

PG-FVEV: Petri-net-constrained genetic scheduling and offline virtual execution validation for a rail-mounted manipulator-CNC flexible manufacturing line

Bowen Zhang

1650116648@qq.com

Shenyang Ligong University <https://orcid.org/0009-0009-9222-4637>

Research Article

Keywords: CNC flexible manufacturing line, Full-process makespan, Genetic algorithm, Petri net, Virtual execution validation

Posted Date: October 16th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11382116/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

PG-FVEV: Petri-net-constrained genetic scheduling and offline virtual execution validation for a rail-mounted manipulator-CNC flexible manufacturing line

Bowen Zhang

School of Mechanical Engineering, Shenyang Ligong University, Shenyang 110159, China

Keywords: CNC flexible manufacturing line; Full-process makespan; Genetic algorithm; Petri net; Virtual execution validation

Correspondence to: Bowen Zhang / 1650116648@qq.com

Abstract

Rail-mounted manipulator and CNC flexible manufacturing lines provide an effective approach to improving machining flexibility and production coordination in small-batch manufacturing. However, conventional numerical scheduling mainly provides dispatching sequences and machine assignments, while resource-state constraints, downstream completion and execution consistency are not fully represented. To address this gap, this paper proposes PG-FVEV, an offline scheduling and virtual execution validation framework. First, genetic scheduling is combined with Petri-net resource constraints and q/g dispatch mapping to convert scheduling decisions into resource-constrained execution events. Second, a new full-process makespan metric is proposed to extend the completion criterion from CNC-stage completion to final rack entry. Finally, a hierarchical validation scheme integrates exhaustive evaluation, MIoT.VC virtual execution checking and a larger offline scheduling case. The results show that PG-FVEV provides a coherent connection between scheduling decisions and production-line execution and supports process-level evaluation of flexible manufacturing-line schedules.

1. Introduction

Rail-mounted manipulator-CNC flexible manufacturing lines support flexible, small-batch automated machining [1]. Their scheduling problem extends beyond CNC assignment because machining and downstream execution must be coordinated across the shared manipulator, multiple CNC resources and subsequent handling stages [2]. A feasible schedule must therefore specify the workpiece dispatching sequence, CNC assignment and shared-resource constraints. In this context, a numerical scheduling result cannot be directly regarded as an executable production-line process. A scheduling result derived from a genetic algorithm (GA), $X=(seq,assign)$, provides the dispatching sequence and CNC assignment, but it still needs resource-state semantics for occupation, release and workstation connection [3]. CNC-stage completion also cannot fully represent whole-line completion, because post-machining handling, inspection, transfer and output operations affect the final completion state [4]. In addition, a finite searchable case is needed to check result validity, whereas a larger problem case is needed to evaluate offline search performance and cross-run variability. GA methods provide a basis for combinatorial search under multiple resource constraints [5]. Petri nets supply the state-transition and mutual-exclusion semantics required at the execution layer. Offline virtual validation can be used to observe workpiece-flow and action-connection consistency [6]. The remaining transformation problem is therefore how a GA-derived schedule can be converted into a resource-state-constrained event sequence that supports coherent offline validation across the virtual production line.

To address this problem, this paper proposes PG-FVEV for a rail-mounted manipulator-CNC flexible manufacturing line. The main contributions of this paper are threefold. First, PG-FVEV develops a schedule-to-execution transformation mechanism that maps the GA-derived scheduling decision $X=(seq,assign)$ into resource-state-constrained execution events through Petri-net resource constraints and q/g dispatch mapping. This mechanism addresses the lack of a clear transformation path between numerical scheduling results and virtual production-line execution logic. Second, this paper proposes the full-process makespan C_{max}^{full} as a new evaluation metric for final schedule evaluation and selection, while retaining C_{max}^{CNC} as a CNC-stage reference. By extending the completion criterion from the CNC stage to final rack entry, C_{max}^{full} reflects the final completion state of the entire production line and provides a more appropriate basis for final schedule selection. Third, to improve experimental scalability and result verifiability, this paper adopts a hierarchical experimental scheme. Case 1 provides an exhaustively enumerable reference and supports MIoT.VC offline virtual execution validation, whereas Case 2 evaluates offline search performance and variability at an expanded problem scale. The purpose of this design is to distinguish finite-case result verification from expanded-scale search evaluation.

2. Related work

2.1 Multi-resource manufacturing scheduling

Scheduling in a rail-mounted manipulator-CNC flexible manufacturing line involves job dispatching, CNC

assignment and shared downstream resources [7]. Its feasibility depends on coordinated machining and downstream execution rather than pure machine sequencing [8]. Flexible job-shop scheduling provides the basis for joint job sequencing and machine selection [9,10].

Recent studies extend production scheduling from machining to logistics and auxiliary-resource coordination [11]. Liang et al. [1] integrated shared AGVs with production- and storage-capacity constraints, demonstrating cross-resource coordination. Han et al. [12] coupled machining and transport through machine selection, operation sequencing and vehicle assignment. Liu et al. [3] incorporated machines, fixtures and pallets into flexible job-shop scheduling, showing that engineering schedules involve multiple discrete resource types. Loading and unloading have likewise been modelled as dual-resource constraints [13]. Yang et al. [14] combined machining, AGV transport and assembly in a three-stage flexible job shop, showing that downstream operations affect overall completion time.

Together, these studies show that multi-resource scheduling increasingly coordinates machining with downstream operations. However, models centred on transport vehicles or auxiliary resources do not fully represent exclusive use of a shared rail-mounted manipulator during machining and transfer. This paper therefore models workpiece dispatching and CNC assignment, then checks resource mutual exclusion and workstation connections at the execution level.

2.2 Genetic-algorithm scheduling

Genetic algorithms are commonly used in production scheduling to search combinatorial spaces involving job sequences and resource assignments [15]. In the PG-FVEV framework, the genetic algorithm undertakes the scheduling-search function, and its output is the scheduling decision structure $X=(seq,assign)$. The two components carry the dispatch-order and CNC-assignment information required by later resource mapping. This decision provides the scheduling input for later constraint mapping and validation. Dual-resource batch-scheduling studies have also used double coding with handling-resource decoding [16].

GA scheduling studies increasingly include multi-objective evaluation, hybrid search and resource-constrained encoding [17]. Burmeister et al. [18] combined memetic NSGA-II and local search for makespan and energy-cost objectives. Meng et al. [5] used an improved GA with population-diversity checking for a flexible job shop with finite AGVs. Fan et al. [19] adjusted chromosome and neighbourhood-search designs for reconfigurable machine tools with limited auxiliary modules. Park et al. [20] used feasibility-independent encoding and three-phase decoding to maintain feasibility

under complex constraints.

This line of work indicates that GA-based scheduling is suitable for search spaces governed by job sequences, resource assignments and engineering constraints [21]. However, existing GA scheduling studies often emphasize machining-stage metrics, energy consumption or resource utilisation, while downstream execution factors remain insufficiently considered. The GA-derived scheduling decision therefore requires execution-layer resource mapping so that shared-resource conditions can be considered before virtual validation. The decision $X=(seq,assign)$ is used here as the input to that mapping, which links the numerical schedule to its event-level execution representation.

2.3 Petri-net resource modelling

A rail-mounted manipulator-CNC flexible manufacturing line has discrete-event characteristics: job, manipulator, CNC and inspection-station states change as events occur. Petri nets describe such systems through places, transitions and token flows. Murata [22] summarised core Petri-net properties and their use in reachability and system-behaviour analysis. In flexible manufacturing-system applications, Petri nets also connect process modelling and scheduling [23]. Xu et al. [24] used P-timed Petri nets for temporal constraints and scheduling search. Yamalidou et al. [25] constructed place-invariant-based Petri-net feedback control, supporting token representation under finite-resource constraints. For resource conflicts and deadlock problems in flexible manufacturing systems, Petri nets have also been widely used for liveness analysis and deadlock prevention [26]. Ezpeleta et al. proposed a Petri-net-based deadlock-prevention strategy for flexible manufacturing systems and used control places to restrict entry into states that may lead to deadlock [27]. Gu et al. [28] used a basis reachability graph to verify and enforce the liveness and deadlock-freeness of bounded Petri nets, showing that Petri nets can provide structured reachability-analysis tools for resource-constrained systems. Lu et al. [29] developed an iterative deadlock-prevention method for flexible manufacturing systems based on resource-requirement graphs, showing that resource-competition relationships can be supervised through formal structures. Beyond resource-conflict and deadlock control, timed coloured Petri-net simulation has also been used to represent production-logistics coordination [30].

