

A Minimax Mathematical Programming Framework For Robust Production Scheduling In Smart Manufacturing Systems

Dr. Maharshikumar B. Shukla¹, Dr. Sanjay G Sakharwade², Alok Kumar³, M. Anitha⁴,
Leena Deshpande⁵, Amanveer Singh⁶, S. Sathiya Raj⁷

¹Associate Professor, Faculty of Science, Gokul Global University, Sidhpur, Gujarat, India.
maharshishukla.chem.gsc@gokuluniversity.ac.in, <https://orcid.org/0009-0004-9071-023X>

²Rungta College of Engineering and Technology, Rungta International Skills University, Bhilai,
Chhattisgarh, India. sanjay.sakharwade@rungta.org

³Assistant Professor, Department of School of Engineering & Technology, Noida International
University, Uttar Pradesh. alok.kumar1@niu.edu.in

⁴Assistant Professor, Department of Mathematics, Meenakshi College of Arts and Science,
Meenakshi Academy of Higher Education and Research, Chennai, Tamil Nadu, India.
anitham@maher.ac.in

⁵Associate Professor, Department of Computer Engineering - Software Engineering,
Vishwakarma Institute of Technology, Pune, Maharashtra. leena.deshpande@vit.edu

⁶Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and
Technology, Chitkara University, Rajpura, Punjab, India. amanveer.singh.orp@chitkara.edu.in,
<https://orcid.org/0009-0008-9361-4664>

⁷Assistant Professor, Gr. II, Department of Mechanical Engineering, Aarupadai Veedu Institute
of Technology, Vinayaka Mission's Research Foundation (DU), Tamil Nadu, India,
sathiyaraj@avit.ac.in, <https://orcid.org/0000-0001-8504-0412>

Abstract

Smart manufacturing systems should have production schedules that are still efficient despite variations in processing times and machine availability from their nominal values. In this paper, propose a minimax mathematical programming framework for robust production scheduling in smart manufacturing systems to explicitly minimize the maximum possible production loss on account of bounded uncertainties. The framework includes job assignment, machine sequencing, start time assignment, tardiness, makespan, and machine capacity constraints within an integrated mixed-integer minimax model. Uncertainties in processing times and machine availability are described by budgeted polyhedral uncertainty sets so that the conservativeness of the robust schedule can be regulated by uncertainty budget parameters. Unlike approaches that consider the uncertainty effect on the objective as an abstract perturbation, this approach explicitly accounts for the propagation of processing time deviations on completion times, tardiness, makespan, and machine capacity requirements. A dual reformulation for the joint uncertain capacity constraints is derived. To solve this problem, a column-and-constraint generation algorithm is devised where a mixed-integer master scheduling problem is repeatedly solved with respect to a continuous linear adversarial subproblem. Studies prove theoretical properties, including feasibility, monotonicity of the robust objective, asymptotic recovery to the deterministic limit, and finite convergence of the decomposition algorithm. The numerical experiments include production problems with configurations of 12×4 to 100×20 jobs-machines and compare the proposed approach with deterministic, stochastic, traditional robust, and dynamic MPC approaches. These results illustrate the superiority of the proposed minimax framework for uncertainty-aware smart manufacturing scheduling.

Keywords: Minimax optimization, robust scheduling, mathematical programming, smart manufacturing, uncertainty modeling, column-and-constraint generation.

1. Introduction

Smart manufacturing has turned production scheduling into a dynamic mathematical optimization problem. Contemporary production systems have interconnected machines, flexible manufacturing processes, varying production requirements, and dynamically updated production data. Thus, the scheduling process should take into account not only the issues of assigning and sequencing the jobs but also the changes in the machine status, processing time, and availability of resources. The possibility of using the dynamic model predictive control for production scheduling in the environment of smart manufacturing has shown the necessity of accounting for the variation in the state of production processes (Bozzi et al., 2023). Perspectives of optimal control have also been applied to the problems of scheduling in production, supply chains, and Industry 4.0 systems, showing the dynamic nature of modern manufacturing (Dolgui et al., 2019).

A significant challenge of such systems is their uncertainty. Processing times may be different from the nominal, machines can become partially unavailable, and the production state can vary during the execution of the scheduling plan. The operation of the production system in the cloud and its interaction with the supply chain make the production decisions dependent on the external operational state and increase the need for resilient scheduling (Kodirova et al., 2025). An optimal schedule under nominal assumptions, therefore, may lead to higher tardiness, makespan, or poor machine utilization.

