

PETRI NET-BASED SCHEDULING OPTIMIZATION IN CNC-FMS USING GLOBAL ESTIMATION ENHANCED ANT COLONY OPTIMIZATION

Tuan Dung Nguyen^{1,*}, Anh Vang Tran¹, Quoc Viet Nguyen¹

¹Faculty of Aerospace Engineering, Le Quy Don Technical University

Abstract

This article investigates the scheduling optimization problem in a CNC-based flexible manufacturing system (FMS). A Petri net model is developed to accurately represent the dynamic, concurrent, and resource-sharing behaviors of the manufacturing system. Based on the reachability graph derived from the Petri net model, the scheduling problem is formulated as an optimal path search problem in the system state space. To efficiently explore this search space, an improved ant colony optimization (IACO) algorithm is proposed. The algorithm integrates a global estimation function, adaptive state transition rules, and a dual pheromone update mechanism to improve convergence performance and solution stability. Experimental results demonstrate that the proposed IACO algorithm significantly outperforms the classical ant colony optimization (ACO) algorithm. Specifically, the IACO reduces the average number of iterations required for convergence from 85.2 to 12.4 and achieves the optimal solution in all 20 independent runs, compared to only 15 runs for the classical ACO. The average makespan is reduced from 1256.8 to 1203.2 time units. Furthermore, in larger-scale production scenarios, the IACO achieves significant makespan reductions compared to the actual factory baseline, while also significantly reducing the convergence time to reach the optimal schedule relative to the number of iterations required by the classical ACO. These results confirm that the proposed IACO framework provides a novel and effective solution for large-scale FMS scheduling by uniquely combining Petri net reachability analysis with a global estimation-enhanced ant colony optimization, thereby addressing the limitations of local heuristic-based methods.

Keywords: *FMS; optimal scheduling; Petri net; improved ant colony optimization.*

1. Introduction

With the rapid development of Industry 4.0 and intelligent manufacturing technologies, flexible manufacturing systems (FMS) have become a core production paradigm for modern CNC-based manufacturing environments. By integrating multiple CNC machines, automated material handling systems, and centralized control architectures, FMSs enable high product variety, improved responsiveness to market demand, and enhanced resource utilization efficiency [1]–[3]. However, the overall

* Corresponding author, email: dungnt@lqdtu.edu.vn
DOI: 10.56651/lqdtu.jst.v21.n2.1136

performance of an FMS is strongly influenced by its production scheduling strategy, which determines the sequencing and allocation of shared resources under complex technological and operational constraints [4].

The scheduling problem in CNC-based FMS is widely recognized as an NP-hard combinatorial optimization problem involving alternative routing options, precedence constraints, limited machine availability, and synchronization of auxiliary resources such as robots and buffers [5], [6]. As the scale and complexity of FMSs increase in real industrial environments, classical exact optimization approaches become computationally infeasible. Consequently, recent research has increasingly focused on intelligent and metaheuristic-based scheduling methods capable of producing high-quality solutions within acceptable computational times [7], [8].

In recent years, significant advances have been made across multiple research streams. In the domain of Petri net-based formal modeling, He *et al.* proposed a scheduling method based on place-timed Petri nets and basis reachability graphs, which compresses the state space and applies an improved beam search to handle large-scale FMS [9]; this method was later extended to incorporate a cost-function-driven transition selection strategy to further improve makespan [10]. In the domain of hybrid ACO algorithms, He *et al.* proposed a hybrid ACO integrating a two-element vector node representation for simultaneous scheduling of production and transportation tasks in FMS [2]; Jin *et al.* introduced a hybrid ACO (HACO) with queue-based warm start initialization and parallel local search, achieving 5–10 speedup over state of the art solvers while maintaining a 3–5% optimality gap on flexible job shop instances [11]; and symmetry aware adaptive ACO (SA-AACO) was recently developed for integrated process planning and scheduling, demonstrating 29.1% makespan improvement over comparative algorithms [12]. Lassoued *et al.* combined Colored-Timed Petri Nets with actor-critic reinforcement learning for dynamic AGV and tool sharing optimization in FMS [1].

Despite these advances, a clear research gap remains. First, although Petri nets are used for modeling, most existing ACO-based scheduling methods lack a global estimation mechanism that accounts for system-level bottlenecks and remaining workload; they rely predominantly on local heuristic information and pheromone feedback [13], [14]. Second, many studies emphasize algorithmic enhancements while relying on simplified system representations, which may restrict applicability to real CNC-based FMSs with complex machine–robot interactions [15], [16]. Third, few studies explicitly integrate the reachability graph of the Petri net with ACO in a way that leverages a forward-looking estimation function to guide transition selection. Consequently, existing methods often

suffer from myopic decision-making, slow convergence, and suboptimal makespan, especially in large-scale FMS environments where resource congestion is critical.

To address the above limitations, this study proposes an integrated scheduling framework that uniquely combines Petri net-based formal modeling with an Improved Ant Colony Optimization (IACO) algorithm tailored for CNC-based FMS. The key novelty lies in the introduction of a global completion-time estimation function, which provides an optimistic lower bound of the remaining makespan from any marking. By incorporating the reciprocal of this estimate into the transition probability, ants are guided toward globally promising schedules rather than locally attractive ones. This mechanism explicitly addresses the lack of global bottleneck awareness in previous ACO variants. In addition, the IACO integrates an adaptive state transition rule with dynamic and a dual pheromone update strategy (local and global) to balance exploration-exploitation and prevent premature stagnation. By leveraging the accurate system representation provided by Petri nets and the global search capability of the improved ACO algorithm [17], [18]–[20], the proposed method aims to achieve faster convergence, improved solution quality, and enhanced scalability for real-world FMS scheduling applications.

The main contributions of this study can be summarized as follows: (1) A formal Petri net model is developed to represent the structural and dynamic characteristics of a CNC-based FMS. The reachability graph of the Petri net is used to transform the scheduling problem into a structured optimal path search problem; (2) An IACO algorithm is proposed, incorporating an enhanced heuristic function, a global completion-time estimation mechanism, and adaptive state transition rules to guide the search process toward globally promising scheduling solutions; (3) A dual pheromone update mechanism combining local and global updates is introduced to balance exploration and exploitation, thereby improving convergence speed and preventing premature stagnation; (4) The effectiveness of the proposed method is validated through scheduling experiments at different production scales, demonstrating improved convergence performance and reduced makespan compared with the classical ACO algorithm.