These studies show that Petri nets provide formal tools for resource-state representation, reachability analysis and deadlock control in flexible manufacturing systems. However, they mainly address model analysis or supervisory control and do not clarify how a GA-derived scheduling decision can be converted into resource-constrained execution events. To address this

gap, PG-FVEV uses the Petri net only as a resource-state and event-enabling constraint layer. The GA remains responsible for schedule search, while q/g dispatch mapping converts $X=(seq,assign)$ into an ordered execution sequence for subsequent MIoT.VC offline virtual execution validation.

2.4 Simulation and virtual validation

Virtual environments have been used to examine manufacturing processes beyond numerical evaluation alone. Barbieri et al. [31] developed a virtual commissioning methodology that integrates digital representations into manufacturing systems. For production scheduling, Ghasemi et al. [32] reviewed simulation optimisation studies and showed that process models can provide a more complete environment for evaluating candidate schedules. These studies indicate that a scheduling result should be examined not only through its objective value but also through the production process represented in the virtual model.

Virtual commissioning emphasizes equipment behaviour and control consistency. Striffler and Voigt [6] showed that it can reveal control and coordination errors before implementation, while Raza et al. [33] extended the concept from machines to production-line and factory levels. Wooley et al. [34] distinguished discrete-event simulation from digital twins by their synchronization and updating mechanisms. Negri et al. [35] examined field-synchronized digital-twin scheduling based on continuous physical-virtual interaction, unlike the deterministic offline validation considered here.

Simulator-GA systems have evaluated flexible routing in FMSs [36], while simulation optimisation evaluates candidate schedules in process models [37]. These approaches support schedule evaluation and equipment-behaviour checking, but often address predefined control logic or individual stages rather than consistency across the complete production flow. Here, MIoT.VC reproduces workpiece flow from dispatching to final rack entry in an offline virtual line. It then checks the recorded execution against the selected dispatching order, CNC routing and workstation transitions under deterministic conditions, providing full-process virtual execution evidence beyond numerical scheduling results.

3. Methodology

Section 3 presents the PG-FVEV methodology, which integrates offline GA scheduling, Petri net resource constraints, q/g dispatch mapping and MIoT.VC virtual execution validation. Sections 3.1-3.2 define the framework and scheduling object. Sections 3.3-3.5 then establish the resource constraints, GA search and q/g dispatch mapping. Section 3.6 completes the methodological chain through MIoT.VC offline virtual execution validation.

3.1 Overall PG-FVEV framework

To address the problem that GA-derived scheduling decisions cannot directly specify resource-constrained execution events, this paper proposes the PG-FVEV framework, as shown in Fig. 1.

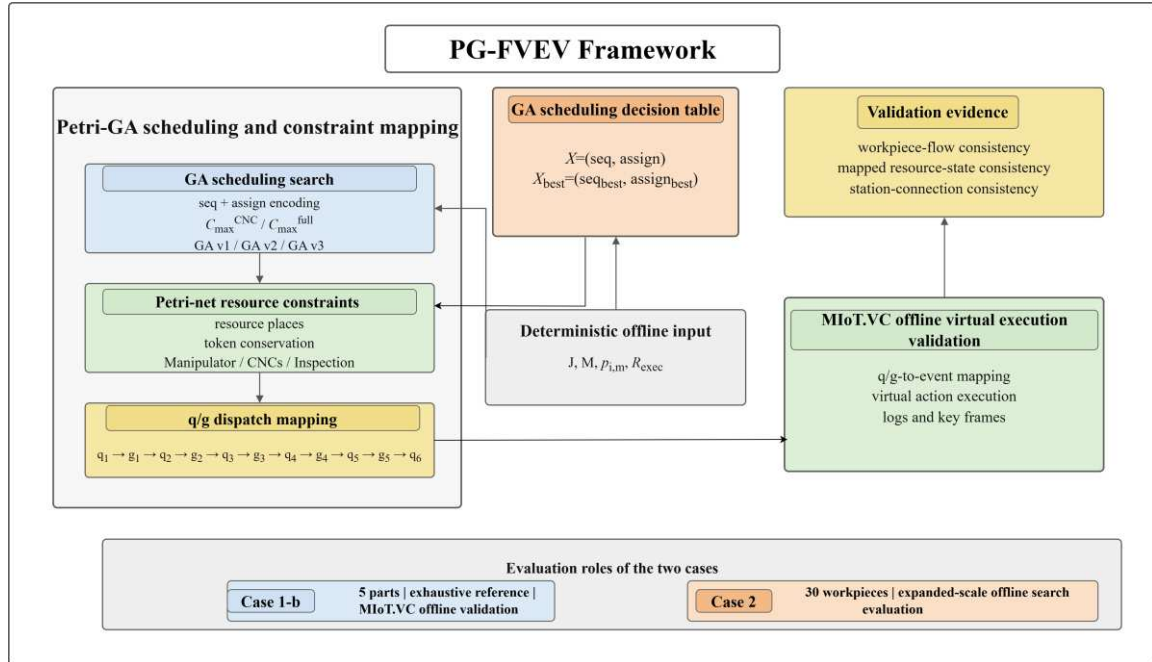


Fig. 1. Overall structure of the PG-FVEV framework.

The PG-FVEV procedure begins with deterministic

offline inputs that define the scheduling objects, processing times and execution-resource scope of the studied line. Based on these inputs, the GA searches the scheduling decision $X=(seq,assign)$ and selects $X_{best}=(seq_{best},assign_{best})$, which specifies the workpiece dispatching sequence and CNC assignments but does not directly describe equipment actions. The Petri-net layer therefore supplements the selected schedule with the resource occupation, release, mutual-exclusion and event-enabling conditions required during execution. Under these conditions, q/g dispatch mapping associates the workpiece at each dispatch position with its assigned CNC machine and forms an ordered resource-constrained dispatch sequence. MIoT.VC then examines the mapped sequence, rather than the raw GA output, to determine whether coherent workpiece flow and workstation connections can be maintained through final rack entry in the deterministic offline virtual environment. Within this framework, the selected Case 1-b schedule is used for offline virtual execution validation, whereas Case 2 is used only to evaluate offline search performance and cross-run variability at an expanded problem scale.

3.2 Scheduling problem overview

The scheduling problem is defined using two cases with different evaluation roles. Case 1 contains five jobs and provides an exhaustively enumerable reference, while the selected Case 1-b schedule is further used for MIoT.VC offline virtual execution validation. Case 2 contains 30 jobs and is used only to examine offline GA search performance and variation across independent runs. The job set is denoted as $J=\{J_1, J_2, \dots, J_n\}$, where $n=5$ corresponds to the five-workpiece exhaustive reference case and $n=30$ to the 30-workpiece offline search case. The CNC machining-resource set is $M=\{M_1, M_2, M_3\}$. The processing time of job J_i on CNC machine M_m is denoted as $p_{i,m}$, and all $p_{i,m}$ values form the deterministic processing-time matrix used for schedule evaluation. On this basis, the execution-resource set is $R_{exec}=\{\text{Robot}, \text{Inspection}, \text{DropBuffer}, \text{Lift}, \text{Rack}\}$, corresponding to the rail-mounted manipulator, inspection station, drop-buffer conveyor, lift platform and multi-layer storage rack. Each job follows the fixed process route defined by the virtual layout used for MIoT.VC validation. This route is constrained by process sequence, single-resource manipulator occupation, mutually exclusive CNC capacity and exclusive inspection-station use. These constraints enforce one-way workpiece flow and exclusive use of constrained resources. With the scheduling object and resource scope defined, the scheduling decision is defined as $X=(seq,assign)$, where $seq=(seq_1, seq_2, \dots, seq_n)$ denotes the job dispatching sequence and seq_k denotes the job entering the k th dispatching position. The mapping $assign:J \rightarrow M$ denotes

the assignment relation from jobs to CNC machines, and $assign(J_i)=M_m$ means that job J_i is assigned to CNC machine M_m for machining. The seq layer determines dispatching order, and the $assign$ layer determines the machining resource selected for each job. These two layers form the GA search object and provide the scheduling and resource basis for subsequent Petri-net q/g dispatch mapping.