Decision-making under the worst-case scenario is one mathematical way of solving such problems. The minimax regret technique has been used for semiconductor capacity expansion under demand uncertainty, showing the applicability of worst-case scenario-oriented optimization in cases where future production needs are unknown (Kuo & Chien, 2023). In much the same way, the use of robust production scheduling should aim not to achieve the optimal schedule for a single expected situation but rather a schedule that performs acceptably under a predetermined set of negative but acceptable scenarios.

The nature of uncertainty is made even more complex in flexible manufacturing systems. There exist interactions between the tasks, machines, and production resources, and this leads to unpredictable results (Bozzi et al., 2023). Maintenance operations are another instance where the correlation between machine state and production performance is shown; this is because machinery breakdowns or maintenance needs may affect production resource availability (Juma & Mdodo, 2024). Simultaneously, real-time integration of planning, scheduling, and execution has become more important in Industry 5.0 manufacturing environments (Li et al., 2025). Such considerations show that robust scheduling should take into account operational uncertainty and not assume deterministic production parameters.

Optimization methodologies have been studied extensively to solve the scheduling problem complexity that arises. Intelligent optimization has been used in relation to flexible manufacturing cell layouts and task scheduling (Luedke & Kingdone, 2025; Suvarna & Bharadwaj, 2024). Integrated planning and scheduling methodologies have been developed to account for the interaction between production planning and scheduling decisions (Marzia et al., 2025). Sustainable scheduling has further included energy availability and demand-side management in production decisions (Jabeur et al., 2023). Recent methodologies have further investigated the development of generative methods for robust job shop scheduling for CNC manufacturing systems (Truong et al., 2025). Though all of the above methodologies highlight the success of advanced optimization and computational methodologies, a robust minimax mathematical methodology still seems vital when protection from worst-case scenarios is required.

From a mathematical programming perspective, robust production scheduling consists of two interconnected parts. The first part refers to the decision-making problem of assigning machines to a particular job, deciding on the job sequence, start time, completion time, and the use of resources. The second part relates to the uncertainty evaluation problem where the extent of damage due to admissible changes of processing conditions and machines has to be determined. It has been highlighted in robustness evaluation literature the importance of assessing scheduling solutions in the light of uncertain parameters (Himmiche et al., 2023). Moreover, smart production scheduling systems emphasize the importance of integrating production decisions with uncertain operational and supply chain conditions (Serrano-Ruiz et al., 2021).

The problem at hand becomes computationally difficult to solve due to the presence of both discrete and continuous elements. Assignments and sequencing constitute combinatorial optimization problems, while processing time, machine availability, completion times, and uncertain parameters are all continuous. Research into metaheuristics has already been conducted for discrete optimization problems, but industrial big-data and cyber-physical-systems-based approaches have been applied to adaptive production scheduling (Shirke & Udayakumar, 2019; Qiao et al., 2021). In addition to that, however, there is no guarantee that this approach will provide a precise mathematical formulation of the worst-case realization of uncertainty as well as the quantitative link between the amount of uncertainty and the

quality of scheduling.

The issue of robustness also depends on the reliability of the information system underpinning the entire process of smart manufacturing. Analysis of the robustness of the smart manufacturing information system shows that changes in availability of information and conditions can affect the reliability of production decisions (Zhu & Zheng, 2024). Manufacturing problems are complex enough in nature, as they can contain several different criteria, constraints, and dependencies, which make achieving reliable schedules even harder (Ma et al., 2023). Predictive control methods combining scheduling and maintenance have been considered for open-shop manufacturing systems (Bozzi et al., 2024). These developments reinforce the need for mathematical scheduling frameworks that can provide robustness while retaining a clear and interpretable optimization structure.

With advancements in artificial intelligence and data-driven techniques, the capabilities of smart manufacturing have been further enhanced. However, AI-based manufacturing systems may encounter issues of data availability, generalization of the model, dynamic changes in the production environment, and computation needs (Xu et al., 2022). For these reasons, mathematical optimization still serves as an important alternative decision-making approach. Specifically, a robust mathematical model is capable of providing worst-case guarantees and explainable scheduling solutions when data-driven technologies provide insights into the present production status.

In response to the above needs, this paper proposes a minimax mathematical programming approach for robust production scheduling in smart manufacturing systems. The presented minimax formulation considers the production scheduling problem as a bi-level optimization model where the upper-level problem establishes the production schedule, and the lower-level problem finds the worst admissible scenario of uncertainty. Processing-time and machine-availability uncertainties are included in the scheduling structure to consider their impact on completion times, tardiness, makespan, and machine capacity utilization. The budget uncertainty sets are used to tune the conservatism, while the dual reformulation technique is applied to derive a tractable robust counterpart.