The remainder of this article is organized as follows: Section 2 reviews the scheduling methods for FMS. Section 3 presents the Petri net modeling of the FMS under study. In Section 4, the Petri net-based formulation of the scheduling problem is described. Section 5 introduces the IACO algorithm, including the basic ACO for Petri net scheduling and the proposed improvement strategies. Section 6 reports and discusses the experimental results. Finally, Section 7 concludes the article.

2. Scheduling methods for flexible manufacturing system

The core challenge in operating a FMS lies in production scheduling determining the optimal sequence and allocation of operations across shared, reconfigurable resources like CNC machines and automated material handlers. In systems akin to the studied CNC-based cell, this involves processing multiple part types with alternative machining routes, subject to stringent precedence and resource constraints. The primary objective is typically to minimize the total completion time (makespan), a critical metric for throughput and resource utilization. This scheduling problem is classified as NP-hard, meaning solution complexity grows exponentially with problem size, rendering exact mathematical methods like Branch and Bound or Integer Programming intractable for real-world, large-scale instances. Consequently, research has pivoted toward robust metaheuristic algorithms capable of finding high quality, feasible solutions within reasonable computational time.

Petri nets provide a powerful formal tool to model, analyze, and visualize the discrete, concurrent, and dynamic nature of FMS scheduling. A Petri net is mathematically defined as a five-tuple $PN = (P, T, F, W, M_0)$, where P is a finite set of places, denoted by circles in the Petri net, representing system states such as machine idle or workpiece waiting in a buffer; T is a finite set of transitions, denoted by rectangles, representing events such as the start of a milling operation or a robot pickup; $F \subseteq (P \times T) \cup (T \times P)$ is a set of directed arcs defining the flow relations. $W : F \rightarrow N^+$ is a weight function assigned to arcs; and M_0 is the initial marking that specifies the initial distribution of tokens across places. The system state is denoted by a marking M , a distribution of tokens. its representing available parts or resources across places. The firing of an enabled transition changes the marking, simulating the flow of jobs through the system. For FMS, this formalism offers distinct advantages: it graphically encapsulates concurrency, resource conflict, and synchronization. More importantly, it enables the automatic generation of a reachability graph, which exhaustively represents all possible system states and their transitions. Consequently, the scheduling problem is transformed into a search for an optimal path from the initial to the final marking in this state space, resulting in an optimal sequence of transition firings to minimizing the makespan.

Ant colony optimization (ACO) is a metaheuristic inspired by the foraging behavior of ants, renowned for solving complex combinatorial optimization problems like scheduling. Artificial "ants" probabilistically construct solutions based on pheromone trails and heuristic information. This positive feedback mechanism enables efficient

exploration of vast solution spaces, such as the reachability graph of a Petri net, to find high quality schedules.

However, the standard ACO algorithm exhibits well documented limitations when applied to complex FMS problems: slow convergence, a tendency to get trapped in local optima, and stagnation. These shortcomings necessitate problem specific enhancements designed to enhance solution quality, accelerate convergence, and increase the algorithm's robustness for dynamic production environments, forming the basis for the enhanced ACO proposed in this work.

3. Petri net modeling of the flexible manufacturing system

This study investigates a real-world FMS implemented in a mechanical manufacturing plant, as shown in Fig. 1. The research focuses on the fabrication process of three specific parts belonging to a single assembly, with uniform production volumes across batches. These components are processed in FMS integrating four CNC units: two 3-axis CNC lathes (FTC 20L) and two vertical machining centers (MVC 610 and MVC 850). A robot is deployed to travel along a conveyor, performing the tasks of supplying raw blanks and transporting in-process workpieces along the technological workflow, then delivering the fully machined components to the finished goods storage.

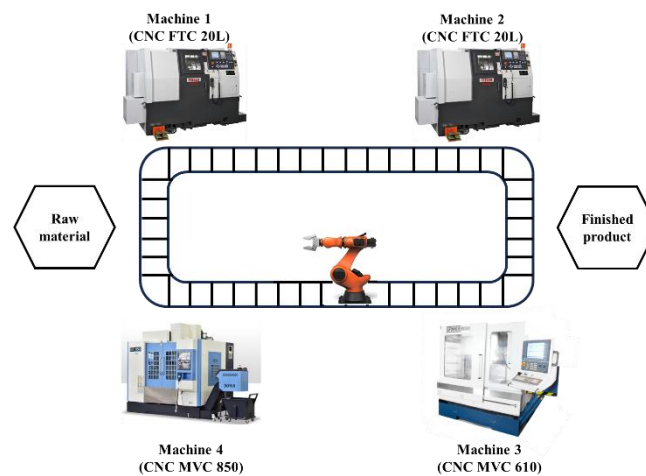


Fig. 1. Layout of FMS.

To maximize productivity while adhering to stringent technical standards and machining parameters, the facility has distributed operations across different machines based on auxiliary factors such as specific tooling, specialized fixtures, and dedicated coolants. However, current improvements are limited to the optimization of individual unit processing times. The core unresolved challenge lies in establishing a flexible scheduling sequence among equipment to enable the concurrent machining of multiple

parts, thereby maximizing the Overall Equipment Effectiveness (OEE) of the entire line. This constitutes the production scheduling problem that this research aims to solve by identifying an optimal operational strategy for the FMS.

The system is tasked with the production of three specific job types: J_1 , J_2 , and J_3 . The detailed process routes and the associated processing times for these jobs are delineated in Fig. 2. For instance, as illustrated in the Fig. 2, J_1 requires execution through three sequential processing steps; the initial operation must be performed on machine M_1 and necessitates 165 units of time. All machines are subject to the constraint of processing only one workpiece concurrently. Similarly, the robot is responsible for the inter-machine handling of workpieces and is restricted to transporting a single workpiece at any given time.

