, thereby improving machine utilisation without compromising production speed. A discrete-event simulation was conducted to evaluate performance against a traditional multi-objective optimization approach. The findings indicated that 76% of machines identified as bottleneck resources demonstrated an average enhancement in utilisation of 1.5%, thus providing an effective solution for efficient and sustainable production management.

1. Introduction

Large-scale discrete manufacturing systems, which produce distinct items such as automobiles, aircraft or electronics through assembly and individualised processing steps, are characterised by complex product routings, diverse resources, and dynamic job arrivals. These systems present a substantial challenge for production scheduling. The core objective is the efficient real-time allocation of tasks to manufacturing machines, a problem central to operational efficiency. In this domain, two paramount yet often conflicting performance metrics are of particular significance: the maximisation of throughput (i.e. the minimisation of make-span) and the maximisation of machine utilisation (Veeramachaneni et al., 2025). The former is crucial for meeting delivery deadlines and reducing work in progress, while the latter is vital for spreading high capital costs over time and improving return on investment.

Smart and Sustainable Manufacturing Research Centre, Aston University, Birmingham, B4 7ET, United Kingdom

Corresponding authors: * c.qian4@aston.ac.uk, † y.xu16@aston.ac.uk

Conventionally, the most widely adopted approach to resolving these competing objectives is through the utilisation of centralised multi-objective optimisation techniques, including genetic algorithms and particle swarm optimisation (Wong & Ngan, 2013). These methods model the entire shop floor as a single entity, seeking a Pareto-optimal set of schedules that represent the best possible trade-offs. However, in dynamic and large-scale environments, this paradigm exhibits significant limitations. The computational complexity of these algorithms often renders them prohibitive for real-time, online scheduling (Megow et al., 2006), where online scheduling is the process of making immediate and dynamic decisions to assign tasks to resources as they arrive, without prior knowledge of future tasks. Moreover, their inherent reliance on a global, often static, snapshot of the system makes them vulnerable in the face of frequent disruptions, such as machine breakdowns or rush orders, necessitating computationally expensive rescheduling (Uhlmann & Frazzon, 2018).

An alternative approach is represented by decentralised, agent-based paradigms, which have gained traction for their flexibility and responsiveness (Barenji et al., 2017). The distribution of decision-making authority to individual machine or task agents enables these systems to react swiftly to local changes (Zhang et al., 2016). Nevertheless, this very strength is a source of a critical weakness: a focus on local optima. When agents act in their own self-interest, for instance, a machine selecting the most immediately profitable task without considering global schedule health, the emergent global schedule can be highly inefficient. A particularly negative consequence of this inefficiency is schedule fragmentation, where the creation of small, non-utilisable time gaps between tasks on machine timelines leads to a significant degradation in overall utilisation. It is therefore evident that a significant research gap exists in this domain: i.e., the necessity for a decentralised scheduling methodology that retains the flexibility of multi-agent systems but is explicitly adapted to counteract local selfishness and the ensuing schedule fragmentation, thereby achieving a system-level balance between throughput and utilisation. The specific research questions are as follows:

(1) How can a cooperative game be formally modelled between machine and process of the task in a discrete manufacturing network to simultaneously address throughput and machine utilisation?

(2) How can metrics of network structural (e.g., PageRank centrality) be integrated into the utility functions to guide the negotiation process towards system-level efficiency?

The present paper addresses this gap by introducing a cooperative game-theoretic model for online resource allocation in manufacturing systems. Unlike conventional competitive game models, which typically frame scheduling as a contest among machines, the proposed approach enables direct negotiation between machines and tasks to determine processing start times.

By structuring these interactions as cooperative efforts aimed at maximising a joint utility function, which includes explicit costs for schedule fragmentation, the system moves towards globally efficient and robust schedules. To capture the dynamic importance of machines, a graph-theoretic importance metric based on PageRank centrality is used to inform the influence and the bargaining power of machines. Task importance, in turn, is derived from urgency and complexity, ensuring balanced prioritization. The practical effectiveness of this approach is demonstrated through a multi-agent discrete-event simulation, benchmarking it against a traditional multi-objective optimisation heuristic.

The remainder of this paper is structured as follows. Section 2 reviews relevant literature on manufacturing networks and the application of multi-objective optimisation in manufacturing systems. Section 3 introduces the methodology for online scheduling in large-scale discrete manufacturing systems with a strategy classifier. Section 4 details the problem formulation and the proposed cooperative game-theoretic model. Section 5 outlines the experimental setup and discusses the comparative results. Finally, Section 6 concludes with findings and prospective future research directions.

2. Literature Review

This section reviews the typical scheduling methods and analytical tools for discrete manufacturing systems.

2.1 Scheduling in Discrete Manufacturing Systems: Modelling, Solving, and the Game-Theoretic Methods

Scheduling in discrete manufacturing is a complex, typically NP-hard, problem that is central to operational efficiency (Allahverdi et al., 2018). The primary challenge is to optimise the allocation of resources and the sequencing of operations over time, with the objective of fulfilling a set of criteria, typically including the minimisation of make-span and the maximisation of machine utilisation. Traditional approaches have largely relied on centralised optimisation techniques. Metaheuristics, such as Simulate Annealing (Gaafar & Masoud, 2005), Genetic Algorithms (Zan et al., 2020) and Particle Swarm Optimisation (Sha & Lin, 2010), have been widely applied to find near-optimal solutions for these multi-objective problems (Zhang et al., 2025). Nevertheless, the computational demands of these approaches tend to restrict their practicality in online scheduling contexts, where the occurrence of dynamic events, such as the arrival of new tasks and machine breakdowns, requires frequent and efficient rescheduling (Marzia & Azab, 2023).

The limitations of centralised planning in dynamic contexts have stimulated interest in decentralised paradigms. Minguillon & Lanza (2019) studied the coupling of centralised and decentralised scheduling for the robustness in agile manufacturing systems. A number of agent-based mechanisms were also developed for production planning and scheduling (Jules & Saadat, 2016; Poudel et al., 2023). Among these, game theory has emerged as a promising framework for modelling the strategic interactions among distributed entities in a manufacturing system (Sanogo et al., 2025). Early applications often modelled scheduling as a competitive process. E.g., auction-based mechanisms have been explored where tasks bid for machines or machines bid for tasks, effectively treating the shop floor as a market. (Kang et al., 2020). Other non-cooperative models have analysed competition among tasks or among manufacturers (Liu et al., 2021; Hu et al., 2025). While these approaches offer enhanced flexibility, they can lead to system-wide inefficiencies, such as the creation of schedule fragmentation, as self-interested agents prioritise local gains over global performance. The potential of cooperative game theory to align local decisions with global objectives in manufacturing scheduling remains a less explored avenue compared to its competitive counterparts, particularly for direct machine-process negotiation over the starting time of the process.