For the selected scheduling solution, the relationship between the scheduling decision and the k th dispatching position is defined as follows.

$$X_{best}=(seq_{best},assign_{best}) \quad (1)$$

$$j_k=(seq_{best})_k \quad (2)$$

$$m_k=assign_{best}(j_k) \quad (3)$$

In Eqs. (1)-(3), X_{best} denotes the selected scheduling decision, j_k is the job at the k th dispatching position, and m_k is the CNC machine assigned to j_k . These definitions provide the scheduling input for the q/g dispatch mapping in Section 3.5.

3.3 Petri-net constraint model

Based on the production-line objects, execution resources and process route defined in Section 3.2, this study establishes a Petri-net constraint model to describe process and resource-state constraints. The system model is denoted as $PN=(P,T,F,W,M_0)$ [22], where P is the set of places, T is the set of transitions, F is the set of directed arcs, W is the arc-weight function and M_0 is the initial marking.

To distinguish modelling functions, P is divided into P_{job} , P_{res} and P_{ctrl} . P_{job} describes the predefined workpiece-state evolution, P_{res} represents idle and occupied states of finite resources, and P_{ctrl} receives the GA-derived dispatching order and CNC assignment relation for subsequent q/g dispatch mapping. The Petri net does not re-search the schedule. Instead, it constrains the selected scheduling decision into an event sequence satisfying process-state and resource-state conditions.

The transition set T represents the principal handling, machining, inspection and output events. A transition can fire only when the required workpiece-state and resource-state input places contain tokens. Firing transfers the corresponding tokens to the output places and updates the workpiece and resource states. For each shared finite resource, the idle and busy places form a token-conservation relation. Specifically, $Mark(Robot_{idle})+Mark(Robot_{busy})=1$, $Mark(CNC_{m,idle})+Mark(CNC_{m,busy})=1$ for each CNC machine m , and $Mark(Insp_{idle})+Mark(Insp_{busy})=1$. Because each resource-state subnet is initialized with one token and the associated transitions only transfer this token between the idle and busy places, these relations are preserved during firing. The corresponding resource-state subnets are therefore 1-

bounded, which prevents simultaneous multiple occupation of the same manipulator, CNC machine or inspection station. This structural property characterizes

the bounded mutual-exclusion behavior of the finite-resource subnets used in the scheduling model.

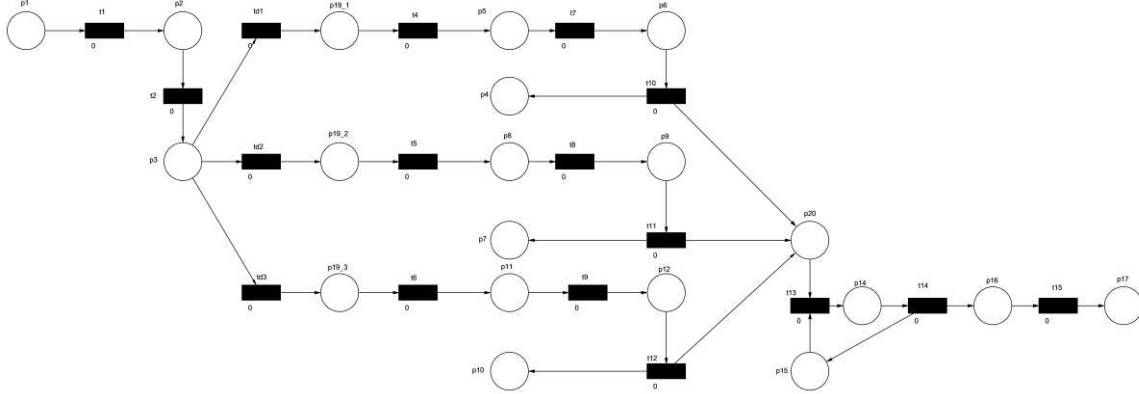


Fig. 2(a). Core process Petri net of the rail-mounted manipulator-CNC flexible manufacturing line.

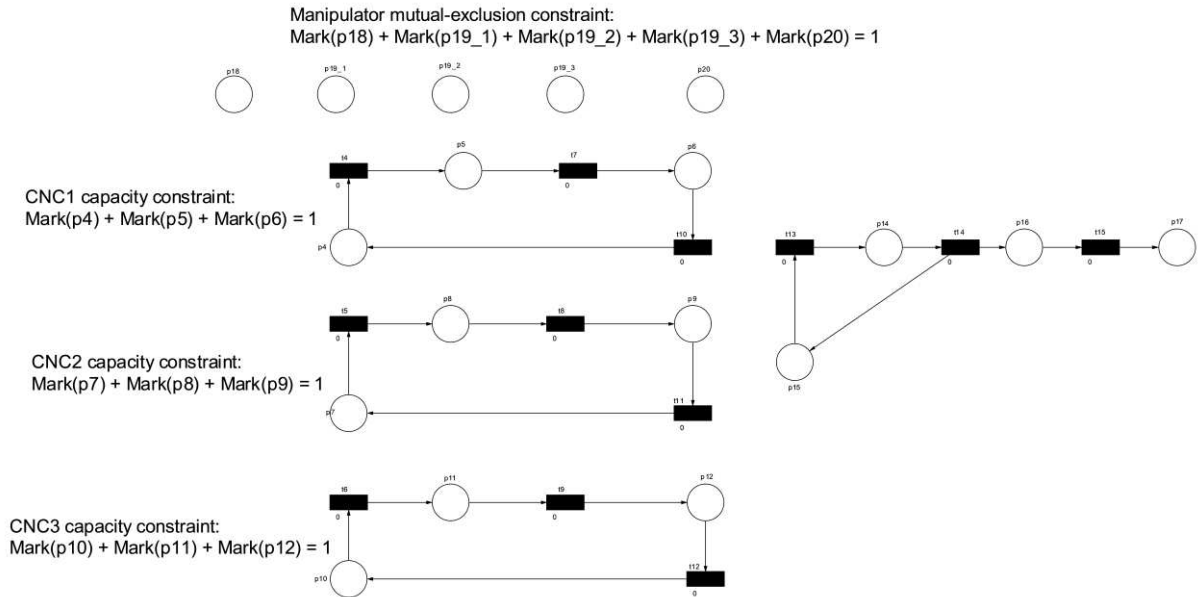


Fig. 2(b). Resource mutual-exclusion subnet for the manipulator, CNC machines and inspection station.

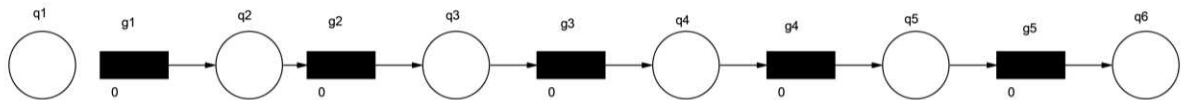


Fig. 2(c). q/g dispatch-control layer instantiated by the genetic-algorithm scheduling result.

The Petri-net model in Fig. 2 integrates the coordination required for the selected schedule. The process-flow subnet defines the prescribed evolution of each workpiece through machining and downstream operations. On this basis, the resource subnet embeds the idle and occupied states of the manipulator, CNC machines and inspection station into the relevant transitions, preventing an operation from proceeding when its required resource is unavailable. The q/g dispatch-con-

trol layer then connects the selected dispatching sequence and CNC assignments to these enabling conditions without performing an additional scheduling search. Together, the three submodels constrain the selected GA decision into an ordered dispatch process that satisfies both workpiece-state and resource-state requirements. The simplified structure is presented in Fig. 2(a)-(c), while the complete engineering model is provided in Supplementary Fig. S1. The Petri-net layer

is therefore used here for resource-state constraint representation and event enabling rather than as a formal deadlock-prevention or liveness-verification mechanism.

3.4 Genetic algorithm scheduling design

After the scheduling object and resource rules have

been defined, the GA searches for the scheduling decision $X=(seq,assign)$, where seq gives the workpiece dispatching order and $assign(J_i) \in M$ gives the CNC machine assigned to workpiece J_i . Algorithm 1 presents the population initialization, evaluation, genetic operations and best-solution update used in the offline search.