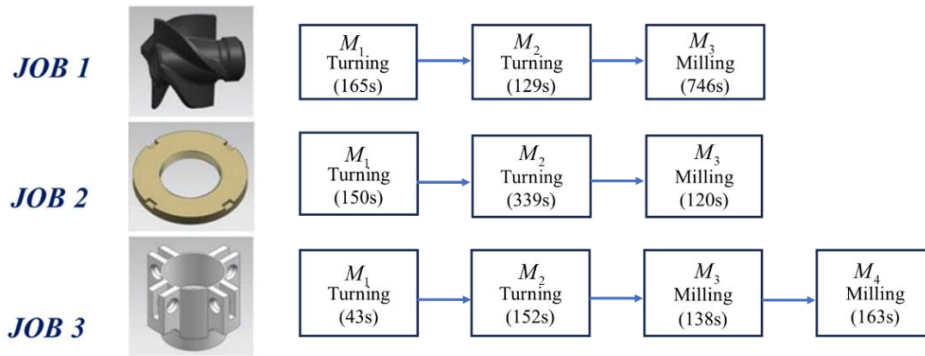


Fig. 2. Routing and processing times of jobs.

The construction of the Petri net model for the FMS in this article employs a top-down design approach. Initially, the system is known to comprise three processing jobs. Based on the sequence of processes and the resource requirements for each job, a separate Petri net sub-model is constructed for every job. In these sub-models, each operation within a job is represented as an operation transition in the Petri net system. Corresponding input places and output places, as well as the relevant resource places, are assigned to each transition. In Fig. 3, the Petri net consists of 13 places, places P_1, P_2, P_3 represent the start of each job (raw workpieces), while $P_{11} - P_{13}$ denote finished parts. Intermediate places $P_4 - P_{13}$ represent partial completion states. Resource places $R_1 - R_4$ model the four CNC machines, each initially marked with one token indicating availability. The set $T = \{T_1, T_2, T_3 \dots T_{10}\}$ contains 10 transitions, each representing a specific machining or handling operation. The initial marking M_0 places one token in

each $P_1 - P_3$ and one token in each resource place $R_1 - R_4$. The final marking M_f is reached when tokens appear only in $P_{11} - P_{13}$ and all resource places are restored.

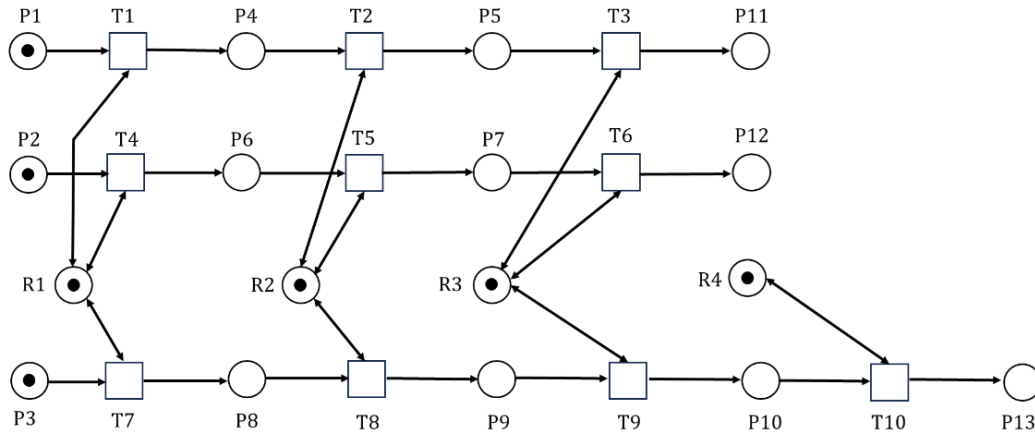


Fig. 3. Petri net.

4. Petri net-based scheduling problem formulation

The scheduling problem is formally defined based on the constructed Petri net model of the FMS. From this Petri net, a Reachability Graph is systematically generated, enumerating all possible system states (markings) and the transitions between them. Each path from the initial state to the final state in this graph represents a feasible sequence of operations. Thus, the scheduling objective is formulated as an optimal path search problem within the reachability graph. The goal is to identify a path, a sequence of transition firings from the initial marking to the target marking, while minimizing a performance metric, typically the makespan. The use of the reachability graph enables structural analysis such as detecting conflicts and deadlocks, and provides a complete state space for solving this complex combinatorial scheduling problem.

The reachability graph of a Petri net can describe the relationship between system reachable markings and transition firings. The firing of a transition allows the Petri net to move from one state (marking) to another, and the reachability graph visually and intuitively illustrates the set of all possible firing sequences and reachable markings starting from the initial marking. In this Petri net, we define the marking vector as the set of tokens distributed across the places $P_1, P_2, \dots, P_{13}$ as well as R_1, R_2, R_3 , and R_4 . The initial marking and the terminal marking are respectively specified as $M_0 = [1, 1, 1, 0, \dots, 0, 1, 1, 1]$ and $M_f = [0, \dots, 0, 1, 1, 1, 1, 1]$, and the resulting simplified reachability graph of the system is illustrated in Fig. 4.

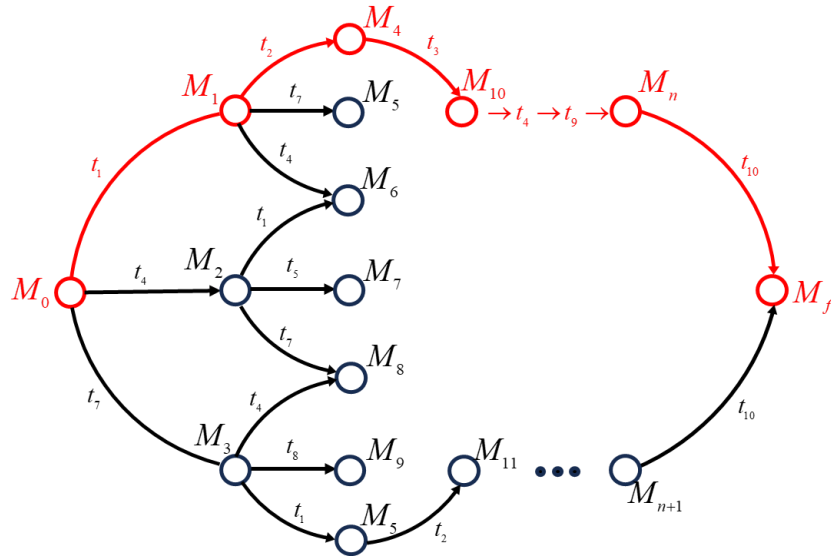


Fig. 4. The reachability graph of a Petri net.