2.2 Multi-Agent Systems and Importance Measures in Discrete Manufacturing Systems

Multi-Agent Systems (MAS) provide a natural architectural blueprint for implementing decentralised manufacturing control (Lee & Kim, 2008). In MAS, autonomous agents interactively collaborate to solve a variety of problems, which exceed their individual capabilities. These agents can be viewed as representations of physical resources, including machines, and logical entities, such as orders (Zhang et al., 2016). This paradigm facilitates collaborative and self-organising behaviour, moving beyond a rigid, centralised control structure (Zheng et al., 2021). Agents can react locally to disruptions (Bi et al., 2024), and negotiate to find feasible schedules (Poudel et al., 2023), leading to increased robustness and adaptability (Leitão, 2009).

The structure of interactions in such a discrete manufacturing system inherently forms a network. In light of this realisation, researchers have commenced an analysis of manufacturing systems through the theoretical framework of network science (Li et al., 2017; Yang et al., 2021). This perspective facilitates the quantification of structural properties and the identification of critical nodes that influence overall system performance. By employing various network centrality metrics, such as betweenness centrality (Xin et al., 2019) and eigenvector centrality (Zeng et al., 2024), the importance of machine nodes within different operational scenarios of manufacturing systems can be quantitatively assessed. Metrics like PageRank, originally developed for web search, can also be adapted to rank machines (Zhang et al., 2017) or processes (Zhu et al., 2023) based on their connectivity and importance within the production network. This provides a data-driven, dynamic measurement of a machine's systemic influence, representing a significant advancement over static, pre-defined priorities.

Although MAS and game theory offer frameworks for decentralised scheduling, current game-theoretic models are essentially competitive, which may result in schedule fragmentation and reduced overall efficiency. Furthermore, while network science provides tools to analyse manufacturing systems structurally, its integration into the operational decision-making of production scheduling is limited. Consequently, a novel approach is required that integrates a cooperative game-theoretic model for direct task-machine negotiation with a network-science-informed perspective, using metrics such as PageRank to explicitly guide agents towards decisions that mitigate fragmentation and balance throughput with utilisation.

3. A Flexible Strategy Classifier for Online Scheduling

This section designs a flexible classifier for machine-process matching strategies, engineered to meet the demand for efficient, high-utilisation resource allocation solutions across diverse production scenarios.

3.1 The Conventional Online Scheduling Methods

Since the processes of tasks may arrive at the manufacturing network at any arbitrary future moment, the resource allocation process for each new task constitutes an online scheduling problem. A "capability-availability-efficiency" filtering principle is typically employed in this regard (Qian et al., 2019). Specifically, a process, based on its processing requirements, first filters a set of candidate machines possessing the specified processing capabilities. Subsequently, the process initiates query requests to these machines to gather their available time slots. Finally, a multi-objective optimisation algorithm is employed to identify the machine and time slot that best satisfy the processing objectives, thereby completing the matching process. In this procedure, the newly arrived task process should explicitly select a specific processing time slot from a range of available slots on the candidate machines. The selection of algorithm has a direct impact on the generated schedule, yielding various solutions with distinct levels of machine utilisation and task completion times. The flexible classifier proposed in Section 3.3 is capable of adapting a machine-process matching strategy based on the specific context, thereby achieving an effective balance between the utilisation of machines and the efficiency of production in the manufacturing network.

3.2 Defining the Importance of Machines and Processes

The selection of a matching strategy is fundamentally influenced by two parameters: the importance of the machine and the task. These are formally defined as the Machine Importance ($\boldsymbol{I}_{\mathrm{m}}$) and the Process Importance ($\boldsymbol{I}_{\mathrm{p}}$), respectively.

Equation (1) defines the importance of a machine i within the manufacturing network. This is calculated as the PageRank centrality value $\boldsymbol{PR}_{\mathrm{i}}$ of the node $\boldsymbol{i}$ within a directed machine cooperation network $\boldsymbol{CopNet_N}$, which is constructed from the most recent $\boldsymbol{N}$ upstream-to-downstream process relationships.

$$I_{\mathrm{m}}(i) = PR_i(CopNet_N) \tag{1}$$

The PageRank value of a node is a metric of node importance computed via a random walk algorithm. Conceptually, it defines the importance of a node as the probability of it being visited during a random traversal of the directed network. During this random walk, the probability of transitioning from one node to another, along a directed edge, is distributed proportionally to the weights of the outgoing edges. This random walk process can be modelled as a first-order Markov chain, and the components of its stationary distribution, PR_i, yield the PageRank values for the network nodes.

The importance of a process j belonging to task i is defined in (2), where *Priority* corresponds to the process priority defined subsequently in (3).

$$I_p(i_j) = Priority(i_j)$$

$$Priority = w_1 P_u^{\text{norm}} + w_2 P_s^{\text{norm}} + w_3 P_r^{\text{norm}}$$

The process priority is typically a composite measure, integrating factors such as the process's urgency (P_u), its static priority (P_s), and its process planning priority (P_r). This ensures that urgent tasks are commenced preferentially over routine ones (Qian et al., 2020). In contexts involving multiple processes, the urgency of the process P_u is determined by the proximity of the delivery deadline. Meanwhile, the static priority P_s is inherent to the process route itself. The process is assigned a higher P_s to those with more subsequent processes, calculated as the total processing time of all its descendant processes in the production route. This metric reflects the potential scale of disruption the current process could cause to subsequent stages. The planning priority stems from typical process planning knowledge. E.g., a common principle is "Shortest Job First", where processes with shorter processing times are assigned higher priority. This approach reduces the average waiting time, thereby preventing a backlog of short-duration processes from accumulating behind longer ones.

Synthesising these influencing factors, the process priority is given by (3), where P_u^{norm}, P_s^{norm}, and P_r^{norm} represent the normalised urgency, static priority, and planning priority, respectively. The coefficients w_i are weighting factors that sum to unity, i.e. $\sum w_i = 1$.

As a general guideline, w_1 should be appropriately increased in scenarios with strict delivery deadlines or tight schedules. For production systems with inherently long average process routes, a larger w_2 can help mitigate the cumulative effect of anomalies or delays. In situations where most process times are similar except for a few outliers with significantly longer durations, increasing can reduce the average process waiting time.