Algorithm 1. Offline GA search for the scheduling decision $X=(seq,assign)$

1: Set the population size N_p , generation limit G , crossover probability p_c , mutation probability p_m , elite size N_e and tournament size N_t

2: if $v=GA$ v1 then

3: Set the objective to minimize C_{max}^{CNC}

4: Fix seq as the prescribed sequence seq_0

5: Initialize N_p assignment components and combine each assignment with seq_0

6: else

7: if $v=GA$ v2 then

8: Set the objective to minimize C_{max}^{CNC}

9: else

10: Set the objective to minimize C_{max}^{full}

11: end if

12: Initialize N_p individuals with random workpiece permutations and random CNC assignments

13: end if

14: Evaluate the initial population using the selected objective

15: Set X_{best} as the individual with the lowest objective value

16: for generation $g=1$ to G do

17: Rank the current population according to the objective values

18: Copy the best N_e individuals to $NewPop$

19: while the size of $NewPop$ is smaller than N_p do

20: Select two parents by tournament selection with size N_t

21: if $v=GA$ v1 then

22: Apply one-point crossover to the assignment components with probability p_c

23: Mutate the CNC assignments with probability p_m

24: Keep the sequence component unchanged

25: else

26: Apply order crossover to the sequence components with probability p_c

27: Apply uniform crossover to the assignment components with probability p_c

28: Apply swap mutation to the sequence components with probability p_m

29: Apply random CNC reassignment to the assignment components with probability p_m

30: end if

31: Add the offspring to $NewPop$

32: end while

33: Evaluate $NewPop$ using the selected objective

34: Update X_{best} if $NewPop$ contains an individual with a lower objective value

35: Replace the current population with $NewPop$

36: end for

37: return X_{best}

GA v1, GA v2 and GA v3 use the same population

search structure but differ in their decision variables

and evaluation objectives. GA v1 keeps seq fixed and changes only $assign$. GA v2 searches both seq and $assign$ using C_{max}^{CNC} , whereas GA v3 uses the same two-part decision structure with C_{max}^{full} . The resulting X_{best} remains a scheduling decision rather than an equipment action program. It is subsequently converted into the q/g dispatch sequence described in Section 3.5.

3.5 q/g dispatch mapping of GA decisions

To support subsequent offline virtual execution validation, the selected scheduling decision must first be

represented at the event level. Although X_{best} specifies the workpiece dispatching order and CNC assignments, it does not directly define the control sequence required for execution. The q/g mapping therefore converts X_{best} into an ordered dispatch-control chain under the resource-state conditions defined above, as shown in Fig. 3.

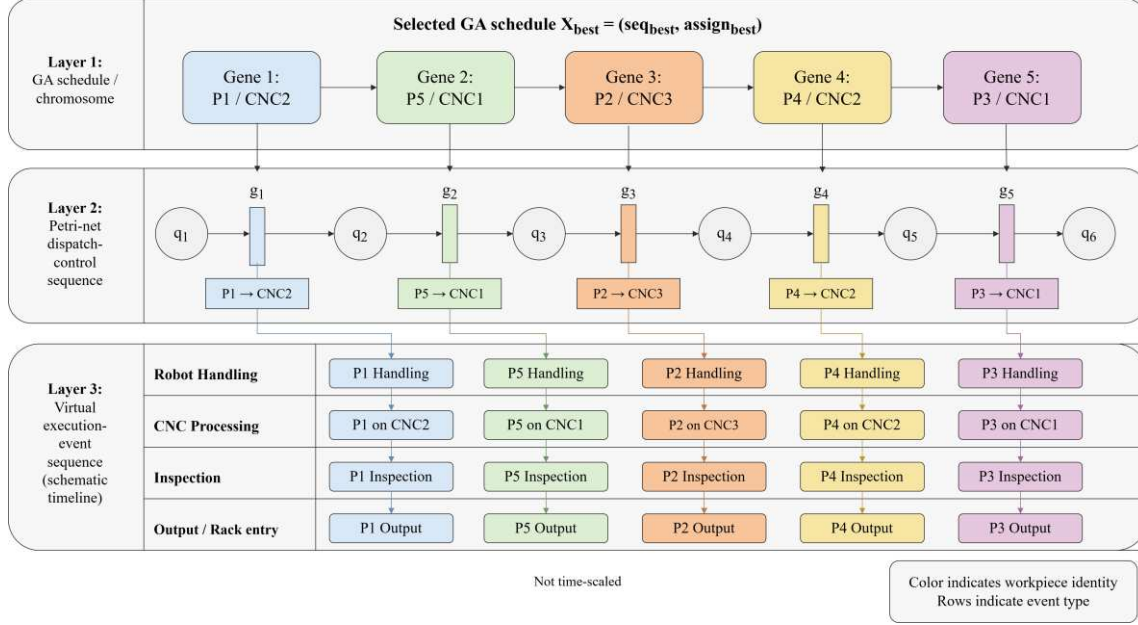


Fig. 3. Mapping from the GA scheduling result to q/g dispatch events.

As represented in Fig. 3, the q/g dispatch-control layer provides the event-level form of the selected scheduling decision $X_{best} = (seq_{best}, assign_{best})$. The sequence seq_{best} determines the order of dispatch positions, whereas $assign_{best}$ identifies the CNC machine assigned to each workpiece. For position k , Eqs. (2) and (3) give the corresponding workpiece j_k and machine m_k . The control place q_k records the current position in the dispatch sequence, and transition g_k represents the dispatch event that advances the control token to the next position. The resulting q/g chain therefore expresses the scheduling decision as an ordered event structure rather than as a list of numerical variables.

However, the event order does not by itself guarantee that a dispatch can occur. Transition g_k is enabled only when the control token is located in q_k , workpiece j_k is ready, the rail-mounted manipulator is idle, and CNC machine m_k is available. Once these conditions are satisfied, firing g_k transfers the token to q_{k+1} and updates the dispatch and resource-occupation states. In this way, the q/g chain retains the order selected by the GA, while the Petri-net conditions determine whether

each event can be executed without violating the represented resource constraints.

Case 1-b illustrates how this event structure is formed for a specific scheduling result. The selected sequence is $seq_{best} = [1, 5, 2, 4, 3]$, and the corresponding CNC assignments are Part1-CNC2, Part5-CNC1, Part2-CNC3, Part4-CNC2 and Part3-CNC1. These relations produce the control chain $q_1 \rightarrow g_1 \rightarrow q_2 \rightarrow g_2 \rightarrow q_3 \rightarrow g_3 \rightarrow q_4 \rightarrow g_4 \rightarrow q_5 \rightarrow g_5 \rightarrow q_6$. Because each transition is checked against the same enabling logic, arrival of the token at q_6 indicates that the five dispatch events have been completed in the specified order under the represented resource conditions. This resource-constrained event sequence, rather than the original GA decision alone, is then used as the input to the MIoT.VC offline virtual execution validation described in Section 3.6.

3.6 MIoT.VC-based virtual execution validation

To complete the methodological link between scheduling optimization and virtual execution validation, the mapped schedule must be examined within a unified

production-line environment. The objective value and *qlg* event sequence describe the intended scheduling result and its resource-constrained dispatch logic, but they do not directly show whether the corresponding workpiece flow, CNC routing, resource occupation and station handovers can be reproduced coherently throughout the virtual process. This additional check is

important because it provides process-level evidence that complements the numerical scheduling result and the event-level mapping. Accordingly, the mapped Case 1-b event sequence is introduced into the MIoT.VC environment for offline execution-consistency validation, as shown in Fig. 4.

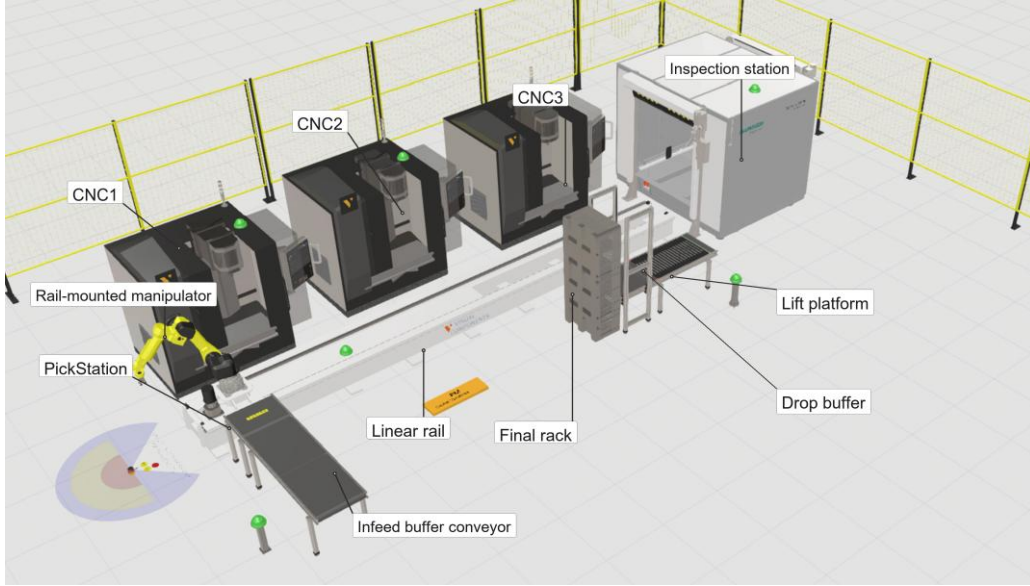


Fig. 4. Virtual layout of the rail-mounted manipulator-CNC flexible manufacturing line.