When using Petri nets to solve the scheduling problem in FMS, from the perspective of Petri nets, scheduling is the process of optimizing the path to the terminal marking with the objective of minimizing the total completion time and constrained by the reachable markings, and ensuring compliance with the prescribed technological machining sequence of FMS. Theoretically, enumerating all transition firing sequences in the reachability graph can yield an optimal solution. However, searching the entire reachability graph is a problem with exponentially increasing complexity. Therefore, an optimization algorithm is required to search a portion of the reachability graph to obtain an optimal or near-optimal scheduling solution within a reasonable time.

Based on the reachability graph of the Petri net model, the scheduling problem is reformulated as a single-objective optimization problem that seeks to minimize the total completion time (makespan).

Let $\sigma = t_1 t_2 \dots t_n$ be a feasible transition firing sequence from the initial marking M_0 to the final marking M_f . Denote by $p(t_k)$ the processing time of transition t_k , and by $\omega(t_k, M_{k-1})$ the waiting time before firing t_k due to resource unavailability or precedence constraints, where M_{k-1} is the marking just before firing t_k . The makespan of σ is then defined as:

$$C_{\max}(\sigma) = \sum_{k=1}^n (p(t_k) + \omega(t_k, M_{k-1})) \quad (1)$$

The scheduling objective is to find a feasible sequence σ^* that minimizes the makespan:

$$\sigma^* = \arg \min_{\sigma \in S} C_{\max}(\sigma) \quad (2)$$

where S is the set of all feasible firing sequences from M_0 to M_f .

5. Improved ant colony optimization algorithm

5.1. Basic ACO for Petri net scheduling

Observational studies on ant foraging behavior have revealed that ants possess severely limited visual acuity. Nevertheless, irrespective of the distance or complexity of the path connecting the food source to the colony, ants are capable of gradually optimizing multiple initially utilized transport routes, eventually converging on a single shortest path. This pathfinding capability is achieved through the deposition of a volatile chemical substance, pheromone, which marks the trail. The pheromone intensity acts as a positive feedback mechanism, attracting subsequent ants to reinforce frequently used paths, while less-utilized trails gradually dissipate. This collective optimization process enables the colony to identify the shortest path between the food source and the nest, demonstrating the effectiveness of swarm intelligence despite individual limitations. To mathematically model the probabilistic decision-making process of ants during path selection, the transition probability of ant k moving from node i to node j at time t is defined as follows:

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{j \in allowed_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta} & j \in allowed_k \\ 0 & else \end{cases} \quad (3)$$

where $p_{ij}^k(t)$ denotes the probability that ant k moves from node i to node j at time step t ; $allowed_k$ represents the set of feasible nodes that ant k is allowed to visit next, considering technological and resource constraints; $\tau_{ij}(t)$ is the pheromone intensity on the path between nodes i and j at time t ; $\eta_{ij}(t)$ denotes the heuristic desirability of moving from node i to node j , which is typically defined as the inverse of the distance or processing time; α and β are the weight coefficients controlling the relative importance of pheromone information and heuristic information, respectively.

The study proposes a hybrid approach integrating Petri nets and the ACO algorithm to address scheduling problems in FMS. From a Petri net perspective, the scheduling problem is transformed into searching for an optimal transition firing sequence that minimizes the makespan within the reachable state space. The Petri net is utilized to model the FMS, where each marking represents a distinct production stage. Instead of exhaustively searching the entire reachability graph, which is computationally

exponential, the ACO algorithm is employed to efficiently explore a subset of paths, constructing a state tree composed of reachable markings. Consequently, this method yields an optimal or near-optimal schedule for the job sequence within a reasonable computational timeframe.

5.2. Toward an improved ant colony optimization algorithm for FMS scheduling

Although applicable to basic FMS scheduling, the standard ACO algorithm suffers from slow convergence and a tendency to become trapped in local optima, leading to solution stagnation. To address these issues, this study enhances the algorithm's parameter settings, informed by improved variants from [12], [15]–[17] and tailored to the specific properties and Petri net characteristics of the FMS. This adaptation aims to achieve scheduling solutions with superior performance indicators.

• Improvement of the heuristic function

In the ant colony algorithm, the heuristic information value η_{ij} represents the desirability of an ant transitioning from transition i to transition j . This value indicates, after the FMS allocates resources to the operation represented by transition i , the expected desirability of allocating resources next to the operation represented by transition j . Since the FMS scheduling process must satisfy both the precedence constraints of job operations and resource constraints, an operation can only begin processing when all its predecessor operations have been completed and the required machine resources are idle. Considering the associated waiting times, the reciprocal of an operation's earliest possible completion time is selected as its heuristic information. Its calculation method is as shown in Eq. (2):

$$\eta_{ij} = \begin{cases} \frac{1}{RT_{j-1} + T_j} & \text{if } RT_{j-1} > JT_{j-1} \\ \frac{1}{JT_{j-1} + T_j} & \text{else} \end{cases} \quad (4)$$

where T_j is the processing time of the operation represented by transition j ; RT_{j-1} is the earliest idle time of the machine resource required for the operation represented by transition; JT_{j-1} is the completion time of the immediately preceding operation of the operation represented by transition j .

• Integration of an estimation function

To overcome the myopic nature of the basic ant colony algorithm in FMS scheduling, a global estimation function is introduced to provide an optimistic lower-

bound estimate of the remaining makespan from the current marking M to the final marking M_f . The estimation function is defined as follows:

$$f_r(M) = \max\{h_{r_k}(M) + ts(r_k)\} \quad k = 1, 2, \dots, N_r \quad (5)$$

where $h_{r_k}(M)$ denotes the optimistic remaining processing time of all operations that still require resource r_k under marking M , computed by ignoring resource conflicts and assuming uninterrupted processing; $ts(r_k)$ represents the earliest idle time at which resource r_k becomes available; and N_r is the total number of shared resources.

The maximum operator reflects the fact that the overall makespan is ultimately determined by the most heavily loaded resource. Therefore, the resource with the largest sum of remaining workload and earliest availability provides a valid lower-bound estimate of the system completion time.