3.3. A Flexible Classifier for Machine-Process Matching Strategies

From the definitions above, the Machine Importance I_m reflects the current degree to which a machine is "needed" within the manufacturing network. A higher I_m indicates that the machine receives a relatively larger number of processing requests and holds greater importance for completing the recent tasks. Besides, the Process Importance I_p embodies the current "desire level" for machine capacity possessed by the process. A higher I_p signifies greater importance and urgency of the process, implying a larger impact on fulfilling customer demands. Based on the respective values of Machine Importance I_m and Process Importance I_p , this section establishes a flexible classifier for machine-process matching strategies, as listed below.

When a high-importance process selects a low-importance machine, the classifier adopts a "Process Dominated Strategy". This strategy prioritises maximising the fulfilment of the process's production requirements. The process can claim any available time slot on the machine, ensuring its demands are met as a primary objective.

Conversely, when a low-importance process selects a high-importance machine, the classifier implements a "Machine Dominated Strategy". This strategy focuses on optimising the resource utilisation of the critical machine. The process's production plan is scheduled to maximise the machine's usage rate, deliberately avoiding the creation of small, unusable time fragments between tasks that could degrade overall efficiency and waste the processing capacity of a high-value asset.

The aforementioned strategies can be realised by implementing different objective functions emphasising the machine utilisation or the completion time of process when scheduling. In all other cases, the classifier facilitates a "Cooperative Game Strategy" between the machine and the process. Based on their relative importance values, this strategy determines a suitable start time aimed at maximising the combined utility for both the process and the machine. The specific mechanics of this cooperative game process are detailed in Section 4.

4. The Cooperative Game Model for Machine-Process Scheduling

This section formalises the matching problem within a manufacturing network as a cooperative game between two primary players: a process and a machine. The core of their interaction is a negotiation concerning the process's start time on the machine.

4.1. Modelling the Time Slots of Machines

For a given machine m, its current available time slots are represented by a set of open intervals, as denoted in (4). Each interval is bounded by the completion time (TL) of the preceding process and the planned start time (TN) of the subsequent process on the machine. When a task (*Order*) searches the network for an available machine for its process p, a machine is included in the candidate set if any of its available slots meets the duration requirement $T_{order(p)}$ specified in (5).

$$AvailSlot_m = \{(TL_1, TN_1), (TL_2, TN_2), \dots, (TL_n, TN_n)\} \quad (4)$$

$$TN_i - TL_i \geq T_{Order(p)}, \; i = 1,2,\dots,n \quad (5)$$

The practical feasibility of a time slot also depends on the lead time required for preparatory work, such as logistics and setup of tools. Depending on whether the time between the current moment t and the slot's start TL_i is sufficient (denoted by state ξ_S) or insufficient (ξ_I) for these preparations, the actual, usable time slot $ActSlot_m$ available to the process is determined by (6). This calculation incorporates the process's latest start time $T_{LS(p)}$, the completion time of its predecessor $T_{Finish(p-1)}$, and the required logistical lead time $T_{log(p-1,p)}$.

$$ActSlot_m = \begin{cases} \left(\left(T_{\text{Finish}(p-1)} + T_{\text{log}(p-1,p)}\right), \min\left(TN_i - T_{Order(p)}, T_{\text{LS}(p)}\right)\right), & \xi_{\text{I}} \\ \left(TL_i, \min\left(TN_i - T_{Order(p)}, T_{\text{LS}(p)}\right)\right), & \xi_{\text{S}} \end{cases} \quad (6)$$

4.2. The Utility Functions for Machines and Processes

Based on the process's assigned weights for production time, cost, and quality, it determines its ideal start time $T_{Start(p)}$ within $ActSlot_m$ to meet its punctuality requirements and minimise inventory costs. On the other hand, the ideal start time for an incoming process, from the machine's perspective, is either at the very beginning of the available slot (TL_i) or at a point sufficiently close to its end (TN_i) to avoid partitioning the available time into short, unusable fragments that cannot accommodate other process. To achieve this objective, a floating pricing strategy is employed by machines as defined in (7) - (9). Here, *StdCost* is the standard price per unit time for the machine, t_p is the start time determined through the game, and the average processing time *avgProcLen* is estimated from the machine's recent processing history.

$$Cost_{Order(p)} = \begin{cases} StdCost \cdot T_{Order(p)}, & \Delta t \geq ProcLen \\ StdCost \cdot (T_{Order(p)} + \Delta t), & \text{Otherwise} \end{cases} \quad (7)$$

$$\Delta t = t_p - TL_i \quad (8)$$

$$ProcLen = \min\,(T_{Order(p)}, \; avgProcLen) \quad (9)$$

The process's utility function $P_{Order(p)}$ in the game is defined by (10) and (11). It represents the product's value minus processing costs and any additional costs arising from early or late production, such as inventory holding costs or penalties for late delivery. t_p denotes the process start time following the game, $T_{LS(p)}$ represents the latest start time for process p (i.e., the endpoint of interval $ActSlot_m$), W_p signifies the storage cost per unit time for this process, and Pen_p indicates the delay penalty rate for this process.

$$P_{Order(p)} = Value_{Order(p)} - Cost_{Order(p)} - Aux_{Order(p)} \quad (10)$$

$$Aux_{Order(p)} = \begin{cases} W_p\left(T_{\text{Start}(p)} - t_p\right), & t_p < T_{\text{Start}(p)} \\ 0, & T_{\text{Start}(p)} \leq t_p \leq T_{\text{LS}(p)} \\ Pen_p\left(t_p - T_{\text{LS}(p)}\right), & t_p > T_{\text{LS}(p)} \end{cases} \quad (11)$$

The machine's utility function P_m is defined by (12), which represents the revenue from the processing cost, minus an opportunity loss $Opploss_{Order(p)}$ incurred if the scheduled process creates a small, unusable time fragment. This loss quantifies the potential revenue foregone from other tasks that could have used that fragment. It reflects a system-wide perspective. The quantifiable opportunity loss is defined by (13) and (14). It models the scenario where a new process arrives during the newly created fragment $\Delta Slot$, and that new process selects this specific machine as a candidate. The probability of this event is a function of the probability of at least one new process arriving during $\Delta Slot$ (denoted as $Prob_{arr}$) and the probability that a random walk visits machine m at least once during that period (denoted as $Prob_{RW}$). This is scaled by the ratio of the expected importance of a newly arrived process to the importance of the current process.