Fig. 4 provides a unified virtual production-line environment in which the mapped Case 1-b event sequence can be compared with the corresponding execution process. The loading stage, rail-mounted manipulator, CNC machines, inspection station, drop buffer, lift and final rack are represented as virtual objects corresponding to the scheduling entities and Petri-net resource states defined above. On this basis, the validation follows each workpiece from its release and CNC assignment through downstream transfer to final rack entry. It examines whether the dispatching order and CNC routing specified by X_{best} are preserved throughout the process and whether the observed state transitions remain consistent with the represented resource conditions. Key frames capture the visible changes in workpiece position and workstation state, while execution logs provide ordered records of routing and completion. This combination of visual and recorded evidence connects the numerical scheduling result with its virtual execution process and therefore adds process-level support that cannot be obtained from the objective value alone. The conclusion is limited to execution consistency in the deterministic Case 1-b virtual environment and does not constitute physical-equipment verification or online control validation.

4. Experiments and results

4.1 Experimental setup and evaluation metrics

To further evaluate the PG-FVEV framework, a hierarchical experimental design is adopted. The experimental cases and evaluation metrics are shown in Table 1.

Table 1. Experimental cases and evaluation metrics

Experimental case	Job scale	Metric
Case 1-a	5	C_{max}^{CNC}
Case 1-b	5	C_{max}^{full}
Case 2	30	$C_{max}^{CNC}/C_{max}^{full}$

Case 1 contains five workpieces and three CNC machines and provides an exhaustively enumerable reference. Case 1-a uses C_{max}^{CNC} to establish the minimum CNC-stage completion time, while Case 1-b uses C_{max}^{full} to establish the minimum full-process completion time. The selected Case 1-b scheduling result is subsequently used for MIoT.VC virtual production-line execution validation.

Case 2 increases the number of workpieces to 30

while retaining three CNC machines. The corresponding joint search space is approximately $30! \times 3^{30} \approx 5.46 \times 10^{46}$, which substantially increases the scheduling-search difficulty. Case 2 is used to examine

offline search performance and variation across independent runs in the 30-workpiece problem. In addition to the two GA objective settings, a deterministic rule-based scheduler is included as an independent scheduling reference.

Table 2. GA parameter settings

Case/ GA	Variable	Metric	Population size	Generations	Crossover probability	Mutation probability	Random seed
C1-GA v1	<i>assign</i>	C_{max}^{CNC}	60	80	0.80	0.15	fixed seed 42
C1-GA v2	<i>seq+assign</i>	C_{max}^{CNC}	80	120	0.85	0.20	0-29
C1-GA v3	<i>seq+assign</i>	C_{max}^{full}	80	120	0.85	0.20	0-29
C2-GA v2	<i>seq+assign</i>	C_{max}^{CNC}	160	3000	0.85	0.20	0-29
C2-GA v3	<i>seq+assign</i>	C_{max}^{full}	160	3000	0.85	0.20	0-29

As shown in Table 2, the GA parameters are configured according to the different roles of Case 1 and Case 2. In Case 1, GA v2 and GA v3 use the same population size of 80, 120 generations, a crossover probability of 0.85 and a mutation probability of 0.20. This common setting supports the comparison between the CNC-stage objective and the full-process objective in the five-workpiece case. GA v1 uses a smaller population and generation setting because its dispatching sequence is fixed and only the CNC assignment is searched. A fixed seed of 42 is used for GA v1, while seeds 0 to 29 are used for the 30 independent runs of GA v2 and GA v3.

When the number of workpieces increases to 30 in Case 2, the population size is increased to 160 and the search length is extended to 3000 generations. GA v2 and GA v3 use the same crossover probability of 0.85, mutation probability of 0.20, tournament selection with a tournament size of 3, and two elite individuals. The two searches also use the same seeds from 0 to 29, chromosome representation, genetic operators, processing-time matrix and deterministic execution-time parameters. Under this common configuration, GA v2 evaluates candidate schedules using C_{max}^{CNC} , whereas GA v3 uses C_{max}^{full} . The resulting comparison therefore provides a direct assessment of how the two completion criteria influence the scheduling results under the same GA search conditions.

4.2 Scheduling strategies

The scheduling strategies use the decision variables defined in Section 3.2. GA v1 retains the fixed sequence [1,2,3,4,5] and searches only the CNC assignment under C_{max}^{CNC} . GA v2 and GA v3 both search the dispatching sequence and CNC assignment, but use

different primary evaluation criteria. GA v2 minimises C_{max}^{CNC} , while GA v3 minimises C_{max}^{full} . The corresponding strategy settings are summarised in Table 3.

Table 3. Scheduling strategy settings

Strategy	<i>seq</i>	<i>assign</i>	C_{max}^{CNC}	C_{max}^{full}
GA v1		✓	✓	
GA v2	✓	✓	✓	
GA v3	✓	✓		✓
Rule v2		✓	✓	

For Case 2, Rule v2 is additionally introduced as a deterministic scheduling reference. It processes workpieces in the prescribed input order and selects the available CNC machine with the minimum predicted completion time according to the existing resource conditions. If all candidate CNC machines are occupied, the next decision proceeds after the earliest resource release. These strategies provide references for examining how the scheduling decision structure and completion criterion affect the resulting schedules.

4.3 Case 1 results

4.3.1 Scheduling-result analysis

The deterministic result of GA v1 and the 30-run statistical results of GA v2 and GA v3 under the CNC-stage and full-process metrics are shown in Table 4.

Table 4. Statistical scheduling results for Case 1

Strategy	Result type	C_{max}^{CNC}/s	C_{max}^{full}/s
GA v1	Deterministic	171.70	224.90
GA v2	Best	162.20	199.60
GA v2	Mean	162.43	210.66
GA v2	Worst	163.50	222.00
GA v2	Std	0.47	10.03
GA v3	Best	163.50	199.30
GA v3	Mean	163.18	199.59
GA v3	Worst	164.20	201.10
GA v3	Std	0.81	0.39

Among these results, GA v1 establishes the fixed-sequence reference, with $C_{max}^{CNC}=171.70$ s and

$C_{max}^{full}=224.90$ s. By jointly optimising the dispatching sequence and CNC assignment, GA v2 reduces C_{max}^{CNC} to 162.20 s and reaches the exhaustive CNC-stage minimum, confirming its effectiveness for machining-stage optimisation. Under the full-process objective, GA v3 reaches $C_{max}^{full}=199.30$ s, which is the exhaustive minimum verified in Fig. 5(b). Its mean and worst values are 199.59 s and 201.10 s, respectively, so the worst result remains only 1.80 s above the optimum. The corresponding standard deviation of 0.39 s is also substantially lower than the 10.03 s obtained by GA v2 under the same full-process metric. These results indicate that GA v3 remains close to the full-process optimum even in its least favourable run and exhibits substantially lower run-to-run variability and more stable repeated-run performance. C_{max}^{CNC} identifies the machining-stage optimum, whereas C_{max}^{full} provides the whole-line criterion for final schedule selection.

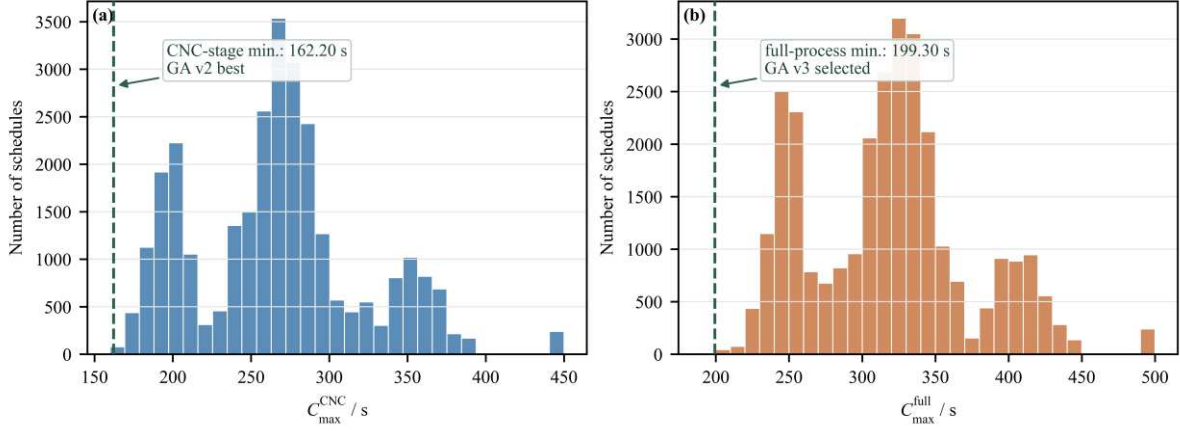


Fig. 5. Exhaustive distributions of Case 1 under the CNC-stage and full-process makespan metrics, with the corresponding minimum values highlighted.