By incorporating $[1/f_r(M)]^\beta$ into the transition probability formula, the heuristic guides ants to favor choices that lead to a smaller estimated global completion time. This mechanism enables proactive decision-making, allowing the algorithm to assess the long-term impact of each scheduling step. Consequently, the search is directed toward the global optimum, emphasizing minimization of the makespan, while inherently promoting resource load balancing by discouraging decisions that would excessively burden critical machines. This approach enhances the algorithm's ability to avoid local optima and achieve more efficient scheduling solutions.

Therefore, an improved transition probability is proposed by incorporating the global estimation function into the state transition rule. Let M_j denote the marking obtained after firing transition j . The probability that ant m selects transition j at the k -th search step is defined as follows:

$$p_{ij}^m(k) = \begin{cases} \frac{[\tau_{ij}(k)]^\alpha [\eta_{ij}(k)]^\beta [1/f_r(M_j)]^\nu}{\sum_{j \in Tc_m^k} [\tau_{ij}(k)]^\alpha [\eta_{ij}(k)]^\beta [1/f_r(M_j)]^\nu} & j \in Tc_m^k \\ 0 & j \notin Tc_m^k \end{cases} \quad (6)$$

where Tc_m^k denotes the set of enabled transitions available to ant m at the k -th search step; $\tau_{ij}(k)$ is the pheromone intensity associated with transition j ; $\eta_{ij}(k)$ represents the local heuristic desirability; M_j is the marking reached after firing transition j from the current marking; $f_r(M_j)$ is the global estimation function defined in Eq. (3), which

provides an optimistic estimate of the remaining makespan from marking M_j ; and ν is the weighting coefficient controlling the influence of the estimation term.

According to above equations, a smaller estimated remaining completion time after firing transition j leads to a larger value of $1/f_r(M_j)$, thereby increasing the probability of selecting transition j . This mechanism enables the ant to favor transitions that are more promising from a global, system-level perspective rather than relying solely on local heuristic information.

• *State transition rule*

In the scheduling algorithm, an ant selects the next operation to be processed from a given state according to a specific state transition mechanism to proceed to the next node. This study employs the pseudo-random proportional rule from the ant colony system for selecting the next operation, as expressed in Eq. (5). The ant colony system incorporates a fixed parameter q_0 where $0 \leq q_0 \leq 1$, which represents the relative importance of an ant selecting the next operation based on prior knowledge versus random selection. However, whether the choice is determined by prior knowledge or made randomly based on transition probability, an ant should exhibit different tendencies during the initial and subsequent stages of the algorithm. Therefore, q_0 must also adaptively adjust as the algorithm progresses. In the initial stage, exploration of the search space should be prioritized, and thus q_0 is set to a smaller value. In later stages, after ants have largely traversed the available paths, exploitation and convergence to the optimal path should be accelerated, requiring q_0 to be set to a larger value.

$$j = \begin{cases} \arg \max_{u \in allowed_k} \left\{ [\tau_{iu}(k)]^\alpha [\eta_{iu}(k)]^\beta [1/f_r(M_u)]^\nu \right\}, & \text{if } q \leq q_0 \\ p_{ij}^m(k), & \text{other} \end{cases} \quad (7)$$

Based on the above considerations, q_0 is defined by Eq. (6):

$$q_0 = \frac{\log_e(NI)}{\log_e(NI_{max})} \quad (8)$$

where NI is the current iteration index and NI_{max} is the maximum number of iterations. According to Eq. (6), for low NI values, q_0 is close to 0. It gradually increases as NI increases and approaches 1 toward the end of the iterations.

During the solution construction process, when an ant selects the next operation to be processed, it first generates a random number q uniformly distributed in $[0, 1]$. If $q < q_0$, the ant selects the operation with the highest probability based on prior knowledge, as determined by Eq. (5). Otherwise, the selection probability for each transition in the set of enabled transitions is calculated according to Equation (4), and an operation is chosen using the roulette wheel selection method.

• *Pheromone update strategy*

To enhance the algorithm's search efficiency and convergence, this study introduces a dual-strategy pheromone update mechanism combining local and global updates.

- *Local update rule*

Applied immediately after an ant selects a transition, this rule prevents path saturation and encourages exploration by reducing the attractiveness of recently used paths, thereby diversifying the search within an iteration. The update is performed as:

$$\tau_{ij}(k+1) = (1-\zeta)\tau_{ij}(k) \quad (9)$$

where ζ denotes the local pheromone evaporation coefficient, with $\zeta \in (0,1)$.

- *Global update rule*

Applied after all ants complete their schedules per iteration, this rule reinforces only the paths belonging to the iteration-best or global-best solution, guiding the colony towards high-quality regions. The update is defined as:

$$\tau_{ij}(k+1) = (1-\rho)\tau_{ij}(k) + \Delta\tau_{ij}(k) \quad (10)$$

$$\Delta\tau_{ij}(k) = \begin{cases} \frac{Q}{L_{ib}} & \text{if } \text{arc}(i, j) \in BPath \\ 0 & \text{else} \end{cases} \quad (11)$$

where ρ represents the global pheromone evaporation coefficient, Q is a positive constant denoting the total pheromone amount deposited by each ant after obtaining the scheduling solution, $\Delta\tau_{ij}$ denotes the pheromone increment between nodes i and j , $BPath$ refers to the optimal transition sequence obtained in the current iteration, and L_{ib} represents the total completion time corresponding to that optimal transition sequence.

This objective function is used to evaluate every candidate schedule constructed by an ant. The optimization process is performed by the IACO algorithm described in Tab. 1.

Tab. 1. Flowchart of the proposed IACO algorithm for FMS scheduling

Step 1: Initialize parameters: $\alpha, \beta, \gamma, \rho_local, \rho_glocal, Q, m, N_{max}$
Step 2: Build Petri net model and generate reachability graph
Step 3: Initialize pheromone trails $\tau_{ij}(0)$
Step 4: For iteration $iter = 1$ to N_{max}
Step 4.1: For each ant $k = 1$ to m :
- Start from initial marking M_0
- While not reached final marking M_f :
+ Select next transition using adaptive pseudo-random rule (Eq. 5-6)
+ Perform local pheromone update (Eq. 7)
+ Fire transition $\rightarrow$ update marking
- Store complete schedule and compute makespan
Step 4.2: Identify iteration-best solution
Step 4.3: Perform global pheromone update (Eq. 8-9)
Step 4.4: If convergence criterion met then break
Step 5: Output optimal transition sequence and minimum makespan

6. Results and discussion

The superiority of IACO over basic ACO stems from three complementary mechanisms. (i) Global estimation function provides a forward looking lower bound of remaining makespan. By incorporating $1/f_r(M_j)$ into the transition probability, ants avoid locally attractive but globally poor choices, effectively balancing machine loads and preventing bottleneck congestion. (ii) Adaptive state transition with dynamic q_0 automatically shifts from early iterations to later iterations, ensuring both diversity and convergence. (iii) Dual pheromone update discourages premature path saturation while reinforcing only the best solutions each iteration. Together, these mechanisms yield faster convergence, higher solution quality, and exceptional stability.