$$P_m = Cost_{Order(p)} - Opploss_{Order(p)} \quad (12)$$

$$Opploss_{Order(p)} = Prob_{\text{arr}} Prob_{\text{RW}} \frac{E(I_p')}{I_p(Order(p))} Cost_{Order(p)} \quad (13)$$

$$\Delta Slot = TN_i - TL_i - T_{Order(p)} \quad (14)$$

The arrival of new processes at a machine is modelled as a Poisson process, characterised by a constant average arrival rate. Consequently, the number of processes arriving before time t follows a Poisson distribution with parameter , as shown in (15). The probability is then derived from the Poisson probability mass function, given by (16) and (17).

$$N(t) \sim Poiss(\lambda_a t) \quad (15)$$

$$Poiss(X = 0) = e^{-\lambda_a \Delta Slot} \quad (16)$$

$$Prob_{\text{arr}} = Poiss(X \geq 1) = 1 - e^{-\lambda_a \Delta Slot} \quad (17)$$

The probability $Prob_{RW}$ is determined by the structure of the directed cooperation network $CopNet_N$. As defined by the PageRank algorithms, the one-step transition probability from node i to node j is given by (18), where A_{ij} denotes the element of the adjacency matrix containing the edge weight, and K represents the out-degree of node i. The probability of the random walk visiting node m at least once during the *times* number of steps within $\Delta Slot$ is calculated by (19) to (21). Here, *times* is the number of processing slots that can theoretically fit into the fragment. $\{m\}$ represents all machine nodes in the network. $I_m(a)$ denotes the importance, or the PageRank value of machine a. The function ceil() performs rounding upwards.

$$Prob_m(X=1) = I_{\mathrm{m}}(m) + \sum_{a\in\{\mathrm{m}\}} I_{\mathrm{m}}(a)\, P_{\mathrm{Trans}}(a,m) \tag{19}$$

$$times = \mathrm{ceil}\left(\frac{\Delta Slot}{avgProcLen}\right) \tag{20}$$

$$\begin{aligned} Prob_{\mathrm{RW}} &= 1 - [1 - Prob_m(X=1)]^{times} \\ &= 1 - \left[1 - I_{\mathrm{m}}(m) - \sum_{a\in\{\mathrm{m}\}} I_{\mathrm{m}}(a)\, P_{\mathrm{Trans}}(a,m)\right]^{\mathrm{ceil}\left(\frac{\Delta Slot}{avgProcLen}\right)} \end{aligned} \tag{21}$$

With all components of the machine's utility function in (12) defined, the overall payoff for the cooperative game can be obtained in 4.3.

4.3. The Cooperative Game Model between Machines and Processes

The players in this game are the machine and the process of the task. The process start time t_p constitutes the various strategies employed, thereby rendering it a continuous variable game. The overall payoff for the cooperative game is formulated as the sum of the utilities of the machine and the process, resulting in (22). Note that the production value $Value_{Order(p)}$ is a constant for a given process-machine pair and thus does not affect the optimisation. When a dominant strategy exists within the decision space (i.e., a strategy optimal for all participants), this solution is selected as the outcome of the game. Otherwise, the Nash equilibrium point of the game is sought: a state where all participants' current strategies are optimal, and no single participant can unilaterally alter their strategy to increase the payoff. The Nash equilibrium solution for this game can be found by solving the constrained optimisation problem specified by (23) and (24), which seeks to maximise the overall payoff within the feasible time slot $ActSlot_m$.

$$Payoff = -Aux_{Order(p)} - Prob_{\mathrm{arr}} Prob_{\mathrm{RW}} \frac{E(I_p')}{I_{\mathrm{p}}(Order(p))} Cost_{Order(p)} \tag{22}$$

$$\text{minimise} \quad y = -Payoff(t_p) \tag{23}$$

$$\text{subject to} \quad t_p \in ActSlot_m \tag{24}$$

The single-variable optimisation problem itself is not complex, and numerous algorithms can be employed to solve it. The BFGS algorithm with the time complexity of $O(n^2)$ is adopted, which applies a Newton-like method approximating the Hessian matrix of the objective function (Dai, 2013). The solved t_p represents the start time of processing determined following the cooperative game between the machine and the process. When multiple Nash equilibrium points exist in the game problem, selecting a smaller start time may allow for greater tolerance in subsequent processing stages.

5. NUMERICAL EXPERIMENT

To validate the efficacy of the proposed machine-process cooperative game model, this section conducts a comparative analysis of resource allocation within a manufacturing network. We compared two approaches: a conventional method as described in Section 3.1, and a new method where machines and processes negotiate start times through our cooperative game model.

5.1. Experiment Setup

A numerical simulation was constructed using a scenario comprising 50 virtual machines and 500 virtual manufacturing tasks. The machine fleet consists of 10 lathes, 6 milling machines, 5 planing machines, 4 grinding machines, 10 drilling machines, 7 polishing machines, and 8 3D printers. Each machine is defined by a set of specific parameters, including its operable dimensional range, processing accuracy, spatial coordinates, and distinct processing speed, which correlates with a corresponding usage cost per unit time. The detailed parameters for the machine fleet are provided in Table 1.

To ensure a realistic testbed for resource matching, the numerical experiment utilises a set of common process routes and process combinations, denoted as *routeSet* in (25). The formulation of this set is consistent with established international and national standards, including ISO 14649, the Chinese standard GB/T 4863-2008, the German standard series DIN 8580 ff, alongside exemplary industrial processes. The digits 0 to 6 in *routeSet* represent lathes, milling, planing, grinding, polishing, drilling machines, and 3D printers, respectively.

$$\begin{aligned} routeSet = \{&(0,1,3,4), (2,1,3,4), (2,3,4), (2,0,3,4), (0,1,2,3), (3,4,6), \\ &(0,1,3,4,6), (6,0,1,3,4), (0,2,1,3,4), (1,4), (2,0,1,3), (5,6), (5,4), (6,0,1), (2,6,0), \\ &(0,1,4), (0,1), (0,0), (1,1), (2,1), (0,6), (3,4), (0,3), (3,6), (0,1), (5,6), (0,0), \\ &(0,1,3,4), (6,0,1), (0,0), (2,3), (0,6), (0,1,3), (0,1), (5,6), (5,6)\} \end{aligned} \tag{25}$$

Table 1: Parameters for the machine fleet in the experiments.