The exhaustive distributions and best scheduling results under the CNC-stage and full-process objectives for Case 1-a and Case 1-b, respectively, are shown in Fig. 5. In Case 1-a, exhaustive enumeration of all 29,160 feasible schedules gives a minimum C_{max}^{CNC} of 162.20 s. GA v2 reaches the same value, whereas the fixed-sequence GA v1 baseline gives 171.70 s. This agreement with the exhaustive minimum confirms that joint optimisation of the dispatching sequence and CNC assignment recovers the CNC-stage optimum in the finite five-workpiece space.

In Case 1-b, the evaluation scope is extended to post-machining handling, inspection and final rack entry. The exhaustive minimum of C_{max}^{full} is 199.30 s. GA v3 reaches this value, while GA v1 gives 224.90 s. The comparison with GA v1 demonstrates the improvement over the fixed-sequence baseline, whereas the agreement with the exhaustive minimum confirms that the

GA v3 result reaches the full-process optimum. The two subfigures therefore establish clear objective-specific references: GA v2 achieves the CNC-stage optimum, while GA v3 achieves the full-process optimum used for final schedule selection.

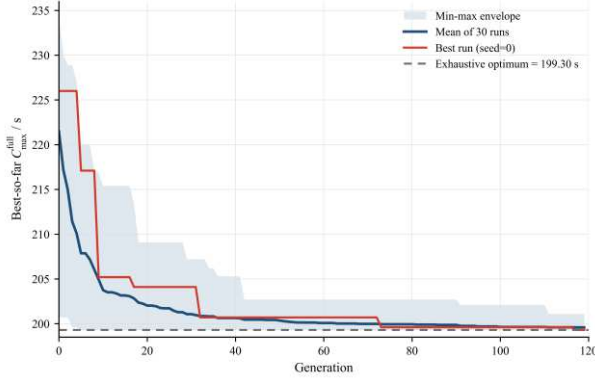


Fig. 6. Convergence process of GA v3 under the full-process objective in Case 1-b.

The convergence process of GA v3 under C_{max}^{full} is

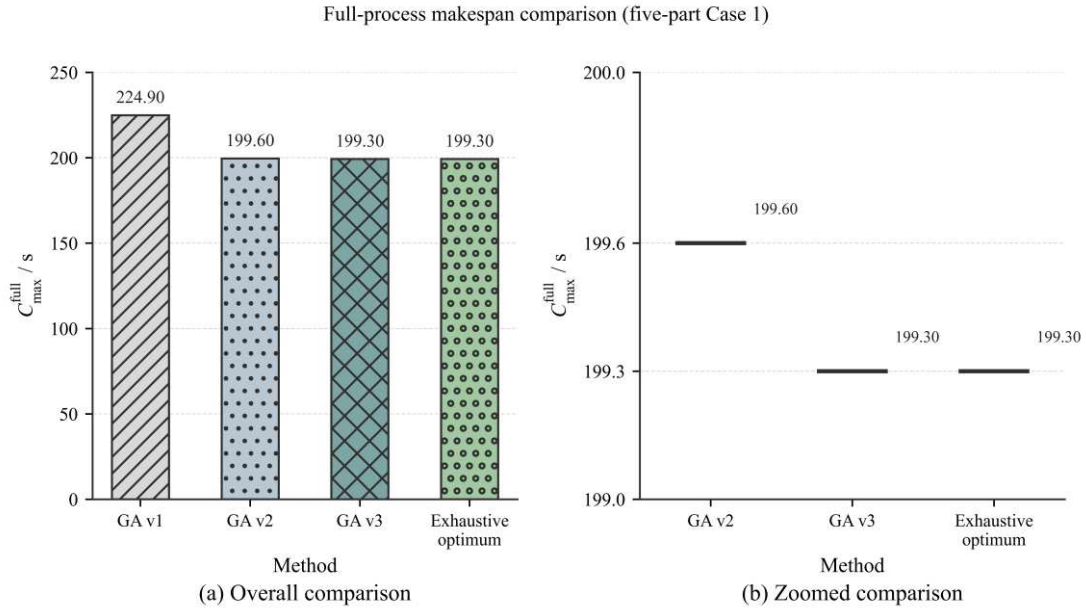


Fig. 7. Best-run comparison among GA v1, GA v2 and GA v3 under the full-process metric in Case 1.

The full-process makespan comparison is presented in Fig. 7. The main view compares the best results obtained by GA v1, GA v2 and GA v3, while the enlarged region provides a clearer view of the two closely spaced joint-search results. Under C_{max}^{full} , the fixed-sequence GA v1 baseline gives 224.90 s. When both the dispatching sequence and CNC assignment are included in the search, the best full-process makespan is reduced to 199.60 s by GA v2 and 199.30 s by GA v3. The enlarged view highlights the 0.30 s difference between these two results and confirms that GA v3 reaches the exhaustive minimum of 199.30 s established for Case 1-b. This best-run comparison should be interpreted together with the 30-run statistics in Table 4.

presented in Fig. 6. The curve decreases rapidly in the early generations and then gradually approaches the minimum full-process value of 199.30 s in the five-work-piece exhaustive distribution, indicating that the full-process objective can guide the search toward the low-value region of C_{max}^{full} within a limited number of generations. The mean curve and the range between the minimum and maximum values show the variation among the reported 30 independent runs. The best GA v3 trajectory fully converged to the exhaustive full-process minimum of 199.30 s, while the small dispersion across 30 independent runs indicates stable and concentrated repeated-run performance.

Across the 30 independent runs, GA v3 gives a mean value of 199.59 s, a worst-run value of 201.10 s and a standard deviation of 0.39 s, compared with 210.66 s, 222.00 s and 10.03 s for GA v2. The lower mean indicates better average full-process performance, while the lower worst-run value shows that the GA v3 results remain close to the optimum even in the least favourable reported run. Its smaller standard deviation further demonstrates that the obtained results are tightly concentrated across independent runs. Combined with the exhaustive optimum of 199.30 s, these results confirm that GA v3 achieves both optimal full-process completion and reliable repeated-run performance, supporting its use to generate the final

schedule for MIoT.VC offline virtual execution validation.

4.3.2 MIoT.VC virtual production-line execution validation

Before the selected scheduling result is used for virtual execution checking, it is necessary to determine whether its mapped dispatching order and CNC assignments are reproduced consistently throughout the virtual production-line process. The selected Case 1-b GA v3 schedule was therefore examined in MIoT.VC under

the deterministic production-line configuration. Three consistency criteria were used: preservation of the prescribed dispatching order, agreement between the scheduled and recorded CNC assignments, and completion of all five workpieces through final rack entry. The prescribed sequence was $seq_{best}=[1,5,2,4,3]$, with Part1, Part5, Part2, Part4 and Part3 assigned to CNC2, CNC1, CNC3, CNC2 and CNC1, respectively. The recorded execution was compared with these mapped relations using the MIoT.VC logs and the key frames shown in Fig. 8.



Fig. 8. Key frames of MIoT.VC virtual production-line execution validation.

The recorded virtual progression covers pickup, CNC loading and unloading, inspection transfer, lift-conveyor receiving and final rack storage (Fig. 8(a)-(f)). It preserves the dispatch order Part1-Part5-Part2-Part4-Part3 and CNC assignments CNC2-CNC1-CNC3-CNC2-CNC1, with all five workpieces reaching final rack entry. Dispatch-order consistency, CNC-routing consistency and terminal-completion consistency are therefore each 5/5 (100%). The selected schedule and its q/g-based event mapping are thus reproduced throughout the recorded virtual process. This offline check tests whether a numerical schedule remains coherent after translation into resource-constrained actions and can reveal discrepancies in dispatch order, CNC routing or workstation handover be-

fore engineering evaluation. The results provide traceable, case-level virtual execution evidence for the selected schedule under the deterministic production-line configuration.