The performance of the proposed improved ACO algorithm is primarily governed by three core parameters: α , β , and ν . The pheromone weight coefficient, α dictates the reliance on historical path information; while higher values accelerate convergence, they increase the risk of premature stagnation. The heuristic coefficient β determines the influence of local optimality, where a larger β speeds up convergence but may trap the algorithm in local optima. Furthermore, the article introduces a predictive function $f_t(M)$ with a weight coefficient ν , reflecting the ants' dependency on estimated

completion times. While a higher ν enhances convergence speed, it may lead to missing the global optimum. Balancing these parameters is essential to maintaining the equilibrium between global exploration and local exploitation.

Table 2 reports the parameter settings for both the basic ACO and the proposed IACO. The number of ants $N_{ant} = 30$ is kept identical for fair comparison, while the maximum number of iterations is reduced from 100 (basic ACO) to 50 (IACO). This reduction is intentional because IACO converges much faster; a larger iteration limit would be wasteful. The pheromone weight $\alpha = 1$ is the same for both, ensuring that the influence of historical trails is balanced. However, the heuristic weight β is increased from 0.1 (basic ACO) to 0.5 (IACO). This reflects that IACO benefits from more accurate heuristic information, thus assigning it higher importance. The introduction of the estimation weight $\nu = 0.5$ in IACO gives moderate influence to the global estimation function, preventing over-reliance that might cause premature convergence. The local evaporation coefficient $\zeta = 0.1$ is newly added in IACO to encourage exploration within each iteration, while the global evaporation coefficient ρ is reduced from 0.3 to 0.2 to preserve more historical information from good solutions. These parameter choices are not arbitrary; they were determined through systematic preliminary experiments and are consistent with recommendations in the previous research.

Tab. 2. Parameter settings

Parameter	Basic ant colony algorithm	Improved ant colony algorithm
N_{ant}	30	30
NI_{max}	100	50
Q	80	100
α	1	1
β	0.1	0.5
ν	–	0.5
ζ	–	0.1
ρ	0.3	0.2

Figure 5 shows the convergence curves of the basic ACO and the proposed IACO for the 3-job instance (optimal solution over iterations). When using the same number of ants for the search, the improved ACO algorithm converges to the optimal solution in an average of 6 iterations, compared to the 42 average iterations required by the basic ACO algorithm. Moreover, the IACO curve descends steeply in the first few iterations and then

stabilises, indicating fast exploitation of promising regions. In contrast, the basic ACO curve exhibits visible oscillations and a longer plateau, reflecting its reliance on local heuristics and slow pheromone build-up. These results demonstrate that the global estimation function and adaptive transition rule effectively guide the search, eliminating wasteful exploration and ensuring rapid, reliable convergence.

Regarding convergence speed: the average number of iterations needed by IACO is 12.4, which is only 14.6% of that required by basic ACO (85.2). The best solution appears at iteration 6 for IACO vs 42 for basic ACO. This indicates a faster convergence speed and shorter algorithm execution time. Furthermore, the average convergence result of the improved ACO algorithm is 1203.2, which is superior to the average result of 1256.8 obtained by the basic ACO algorithm, demonstrating the superior search performance of the improved ACO algorithm. In the repeated scheduling process of 20 runs, although the scheduling method using the basic ACO algorithm could also obtain the optimal solution in most cases, when using the improved ACO algorithm, the optimal solution was obtained in all 20 runs, indicating that the algorithm is more stable. The makespan of IACO improvement of 4.27% might seem modest, but in a continuous manufacturing setting with multiple batches, this translates into significant throughput gain. Compared to the assumed sequential processing scenario of the manufacturing system, the scheduling results obtained by the proposed method for this specific batch of parts demonstrate that the makespan can be reduced by approximately 56%.

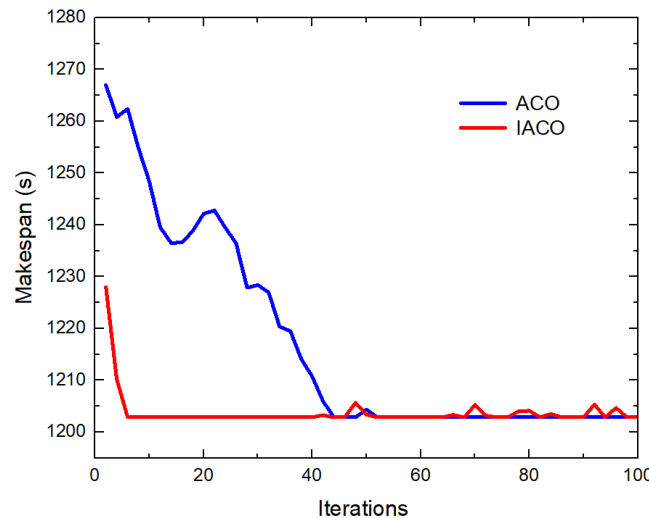


Fig. 5. Convergence curve of the optimal solution over iterations.

Figure 6 presents the Gantt chart of the optimal schedule obtained by the proposed IACO algorithm for the three-job instance. In the IACO schedule, operations are more evenly distributed and machine idle times are reduced. The makespan is shortened to 70

1203 s, compared to 2145 s for the sequential production baseline. This improved load balancing and reduced waiting time directly result from the global estimation function, which guides ants away from decisions that would overload any single resource. Thus, Figure 6 demonstrates that IACO not only yields a shorter makespan but also produces a more balanced and practical schedule.