ID	Type	Machinable dimensions		Machining precision	Machine position		Unit cost	Processing speed
		Axial / X	Radial / Y		X	Y		
1	Lathe	1200	350	10	118.4	29.6	22	1
2	Lathe	1200	350	12	122.2	32.1	12	0.9
3	Lathe	2000	400	7	121	32.4	18	1.1
4	Lathe	2000	400	12	119.4	30.7	12	0.9
5	Lathe	2000	400	7	121.2	29.8	18	1.1
6	Lathe	2000	400	10	121.1	30.9	22	1
7	Lathe	2000	400	7	122.2	30.6	18	1.1
8	Lathe	2000	400	10	119.3	31.9	22	1
9	Lathe	2000	400	12	120.7	31.5	12	0.9
10	Lathe	2000	400	7	119.3	30.3	18	1.1
11	Milling	250	650	8	121	29.2	25	1
12	Milling	200	700	8	120.4	29.7	25	1
13	Milling	360	900	7	121.4	29.8	32	1.06
14	Milling	400	1200	8	120.7	30.5	25	1
15	Milling	360	900	11	122	32.3	18	0.9
16	Milling	360	900	11	121.8	30.4	18	0.9
17	Planing	640	320	8	120.3	32.6	18	1.05
18	Planing	640	320	8	118.6	30.3	18	1.05
19	Planing	800	450	11	120.8	30.4	15	1
20	Planing	1000	500	10	121.6	29.2	23	1
21	Planing	800	450	10	121.1	30.3	23	1
22	Grinding	320	152	8	122.5	30.4	16	1
23	Grinding	320	152	8	120.7	32	16	1
24	Grinding	500	260	8	121.9	28.6	25	1
25	Grinding	1000	320	6	122.5	29.8	29	1.08
26	Polishing	200	80	4	120.9	31.8	15	1
27	Polishing	200	80	4	118.9	28.5	15	1
28	Polishing	200	80	1	121.9	31.7	30	1.1
29	Polishing	450	250	4	119.8	32.3	18	1
30	Polishing	610	180	4	122.1	32.1	15	1
31	Polishing	610	180	4	121.9	32.6	15	1
32	Polishing	610	180	1	119.9	29.5	30	1.1
33	3D Printing	320	320	10	120.2	29.9	16	1
34	3D Printing	320	320	10	119.7	32.2	22	0.95
35	3D Printing	165	380	8	118	29.6	42	1.08
36	3D Printing	165	380	10	119.2	30.7	22	0.95
37	3D Printing	350	350	10	119.4	32.6	22	0.95
38	3D Printing	165	380	10	120.5	28.8	16	1
39	3D Printing	320	320	8	119.3	30.5	42	1.08
40	3D Printing	650	600	10	120.2	29.6	16	1
41	Drilling	440	6	13	118.5	31.9	20	1
42	Drilling	68	3	13	119.7	29.4	20	1
43	Drilling	440	6	10	118.8	29.9	45	1.2
44	Drilling	440	6	10	120.3	31	45	1.2
45	Drilling	440	6	13	118.8	31.4	20	1
46	Drilling	293	6	10	122	31.8	45	1.2
47	Drilling	68	3	13	122	29	20	1
48	Drilling	68	3	13	120.3	31.7	26	0.9
49	Drilling	293	6	10	119.2	32.1	45	1.2
50	Drilling	293	6	10	122.1	30.6	45	1.2

To ensure a realistic testbed for resource matching, the numerical experiment utilises a set of common process routes and process combinations, denoted as *routeSet* in (25). The formulation of this set is consistent with established international and national standards, including ISO 14649, the Chinese standard GB/T 4863-2008, the German standard series DIN 8580 ff, alongside exemplary industrial processes. The digits 0 to 6 in *routeSet* represent lathes, milling, planing, grinding, polishing, drilling machines, and 3D printers, respectively.

$$routeSet = \{(0,1,3,4), (2,1,3,4), (2,3,4), (2,0,3,4), (0,1,2,3), (3,4,6), (0,1,3,4,6), (6,0,1,3,4), (0,2,1,3,4), (1,4), (2,0,1,3), (5,6), (5,4), (6,0,1), (2,6,0), (0,1,4), (0,1), (0,0), (1,1), (2,1), (0,6), (3,4), (0,3), (3,6), (0,1), (5,6), (0,0), (0,1,3,4), (6,0,1), (0,0), (2,3), (0,6), (0,1,3), (0,1), (5,6), (5,6)\} \quad (25)$$

The 500 virtual manufacturing tasks were generated by randomly selecting a process route from the *routeSet* for each task. For each process within a route, specific requirements for parameters such as processing dimensions and accuracy were randomly generated, thereby defining the virtual production tasks. A key logical constraint was enforced during generation, i.e., for any process sequence, the workpiece dimensions required by a subsequent process must not exceed those of its predecessor (with exceptions made for 3D printing), and the specified accuracy must not degrade throughout the process.

Each manufacturing task is also characterised by additional attributes, including the time they are received by manufacturing systems, locations for raw material and delivery, delivery due time, standard processing time, and its specific weights for production time, cost, and quality. Weightings were assigned randomly. The time weighting was used to determine the allowable slack time for each process, which was then combined with estimated logistics duration to calculate the final delivery due date for the task. The 500 independent tasks generated for the experiment comprise a total of 1,326 individual processes. A selection of the generated process parameters is detailed in Table 2, where the six-tuple series in the "Processes" column denotes the sequence of processes, machining methods, machining dimensions (x, y), precision requirements, and the standard process durations.

5.2. The Comparative Experiment on Machine-Process Pairing with Different Strategies

The first process of each task arrives at the manufacturing network at its pre-defined release time. Subsequent processes initiate their resource requests upon the completion of their immediate predecessors. A comparative experiment is designed as follows.The control group employs the "capability-availability-efficiency" filtering mechanism described in Section 3.1,

Table 2: Details of tasks and processes in the experiments (Excerpt)