4.4 Case 2: offline search evaluation with 30 workpieces

Case 1 provides an exhaustive five-workpiece reference, whereas Case 2 evaluates offline search performance as the problem expands to 30 workpieces and three CNC machines. The joint search space is approximately $30! \times 3^{30}$. Case 2 compares the two GA objectives under a common configuration across 30 matched independent runs, with Rule v2 as a deterministic reference. The analysis focuses on full-process results, cross-run variation and search characteristics.

Table 5. Full-process results for Case 2 under matched GA settings

Method	Objective	Runs	Best C_{max}^{full}/s	Mean	Median	Worst	Std.	Mean runtime/s
Rule v2	Rule based	1	1104.60	/	/	/	/	0.0011
GA v2	C_{max}^{CNC}	30	1036.60	1119.31	1121.70	1187.40	35.44	78.73
GA v3	C_{max}^{full}	30	1053.70	1091.49	1090.15	1148.90	22.21	411.80

Table 5 shows clear differences between the two scheduling objectives in the 30-workpiece case. GA v2 records the lowest individual C_{max}^{full} value of 1036.60 s, while GA v3 achieves a best value of 1053.70 s. Across the 30 matched runs, GA v3 obtains a mean C_{max}^{full} of 1091.49 s, compared with 1119.31 s for GA v2, and reduces the median from 1121.70 s to 1090.15 s. Its worst-run value decreases from 1187.40 s to 1148.90 s, while the standard deviation decreases from 35.44 s to 22.21 s. These results indicate that the full-process objective improves the typical completion-time performance and produces a more concentrated distribution across repeated runs. The reduction in both the central tendency and run-to-run variation supports the use of C_{max}^{full} for evaluating schedules when downstream operations through final rack entry are included in the completion criterion.

The paired results further support this pattern. With identical seeds used for the two GA searches, GA v3 obtains a lower C_{max}^{full} in 25 of the 30 paired runs. The mean paired difference is 27.82 s and the median difference is 29.10 s in favour of the full-process objective. The two-sided Wilcoxon signed-rank test gives $p=0.0002$, together with a matched-pairs rank-biserial value of 0.7462, showing a consistent shift toward lower full-process completion times across the matched runs. Rule v2 gives $C_{max}^{full}=1104.60$ s, while the GA v3 mean and median values are 1091.49 s and 1090.15 s, respectively. The comparison with the deterministic rule reference provides additional evidence that full-process-oriented GA search can improve the overall completion performance of the 30-workpiece scheduling case.

The two objective functions also differ in computational cost. Under the same population size and generation setting, the mean runtime is 78.73 s for GA v2 and 411.80 s for GA v3. The additional computation required by GA v3 results from the full-process evaluation, which includes both CNC machining and the downstream execution stages used to calculate C_{max}^{full} . The measured runtimes provide a quantitative reference for the computational effort required when full-process information is incorporated into offline schedule evaluation.

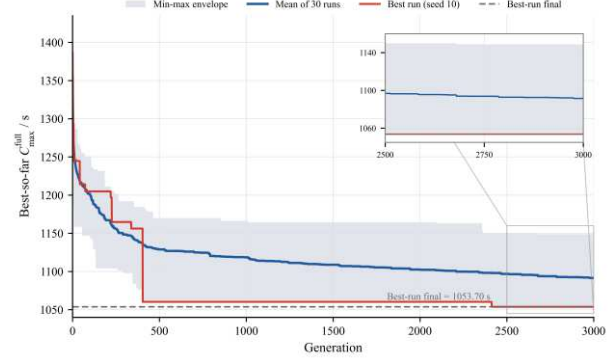


Fig. 9. Search progression of GA v3 under the full-process objective in Case 2.

The search progression of GA v3 under C_{max}^{full} is shown in Fig. 9. The mean best value decreases from 1096.80 s at generation 2500 to 1092.77 s at generation 2800 and reaches 1091.49 s at generation 3000. The improvement becomes progressively smaller in the later generations, indicating that the population has entered a relatively stable solution region. Four of the 30 runs still obtain further improvement during the final 100 generations, and 14 runs show additional refinement over the final 500 generations. These results show that the 3000-generation setting provides sufficient search depth for the population to reach a stable late-stage region while retaining the ability to refine individual solutions.

A limited parameter sensitivity analysis was conducted for GA v3 using five fixed seeds and 3000 generations for each configuration. Starting from the reported setting of population 160, crossover probability 0.85 and mutation probability 0.20, the population size was varied to 120 and 200, the crossover probability to 0.80 and 0.90, and the mutation probability to 0.15 and 0.25. The mean C_{max}^{full} values obtained from the tested settings range from 1052.92 s to 1101.08 s. The results show that changes in population size, crossover probability and mutation probability can influence the full-process scheduling outcome, with mutation probability showing a particularly visible effect within the tested range. This analysis provides a quantitative view of the sensitivity of GA v3 to bounded changes in its principal search parameters and supports the interpretation of the reported parameter setting within the tested configuration range.

5. Conclusions

This paper proposed PG-FVEV, a Petri-GA-based full-process virtual execution validation framework for a rail-mounted manipulator-CNC flexible manufacturing line. The framework links offline GA scheduling search, Petri-net resource constraints, q/g dispatch mapping and MIoT.VC virtual execution within a continuous

schedule-to-execution process. The GA determines the workpiece dispatching sequence and CNC assignments, the Petri-net layer constrains the use of shared production resources, and the q/g mapping converts the selected scheduling decision into event-level dispatch logic. This formulation enables a numerical scheduling result to be examined through offline virtual execution rather than remaining only an objective-function value.

The experimental evaluation further confirms the effectiveness of the proposed framework. In the five-workpiece case, exhaustive enumeration provides objective-specific references for the two completion criteria. GA v2 reaches the minimum C_{max}^{CNC} of 162.20 s, while GA v3 reaches the minimum C_{max}^{full} of 199.30 s and maintains concentrated repeated-run performance near this reference. The selected GA v3 schedule is subsequently reproduced in MIoT.VC, where the prescribed dispatching order, CNC routing and final rack completion remain fully consistent with the mapped scheduling result. Building on this finite-case validation, the 30-workpiece case evaluates GA v2 and GA v3 under matched search settings. GA v3 reduces the mean C_{max}^{full} from 1119.31 s to 1091.49 s and the standard deviation from 35.44 s to 22.21 s, showing improved full-process scheduling performance and more concentrated repeated-run results. These findings demonstrate that PG-FVEV can connect offline scheduling optimisation with resource-constrained execution logic and provide a practical basis for evaluating complete production-line schedules.

However, the present study remains limited to deterministic offline virtual cases. Future work could investigate disturbance-aware dynamic scheduling under uncertain event sequences [38]. Robust supervision for unreliable resources would require a separate failure model [39]. Physical-virtual feedback and controller-level integration would require online interfaces beyond the current PG-FVEV scope [31].

Acknowledgments

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declarations

Conflict of Interest

The author declares no competing interests.

Data availability

The data supporting the findings of this study are available from the author upon reasonable request.

References

[1] T. Liang, L. Zhou and Z. Jiang, Integrated scheduling

of production and material delivery for the intelligent manufacturing system, *International Journal of Production Research*, 63 (3) (2025) 882-903.

[2] Z. Shakeri, K. Benfriha, M. Varmazyar, E. Talhi and A. Quenehen, Production scheduling with multi-robot task allocation in a real industry 4.0 setting, *Scientific Reports*, 15 (1) (2025) 1795.

[3] M. Liu, J. Lv, S. Du, Y. Deng, X. Shen and Y. Zhou, Multi-resource constrained flexible job shop scheduling problem with fixture-pallet combinatorial optimisation, *Computers & Industrial Engineering*, 188 (2024) 109903.

[4] L. Ba, Y. Li and M. S. Yang, Modelling and Simulation of a Multi-Resource Flexible Job-Shop Scheduling, *International Journal of Simulation Modelling*, 15 (1) (2016) 157-169.

[5] L. Meng, W. Cheng, B. Zhang, W. Zou, W. Fang and P. Duan, An Improved Genetic Algorithm for Solving the Multi-AGV Flexible Job Shop Scheduling Problem, *Sensors*, 23 (8) (2023) 3815.