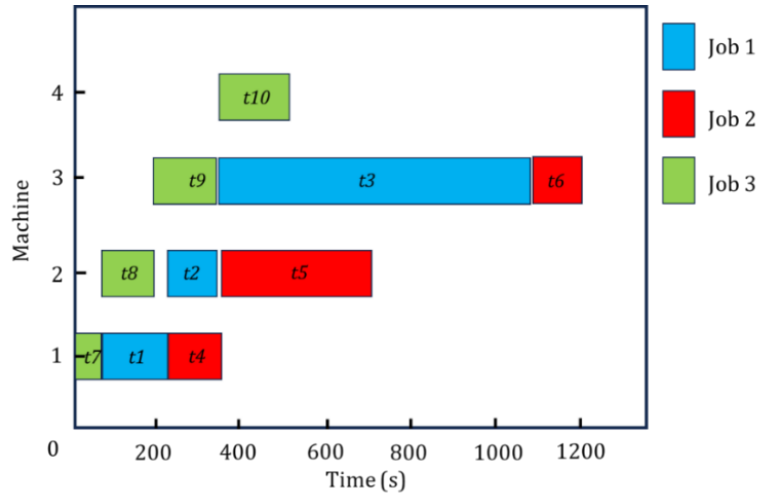


Fig. 6. The optimal scheduling gantt chart.

Tab. 3. Comparison of results between basic ant colony algorithm and improved ant colony algorithm

Parameter	Basic ant colony algorithm	Improved ant colony algorithm
Average number of iterations	85.2	12.4
Optimal number of iterations	42	6
Average completion time	1256.8	1203.2
Optimal completion time	1203	1203
Optimal transition firing sequence	$t_7 \rightarrow t_1 \rightarrow t_8 \rightarrow t_2 \rightarrow t_4 \rightarrow t_9 \rightarrow t_5 \rightarrow t_3 \rightarrow t_{10} \rightarrow t_6$	$t_7 \rightarrow t_1 \rightarrow t_8 \rightarrow t_2 \rightarrow t_4 \rightarrow t_9 \rightarrow t_5 \rightarrow t_3 \rightarrow t_{10} \rightarrow t_6$
Number of times optimal solution was reached in 20 runs	15	20

Furthermore, to evaluate the efficacy of the IACO algorithm in practical FMS environments, this study extends the application of the proposed algorithm to larger production scales, specifically involving 10, 100, and 500 parts. The resulting scheduling outcomes for each scale are presented in Tab. 4. Observations at larger production scales reveal that the IACO algorithm not only excels in convergence rate and computational

efficiency but also delivers superior makespan optimization relative to the conventional ACO. Specifically, for above various scales, the proposed algorithm reduced makespan to 54.82%, 70.86%, and 64.24% of the current factory baseline, respectively. While the marginal gains over the basic ACO may appear incremental, such optimizations are of paramount importance in practical FMS.

Based on the analysis summarized above, the improved ant colony optimization algorithm proposed in this article shows significant improvements in both the quality of the solution and the solution efficiency (or computational efficiency).

Tab. 4. Comparison of results between basic ant colony algorithm and improved ant colony algorithm

Parameter	10 parts		100 parts		500 parts	
	ACO	IACO	ACO	IACO	ACO	IACO
Optimal number of iterations	165	25	195	32	316	52
Optimal completion time (s)	12300	11760	162000	152000	725000	689000
Makespan improvement ratio (%)	57.34	54.82	75.52	70.86	67.6	64.24

7. Conclusion

This study has presented an integrated scheduling optimization framework for CNC-based flexible manufacturing systems by combining Petri net modeling with an improved ant colony optimization algorithm. The main novelty lies in the introduction of a global estimation function, which provides an optimistic lower bound of the remaining makespan and is directly incorporated into the transition probability. Unlike conventional ACO that relies only on local heuristics, this mechanism enables the algorithm to proactively compute global heuristics and balance resource loads. The advantages of the proposed IACO are faster convergence, better solution quality, and higher stability. Experimental results show that IACO reaches the optimal solution after an average of 12.4 iterations compared to 85.2 for basic ACO, reduces the average makespan from 1256.8 to 1203.2 in small-scale and by up to 6.2% in large-scale scenarios, and achieves a 100% success rate over 20 runs while basic ACO reaches only 75%. Compared to the sequential production baseline, the makespan is reduced by approximately 56%. Future research will extend the evaluation to larger scales with more jobs and alternative routes, as well as compare the effectiveness of the Petri net modeling approach combined with

the IACO algorithm against other metaheuristic algorithms, in order to comprehensively assess the scheduling performance of different research methods. The proposed framework therefore provides an effective and practical scheduling optimization strategy for real-world FMS.

References

- [1] S. Lassoued *et al.*, "Flexible Manufacturing Systems intralogistics: Dynamic optimization of AGVs and tool sharing using Colored-Timed Petri Nets and actorcritic RL with actions masking", *Journal of Manufacturing Systems*, Vol. 82, pp. 405–419, 2025. DOI: 10.1016/j.jmsy.2025.06.017
- [2] N. He, M. Sahnoun, D. Zhang, and B. Bettayeb, "A hybrid approach using ant colony optimisation for integrated scheduling of production and transportation tasks within flexible manufacturing systems", *Computers & Operations Research*, Vol. 180, 2025. DOI: 10.1016/j.cor.2025.107059
- [3] A. Momenikorbekandi and T. Kalganova, "Intelligent scheduling methods for optimisation of job shop scheduling problems in the manufacturing sector: A systematic review", *Electronics*, Vol. 14, No. 8, 2025. DOI: 10.3390/electronics14081663
- [4] B. Xin, S. Lu, Q. Wang, and F. Deng, "A review of flexible job shop scheduling problems considering transportation vehicles", *Frontiers of Information Technology and Electronic Engineering*, Vol. 26, No. 3, pp. 332–353, 2025. DOI: 10.1631/FITEE.2400123
- [5] Y. Fu, Y. Hou, Z. Wang *et al.*, "Distributed scheduling problems in intelligent manufacturing systems", *Tsinghua Science and Technology*, Vol. 26, No. 5, pp. 625–645, 2021. DOI: 10.26599/TST.2021.9010021
- [6] J. Lohmer and R. Lasch, "Production planning and scheduling in multi-factory networks: A systematic literature review", *International Journal of Production Research*, Vol. 59, No. 7, pp. 2028–2054, 2021. DOI: 10.1080/00207543.2020.1834162
- [7] D. Marco, M. Birattari, and T. Stutzle, "Ant colony optimization", *IEEE Computational Intelligence Magazine*, Vol. 1, No. 4, pp. 28–39, 2006.
- [8] Z. Zhang and Y. Fu, "A learning-driven multi-objective cooperative artificial bee colony algorithm for distributed flexible job shop scheduling with maintenance and transportation", *Computers & Industrial Engineering*, Vol. 196, 2024. DOI: 10.1016/j.cie.2024.110484
- [9] Z. He *et al.*, "Scheduling of flexible manufacturing systems based on place-timed Petri nets and basis reachability graphs", *IEEE Transactions on Control Systems Technology*, 2025.