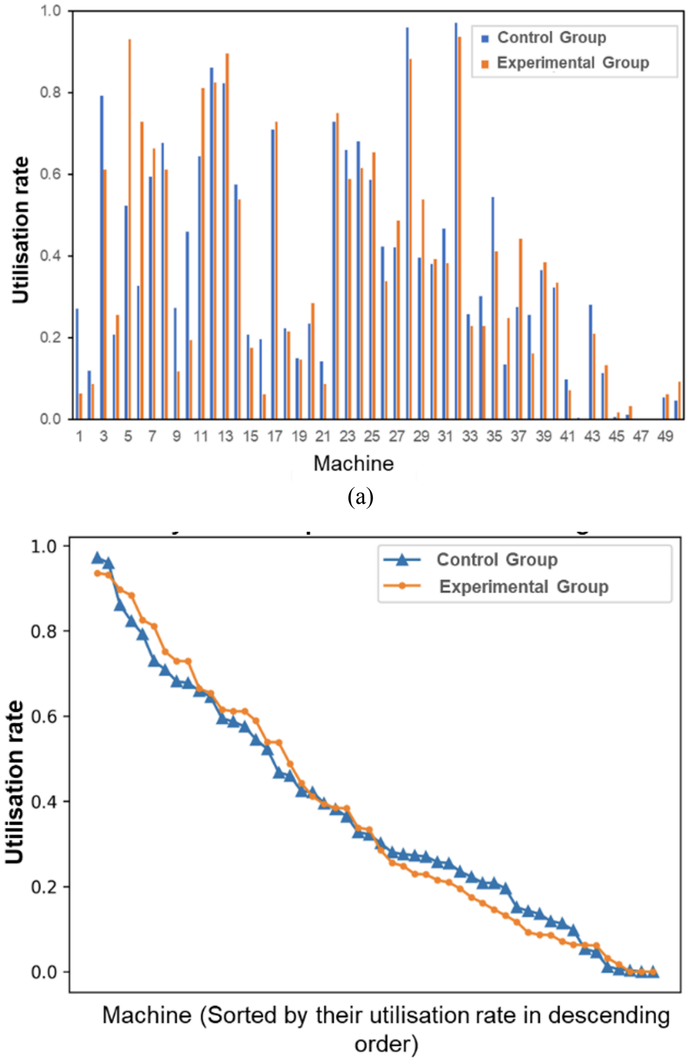
Task ID	Task arrival time	Raw material location (x, y)	Delivery location (x, y)	Delivery due time	Processes
0	18	(118.9, 30.3)	(119.1, 29.8)	412	(0, 'Lathe', 623, 281, 14, 171) (1, 'Milling', 604, 137, 11, 126)
1	26	(122.0, 29.5)	(120.2, 31.5)	296	(0, 'Lathe', 873, 114, 13, 139) (1, 'Lathe', 450, 87, 13, 86)
2	175	(122.0, 29.5)	(121.9, 31.2)	986	(0, 'Planing', 191, 212, 15, 159) (1, 'Lathe', 116, 189, 12, 94) (2, 'Milling', 56, 183, 12, 63) (3, 'Polishing', 55, 110, 5, 313)
3	112	(120.2, 29.3)	(118.0, 29.4)	387	(0, 'Drilling', 168, 4, 14, 32) (1, 'Lathe', 725, 299, 11, 79) (2, 'Milling', 666, 111, 11, 116)
4	11	(121.1, 30.5)	(122.0, 29.0)	1072	(0, 'Lathe', 823, 326, 13, 93) (1, 'Planing', 547, 78, 13, 167) (2, 'Milling', 252, 59, 12, 173) (3, 'Milling', 74, 52, 11, 113) (4, 'Polishing', 60, 51, 5, 193)
5	2	(122.0, 29.5)	(119.3, 30.5)	850	(0, 'Milling', 400, 327, 14, 123) (1, 'Polishing', 366, 122, 3, 356) (2, 'Drilling', 44, 3, 12, 26)
…	…	…	…	…	…
499	1752	(121.1, 30.5)	(120.7, 29.8)	916	(0, '3D Printing', 135, 268, 9, 714) (1, 'Drilling', 13, 4, 10, 26)

where the process autonomously selects its start time without considering the possible fragments caused at the machine timeline. In contrast, the experimental group implements the proposed game theoretic model, i.e., after filtering for machines of the available type, the process engages in a cooperative game process with each candidate machine over its available time slots to determine a mutually suitable start time. The importance of machine and the priority of processes are calculated dynamically based on $CopNet_{500}$, which denotes the collaborative network formed by up to 500 processes that have commenced recently. the storage cost per unit time W_p is defined as 1, while delay penalty rate Pen_p=15.

The experimental validation was conducted through a discrete-event simulation tool developed in Python. As the simulation time advanced from 0 to 10,000 time units, the 500 predefined manufacturing tasks were released into the system at their specified initiation times. The multi-agent system facilitated online scheduling, wherein the Technique for Order Preference by Similarity to Ideal Solution was employed by the agents to evaluate and rank candidate machines during the initial filtering phase based on the "capability-availability-efficiency" principle (Qian et al., 2019). During the simulation run, the *networkx* library in Python was utilised to dynamically compute the PageRank values for all machines within the evolving cooperation network $CopNet_{500}$. Key parameters for the game model (e.g., the Machine Importance I_m, the average operation duration per machine *avgProcLen*, and the process arrival rate λ_a) were updated in real-time, ensuring the model's responsiveness to the current network conditions. The simulation for both the control and experimental groups was conducted under identical task loads, parameters and machine properties, with different scheduling mechanisms incorporating either the game theory model or not. For the experimental group, once a set of candidate machines and their available slots were identified for a process, the cooperative game model was invoked. The negotiation to determine the optimal start time t_p was formulated as an optimisation problem, which was solved using the *scipy.optimize* module for Python to find the Nash equilibrium solution that maximised the joint payoff function. The weights to calculate the priority of process were defined as (w_1,w_2,w_3)=(0.3,0.45,0.25) in this experiment.

Fig. 1(a) illustrates the utilisation rate of each machine observed in both experimental runs. The machines are sorted along the horizontal axis in descending order of their utilisation rate in Fig. 1(b). The vertical axis represents the corresponding utilisation rate of each machine.

The average utilisation rate of all machines has barely changed for both the control and experimental groups, at 37.3% and 37.5%, respectively.



(a)

(b)

Figure. 1: The utilisation rate of machines in the comparative experiments showing a) Utilisation rate of machines in the comparative experiments, and b) Utilisation rate of machines sorted by their occupied times

However, Fig. 1(b) shows that, for machines with initially high utilisation rate (indicating bottleneck resources), applying the cooperative game model generally led to an increase in utilisation. Among the 25 machines with initial utilisation rates above the average, 19 showed improved utilisation, with an average increase of 1.5%. This localised improvement occurred while the overall average utilisation rate across the entire manufacturing network remained virtually unchanged. Fig. 1 demonstrated that the new method specifically boosts the utilisation of bottleneck resources, while maintaining the overall production pace and stability of the system. It indicated a redistribution of workload that enhances the throughput of critical machines without negatively overloading the system. Furthermore, all 1,326 processes were scheduled and processed successfully within the observed time horizon, confirming that the integration of the game model did not introduce detrimental delays or disrupt the overall production flow, thereby ensuring the stability of production management.