[6] N. Striffler and T. Voigt, Concepts and trends of virtual commissioning – A comprehensive review, *Journal of Manufacturing Systems*, 71 (2023) 664-680.

[7] S. Guo, T. Qiu, J. Guo, B. Du and K. Wang, A novel MOEA/D with Q-Learning initialization for dual-resource constrained flexible job shop scheduling problem with limited multi-skilled workers, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 240 (8) (2026) 1212-1228.

[8] X. Lyu, Y. Song, C. He, Q. Lei and W. Guo, Approach to Integrated Scheduling Problems Considering Optimal Number of Automated Guided Vehicles and Conflict-Free Routing in Flexible Manufacturing Systems, *IEEE Access*, 7 (2019) 74909-74924.

[9] P. Brandimarte, Routing and scheduling in a flexible job shop by tabu search, *Annals of Operations Research*, 41 (3) (1993) 157-183.

[10] Ü. Bilge and G. Ulusoy, A Time Window Approach to Simultaneous Scheduling of Machines and Material Handling System in an FMS, *Operations Research*, 43 (6) (1995) 1058-1070.

[11] S. Li, K. Han, Z. Pei, W. Yi and R. Lv, Green flexible job shop integrated scheduling optimization for machines and AGVs based on the INSGA-II algorithm, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 240 (4) (2026) 477-494.

[12] X. Han, W. Cheng, L. Meng, B. Zhang, K. Gao, C. Zhang and P. Duan, A dual population collaborative genetic algorithm for solving flexible job shop scheduling problem with AGV, *Swarm and Evolutionary Computation*, 86 (2024) 101538.

[13] X. Wu, J. Peng, X. Xiao and S. Wu, An effective approach for the dual-resource flexible job shop scheduling problem considering loading and unloading, *Journal of Intelligent Manufacturing*, 32 (3) (2021) 707-728.

[14] S. Yang, L. Meng, S. Ullah, C. Zhang, H. Sang and B. Zhang, MILP Modeling and Optimization of Three-Stage Flexible Job Shop Scheduling Problem with

- Assembly and AGV Transportation, *Chinese Journal of Mechanical Engineering*, 38 (1) (2025) 115.
- [15] I. A. Chaudhry, A. F. Rafique, I. A.-Q. Elbadawi, M. Aichouni, M. Usman, M. Boujelbene and A. Bou-djemline, Integrated scheduling of machines and automated guided vehicles (AGVs) in flexible job shop environment using genetic algorithms, *International Journal of Industrial Engineering Computations*, 13 (3) (2022) 343-362.
- [16] Q. Liu, Z. Gao, J. Li, S. Li and L. Zhu, Research on Optimization of Dual-Resource Batch Scheduling in Flexible Job Shop, *Computers, Materials & Continua*, 76 (2) (2023) 2503-2530.
- [17] J. Li, X. Li, L. Gao and Q. Liu, A Flexible Job Shop Scheduling Problem Considering On-Site Machining Fixtures: A Case Study From Customized Manufacturing Enterprise, *IEEE Transactions on Automation Science and Engineering*, 22 (2025) 8415-8426.
- [18] S. C. Burmeister, D. Guericke and G. Schryen, A memetic NSGA-II for the multi-objective flexible job shop scheduling problem with real-time energy tariffs, *Flexible Services and Manufacturing Journal*, 36 (4) (2024) 1530-1570.
- [19] J. Fan, C. Zhang, Q. Liu, W. Shen and L. Gao, An improved genetic algorithm for flexible job shop scheduling problem considering reconfigurable machine tools with limited auxiliary modules, *Journal of Manufacturing Systems*, 62 (2022) 650-667.
- [20] I.-B. Park and J. Huh, A genetic algorithm with feasibility-agnostic encoding and three-phase decoding for scheduling semiconductor manufacturing facilities under queue time limits, *Expert Systems with Applications*, 301 (2026) 130310.
- [21] J. Yang and H. Xu, Hybrid Memetic Algorithm to Solve Multiobjective Distributed Fuzzy Flexible Job Shop Scheduling Problem with Transfer, *Processes*, 10 (8) (2022) 1517.
- [22] T. Murata, Petri nets: Properties, analysis and applications, *Proceedings of the IEEE*, 77 (4) (1989) 541-580.
- [23] J. Ahn and H.-J. Kim, A Branch and Bound Algorithm for Scheduling of Flexible Manufacturing Systems, *IEEE Transactions on Automation Science and Engineering*, 21 (3) (2024) 4382-4396.
- [24] G. Xu and Y. Chen, Petri-Net-Based Scheduling of Flexible Manufacturing Systems Using an Estimate Function, *Symmetry*, 14 (5) (2022) 1052.
- [25] K. Yamalidou, J. Moody, M. Lemmon and P. Antsaklis, Feedback control of petri nets based on place invariants, *Automatica*, 32 (1) (1996) 15-28.
- [26] M. Bashir, J. Zhou and B. B. Muhammad, Optimal Supervisory Control for Flexible Manufacturing Systems Model With Petri Nets: A Place-Transition Control, *IEEE Access*, 9 (2021) 58566-58578.
- [27] J. Ezpeleta, J. M. Colom and J. Martinez, A Petri net based deadlock prevention policy for flexible manufacturing systems, *IEEE Transactions on Robotics and Automation*, 11 (2) (1995) 173-184.
- [28] C. Gu, Z. Ma and Z. Li, Liveness and deadlock-freeness verification and enforcement in bounded Petri nets using basis reachability graphs, *Automatica*, 164 (2024) 111625.
- [29] Y. Lu, Y. Chen, C. N. Hadjicostis and Z. Li, Efficient iterative deadlock prevention for flexible manufacturing systems utilizing modified resource requirement graphs, *Automatica*, 183 (2026) 112631.
- [30] Z. Guo, Y. Zhang, X. Zhao and X. Song, A Timed Colored Petri Net Simulation-Based Self-Adaptive Collaboration Method for Production-Logistics Systems, *Applied Sciences*, 7 (3) (2017) 235.
- [31] G. Barbieri, A. Bertuzzi, A. Capriotti, L. Ragazzini, D. Gutierrez, E. Negri and L. Fumagalli, A virtual commissioning based methodology to integrate digital twins into manufacturing systems, *Production Engineering*, 15 (3-4) (2021) 397-412.
- [32] A. Ghasemi, F. Farajzadeh, C. Heavey, J. Fowler and C. T. Papadopoulos, Simulation optimization applied to production scheduling in the era of industry 4.0: A review and future roadmap, *Journal of Industrial Information Integration*, 39 (2024) 100599.
- [33] M. Raza, A. Bilberg and J. P. De Oliveira Hansen, Multiple-level virtual commissioning in manufacturing systems: possibilities and challenges for integration, *Procedia CIRP*, 134 (2025) 396-400.
- [34] A. Wooley, J. Bitencourt and D. Silva, Bridging the gap between discrete event simulation and digital twin: A manufacturing case study, *Manufacturing Letters*, 44 (2025) 1274-1284.
- [35] E. Negri, V. Pandhare, L. Cattaneo, J. Singh, M. Macchi and J. Lee, Field-synchronized Digital Twin framework for production scheduling with uncertainty, *Journal of Intelligent Manufacturing*, 32 (4) (2021) 1207-1228.
- [36] J. E. Abu Qudeiri, Production simulator system for flexible routing optimization in flexible manufacturing systems, *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231 (7) (2017) 1237-1247.
- [37] Y. Fu, K. Gao, L. Wang, M. Huang, Y.-C. Liang and H. Dong, Scheduling stochastic distributed flexible job shops using an multi-objective evolutionary algorithm with simulation evaluation, *International Journal of Production Research*, 63 (1) (2025) 86-103.
- [38] D. Lefebvre, Dynamical Scheduling and Robust Control in Uncertain Environments with Petri Nets for DESs, *Processes*, 5 (4) (2017) 54.
- [39] U. S. Abubakar, G. Liu and M. Uzam, Petri Net-Based Robust Supervisory Control of Automated Manufacturing Systems With Multiple Unreliable Resources, *IEEE Access*, 9 (2021) 100264-100278.

Author information

Bowen Zhang



Bowen Zhang is a master's student at the School of Mechanical Engineering, Shenyang Ligong University, Shenyang, China. His research interests include advanced CNC technology and its applications.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SupplementaryMaterialssubmissionfinal.docx](#)
- [AppendixFigA1completeG2cleaned600dpi.png](#)