- [10] X. Li, L. Li, and Z. He, "Optimal scheduling of flexible manufacturing systems based on timed Petri nets and cost function", *Control Theory & Applications*, Vol. 43, No. 2, pp. 415–424, 2026.
- [11] H. Jin *et al.*, "Bridging speed and optimality in job scheduling: A hybrid ant colony optimization approach for distributed systems", in *Proceedings of 2025 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis (SC Workshops)*, 2025, pp. 2190–2200.
- [12] A. Raza, G. Yuan, C. Wang, X. Liu, and T. Hu, "Enhancing manufacturing efficiency through symmetry-aware adaptive ant colony optimization algorithm for integrated process planning and scheduling", *Symmetry*, Vol. 17, No. 6, 2025. DOI: 10.3390/sym17060824
- [13] C. Shen and Y. L. Chen, "Blocking flow shop scheduling based on hybrid ant colony optimization", *International Journal of Simulation Modelling*, Vol. 19, No. 2, pp. 313–322, 2020. DOI: 10.2507/IJSIMM19-2-CO7
- [14] R. Li *et al.*, "Two-stage knowledge-driven evolutionary algorithm for distributed green flexible job shop scheduling with type-2 fuzzy processing time", *Swarm and Evolutionary Computation*, Vol. 74, 2022. DOI: 10.1016/j.swevo.2022.101139
- [15] M. Gonzalo and J. Pereira, "Multiobjective scheduling algorithm for flexible manufacturing systems with Petri nets", *Journal of Manufacturing Systems*, Vol. 54, pp. 272–284, 2020.
- [16] S. Zhang, T. Hou, and Q. Qu, "An improved mayfly algorithm for distributed flexible job shop scheduling under dual resource constraints", *Sustainability*, Vol. 14, No. 19, 2022. DOI: 10.3390/su141912120
- [17] M. Dorigo, V. Maniezzo, and A. Coloni, "Ant system: Optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, Vol. 26, No. 1, pp. 29–41, 1996.
- [18] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem", *IEEE Transactions on Evolutionary Computation*, Vol. 1, No. 1, pp. 53–66, 1997.
- [19] T. Stützle and H. Hoos, "Improvements on the ant-system: Introducing the MAX–MIN ant system", *Artificial Neural Nets and Genetic Algorithms*, Springer, 1996, pp. 245–249.
- [20] B. Bullnheimer, R. F. Hartl, and C. Strauss, "A new rank based version of the ant system: A computational study", *Central European Journal of Operations Research*, Vol. 7, No. 1, pp. 25–38, 1999.

TỐI ƯU HÓA LẬP LỊCH DỰA TRÊN MẠNG PETRI TRONG HỆ THỐNG SẢN XUẤT LINH HOẠT CNC SỬ DỤNG THUẬT TOÁN ĐÀN KIẾN TĂNG CƯỜNG ƯỚC LƯỢNG TOÀN CỤC

Nguyễn Tuấn Dũng¹, Trần Anh Vàng¹, Nguyễn Quốc Việt¹

¹Khoa Hàng không vũ trụ, Trường Đại học Kỹ thuật Lê Quý Đôn

Tóm tắt: Bài báo nghiên cứu bài toán tối ưu hóa lập lịch trong hệ thống sản xuất linh hoạt (FMS) dựa trên máy CNC. Một mô hình mạng Petri được phát triển để biểu diễn chính xác các hành vi động, đồng thời và chia sẻ tài nguyên của hệ thống sản xuất. Dựa trên đồ thị khả đạt (reachability graph) thu được từ mô hình mạng Petri, bài toán lập lịch được phát biểu thành bài toán tìm đường đi tối ưu trong không gian trạng thái của hệ thống. Để khám phá hiệu quả không gian tìm kiếm này, một thuật toán tối ưu hóa đàn kiến cải tiến (IACO) được đề xuất. Thuật toán tích hợp hàm ước lượng toàn cục, quy tắc chuyển trạng thái thích nghi và cơ chế cập nhật pheromone kép nhằm cải thiện hiệu suất hội tụ và độ ổn định của lời giải. Kết quả thực nghiệm cho thấy thuật toán IACO được đề xuất vượt trội đáng kể so với thuật toán tối ưu hóa đàn kiến (ACO) cổ điển. Cụ thể, IACO giảm số lần lặp trung bình cần thiết để hội tụ từ 85,2 xuống 12,4 và đạt được lời giải tối ưu trong cả 20 lần chạy độc lập, trong khi ACO cổ điển chỉ đạt được 15 lần. Thời gian hoàn thành trung bình được giảm từ 1256,8 xuống 1203,2 đơn vị thời gian. Hơn nữa, trong các kịch bản sản xuất quy mô lớn hơn, IACO đạt được sự cải thiện đáng kể về thời gian hoàn thành so với đường cơ sở thực tế của nhà máy, đồng thời cũng giảm đáng kể thời gian hội tụ để đạt được lịch trình tối ưu so với số lần lặp yêu cầu của ACO cổ điển. Những kết quả này khẳng định rằng khung IACO được đề xuất cung cấp một giải pháp mới và hiệu quả cho lập lịch FMS quy mô lớn bằng cách kết hợp độc đáo phân tích đồ thị khả đạt mạng Petri với tối ưu hóa đàn kiến tăng cường ước lượng toàn cục, từ đó giải quyết những hạn chế của các phương pháp dựa trên heuristic cục bộ.

Từ khóa: Hệ thống sản xuất linh hoạt; lập lịch tối ưu; mạng Petri; thuật toán tối ưu đàn kiến cải tiến.

Received: 10/03/2026; Revised: 19/04/2026; Accepted for publication: 26/05/2026