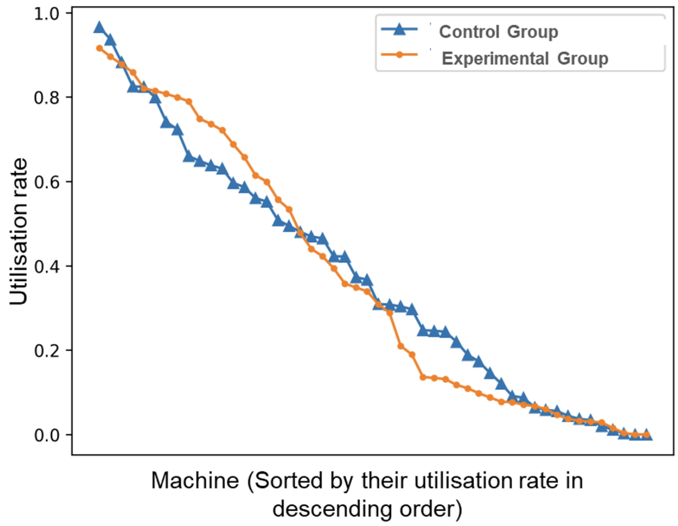
To assess the robustness of the experimental findings, a sensitivity analysis was conducted. Three additional sets of experiments were performed, each modifying a key parameter from the base case:

- Sensitivity test (a): The random seed was altered to generate a new set of 500 distinct manufacturing tasks.
- Sensitivity test (b): The inventory holding cost rate, i.e. in (11), was substantially increased to a new value of 10.
- Sensitivity test (c): The weighting parameters within the process priority calculation in (3) were modified as (w_1,w_2,w_3)=(0.6,0.25,0.15), to reflect a different decision-making bias.

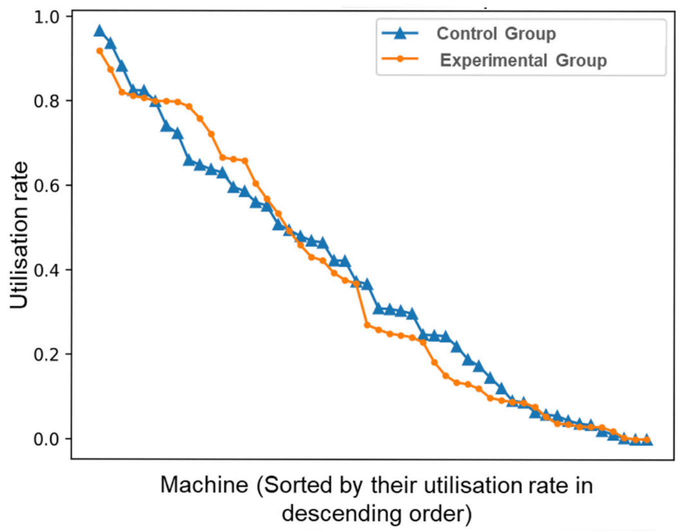
As shown in Fig.2, the results from all three sensitivity tests consistently demonstrated the same characteristic outcome observed in the initial experimental group. Specifically, the cooperative game model continued to facilitate a further increase in the duty cycle of identified bottleneck resources compared to the baseline scheduling mechanism. This recurring pattern, observed under varying conditions of task load, cost structure, and priority calibration, strengthens the assertion that the proposed model possesses a robust capability to enhance the utilisation of critical system resources without compromising overall scheduling feasibility.

6. CONCLUSIONS

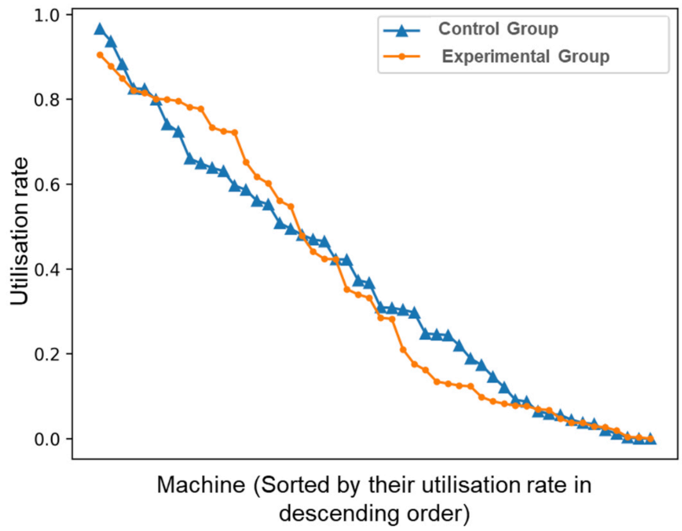
This paper addressed the complex challenge of allocating resources online in large-scale discrete manufacturing systems, where high throughput and high machine utilisation are conflicting objectives that require balancing. The study successfully addressed its core research questions. Firstly, it demonstrated how a cooperative game can be formally modelled between machine and process agents. The model establishes a framework for direct negotiation over processes' start times by defining specific utility functions that incorporate processing costs, earliness/tardiness penalties, and a novel opportunity loss cost for time fragmentation. Resolving this game to its Nash equilibrium yields schedules that balance the interests of both parties while addressing throughput and utilisation simultaneously. Secondly, this paper detailed how network science metrics, specifically PageRank centrality, can be integrated into the negotiation process among manufacturing resources. The dynamic importance of machines is a key parameter in quantifying the opportunity loss associated with schedule fragmentation. This integration provides a data-driven, systemic perspective that guides locally negotiating agents towards decisions that enhance global efficiency. Comparative experimental results have validated the model's practical efficacy in achieving a superior balance between throughput and



(a)



(b)



(c)

Figure. 2: The utilisation rate of machines in three sensitivity tests, showing a) utilisation rate of machines with a different set of tasks, b) utilisation rate of machines with a different cost parameter W_p, and c) utilisation rate of machines with a differenct weights for priority calculation utilisation compared to conventional online scheduling methods.

Several promising directions for future research can be identified. For example, the current bilateral negotiation model could be extended to a multilateral cooperative game, in which groups of machines and complex tasks negotiate simultaneously to optimise the entire process route. Furthermore, investigating collaboration patterns within and across functional clusters in the manufacturing network (e.g. milling cells and final assembly lines) could lead to greater efficiency. Finally, it would be beneficial to explore the integration of machine learning to adaptively tune the parameters of the utility functions in response to changing production dynamics.

References

Allahverdi, A., Pesch, E., Pinedo, M., & Werner, F. (2018). Scheduling in manufacturing systems: new trends and perspectives. *International Journal of Production Research*, *56*(19), 6333-6335.

Barenji, A. V., Barenji, R. V., Roudi, D., & Hashemipour, M. (2017). A dynamic multi-agent-based scheduling approach for SMEs. *The International Journal of Advanced Manufacturing Technology*, *89*(9), 3123-3137.

Bi, M., Kovalenko, I., Tilbury, D. M., & Barton, K. (2024). Dynamic distributed decision-making for resilient resource reallocation in disrupted manufacturing systems. *International Journal of Production Research*, *62*(5), 1737-1757.

Dai, Y. H. (2013). A perfect example for the BFGS method. *Mathematical Programming*, *138*(1), 501-530.

Gaafar, L. K., & Masoud, S. A. (2005). Genetic algorithms and simulated annealing for scheduling in agile manufacturing. *International journal of production research*, *43*(14), 3069-3085.

Hu, M., Qin, S., Wang, S., Zhang, J., & Ding, G. (2025). An energy-saving real-time scheduling method based on bi-level multi-agent architecture with bargaining game for flexible job shops. *Expert Systems with Applications*, *269*, 126527.

Jules, G., & Saadat, M. (2016). Agent cooperation mechanism for decentralized manufacturing scheduling. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *47*(12), 3351-3362.

Kang, K., Xu, S. X., Zhong, R. Y., Tan, B. Q., & Huang, G. Q. (2020). Double auction-based manufacturing cloud service allocation in an industrial park. *IEEE Transactions on Automation Science and Engineering*, *19*(1), 295-307.

Lee, J. H., & Kim, C. O. (2008). Multi-agent systems applications in manufacturing systems and supply chain management: a review paper. *International Journal of Production Research*, *46*(1), 233-265.

Leitão, P. (2009). Agent-based distributed manufacturing control: A state-of-the-art survey. *Engineering applications of artificial intelligence*, *22*(7), 979-991.

Li, Y., Tao, F., Cheng, Y., Zhang, X., & Nee, A. Y. C. (2017). Complex networks in advanced manufacturing systems. *Journal of Manufacturing Systems*, *43*, 409-421.

Liu, S., Zhang, L., Zhang, W., & Shen, W. (2021). Game theory based multi-task scheduling of decentralized 3D printing services in cloud manufacturing. *Neurocomputing*, *446*, 74-85.

Marzia, S., & Azab, A. (2023). Automated process planning and dynamic scheduling for smart manufacturing: A systematic literature review. *Manufacturing Letters*, *35*, 861-872.

Megow, N., Uetz, M., & Vredeveld, T. (2006). Models and algorithms for stochastic online scheduling. *Mathematics of Operations Research*, *31*(3), 513-525.

Minguillon, F. E., & Lanza, G. (2019). Coupling of centralized and decentralized scheduling for robust production in agile production systems. *Procedia CIRP*, *79*, 385-390.

Poudel, L., Elagandula, S., Zhou, W., & Sha, Z. (2023). Decentralized and centralized planning for multi-robot additive manufacturing. *Journal of Mechanical Design*, *145*(1), 012003.

Qian, C., Zhang, Y., Liu, Y., & Wang, Z. (2019). A cloud service platform integrating additive and subtractive manufacturing with high resource efficiency. *Journal of Cleaner Production*, *241*, 118379.

Qian, C., Zhang, Y., Jiang, C., Pan, S., & Rong, Y. (2020). A real-time data-driven collaborative mechanism in fixed-position assembly systems for smart manufacturing. *Robotics and Computer-Integrated Manufacturing*, *61*, 101841.

Sanogo, K., Benhafssa, A. M., & Sahnoun, M. H. (2025). A game theory approach for optimizing job shop scheduling problems with transportation in common shared human-robot environments. *Computers & Industrial Engineering*, 111366.

Sha, D. Y., & Lin, H. H. (2010). A multi-objective PSO for job-shop scheduling problems. *Expert Systems with Applications*, *37*(2), 1065-1070.

Uhlmann, I. R., & Frazzon, E. M. (2018). Production rescheduling review: Opportunities for industrial integration and practical applications. *Journal of manufacturing systems*, *49*, 186-193.

Veeramachaneni, V., Mallick, P. K., Rout, S., & Pareek, P. K. (2025). Efficient task scheduling for cloud environments with MSA: Enhancing makespan and resource utilization. In 2025 *International Conference on Emerging Systems and Intelligent Computing* (*ESIC*) (pp. 740-747). IEEE.

Wong, T. C., & Ngan, S. C. (2013). A comparison of hybrid genetic algorithm and hybrid particle swarm optimization to minimize makespan for assembly job shop. *Applied Soft Computing*, *13*(3), 1391-1399.

Xing, L., Dong, X., Guan, J., & Qiao, X. (2019). Betweenness centrality for similarity-weight network and its application to measuring industrial sectors' pivotability on the global value chain. *Physica A: Statistical mechanics and its applications*, *516*, 19-36.

Yang, M., Abubakar, A. H., & Jiang, P. (2021). Deep learning and complex network theory based analysis on socialized manufacturing resources utilisations and an application case study. *Concurrent Engineering*, *29*(3), 236-248.

Zan, X., Wu, Z., Guo, C., & Yu, Z. (2020). A Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems. *Advances in Mechanical Engineering*, *12*(1), 1687814019885294.

Zeng, L., Chen, H., Chen, M., & Zhao, X. (2024). Resilience assessment of the aircraft manufacturing core products supply chain: the international trade network perspective. *Annals of Operations Research*, 1-39.

Zhang, W., Bao, X., Hao, X., & Gen, M. (2025). Metaheuristics for multi-objective scheduling problems in industry 4.0 and 5.0: a state-of-the-arts survey. *Frontiers in Industrial Engineering*, *3*, 1540022.

Zhang, W. Y., Zhang, S., & Guo, S. S. (2017). A PageRank-based reputation model for personalised manufacturing service recommendation. *Enterprise Information Systems*, *11*(5), 672-693.

Zhang, Y., Qian, C., Lv, J., & Liu, Y. (2016). Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor. *IEEE Transactions on Industrial Informatics*, *13*(2), 737-747.

Zheng, P., Xia, L., Li, C., Li, X., & Liu, B. (2021). Towards Self-X cognitive manufacturing network: An industrial knowledge graph-based multi-agent reinforcement learning approach. *Journal of Manufacturing Systems*, *61*, 16-26.

Zhu, J. Y., Qin, W., Hu, J. H., Sun, Y. N., & Chen, Y. (2023). Influential process nodes identification strategy for aircraft assembly system based on complex network and improved PageRank. *Advanced Engineering Informatics*, *58*, 102187