The main contributions of this study are:

- **Minimax mathematical formulation:** A generic minimax programming formulation is devised to minimize maximum production loss due to uncertain operations.
- **Explicit uncertainty propagation:** Uncertain processing times and machine availabilities are explicitly integrated into completion time, lateness, makespan, and capacity considerations.
- **Robust counterpart derivation:** Mathematical duality is exploited to transform uncertain joint capacity conditions to a robust formulation.
- **Theoretical analysis:** Properties such as feasibility, optimality, budget monotonicity, deterministic limit, and robust dominance are examined.
- **Decomposition-based solution:** A column-and-constraint generation algorithm is designed for separation of the scheduling master problem and the continuous adversarial uncertainty subproblem.
- **Numerical and engineering evaluation:** Performance evaluation is conducted in terms of benchmark comparison, budget sensitivity, convergence analysis, scalability study, and robust scheduling decisions.

The rest of the paper is organized as follows. In Section 2, the mathematical preliminaries, notations, production scheduling setting, and uncertainty are discussed. The proposed minimax mathematical programming formulation and its robust version are developed in Section 3. The theoretical analysis and solution approach, along with their convergence properties, are discussed in Section 4. Section 5 includes numerical examples, sensitivity analysis, scalability assessment, and the engineering application of smart manufacturing. Section 6 draws the conclusions and future directions in mathematics.

2. Mathematical preliminaries and problem definition

This subsection introduces the mathematical formulation of the robust production scheduling problem. Let us take the case of a production environment in which there is a finite number of jobs $J = \{1, \dots, n\}$ and machines $M = \{1, \dots, m\}$. Jobs need to be assigned to a suitable machine and processed without interruption while meeting their due dates and capacity constraints.

2.1 Notation and decision variables

Let $x_{ij} \in \{0,1\}$ indicate whether job j is assigned to machine i , and let $y_{jk}^i \in \{0,1\}$ indicate whether job j precedes job

kon machine i . The variables S_j and C_j denote the starting and completion times of job j , respectively, while T_j denotes its tardiness. The production makespan is represented by $C_{\max}$.

Each job must be assigned to exactly one eligible machine:

$$\sum_{i \in M_j} x_{ij} = 1, \quad \forall j \in J \quad (1)$$

The completion time is related to the starting time and processing duration by:

$$C_j = S_j + p_j, \quad \forall j \in J \quad (2)$$

Tardiness is defined by:

$$T_j \geq C_j - d_j, \quad T_j \geq 0, \quad \forall j \in J \quad (3)$$

while the makespan satisfies:

$$C_{\max} \geq C_j, \quad \forall j \in J \quad (4)$$

Therefore, the equations (1)–(4) provide the initial setting up of the scheduling problem.

2.2 Deterministic scheduling formulation

At normal conditions of operation, let $\bar{p}_j$ be the normal processing time and $\bar{a}_i$ be the normal availability of machine i . The weighted objective function is used to incorporate the cost of tardiness, makespan, and resources (Equation 5):

$$F(x, \bar{\xi}) = \lambda_1 \sum_{j \in J} T_j + \lambda_2 C_{\max} + \lambda_3 \sum_{i \in M} I_i \quad (5)$$

where I_i denotes machine idle time and $\lambda_1, \lambda_2, \lambda_3 \geq 0$ are weighting coefficients. The corresponding deterministic problem is:

$$\min_{x \in \mathcal{X}} F(x, \bar{\xi}) \quad (6)$$

where $\mathcal{X}$ is the feasible scheduling region. As it can be seen from equation (6), the optimization of the schedule has been done only under normal operating conditions.

2.3 Scheduling uncertainty

The processing time is considered to be an uncertain variable. The processing time of job j is given by equation 7:

$$p_j(\xi_j) = \bar{p}_j(1 + \delta_j \xi_j), \quad 0 \leq \xi_j \leq 1 \quad (7)$$

where δ_j represents the maximum relative processing-time deviation.

Machine availability is similarly modeled as equation 8:

$$a_i(\eta_i) = \bar{a}_i - \Delta a_i \eta_i, \quad 0 \leq \eta_i \leq 1 \quad (8)$$

Here, Δa_i denotes the maximum value of lost machine capacity. Therefore, the constraint on machine capacity will be

$$\sum_{j \in J} p_j(\xi_j) x_{ij} \leq a_i(\eta_i), \quad \forall i \in M \quad (9)$$

Equation (9) explicitly accounts for uncertainties in processing times and machine availability.

2.4 Uncertainty set

A budget of uncertainty approach is taken, which does not make the assumption that all uncertain parameters have the maximum deviation at once. The uncertainty set for processing time is as equation 10:

$$\mathcal{U}_p = \{\xi: 0 \leq \xi_j \leq 1, \sum_{j \in J} \xi_j \leq \Gamma_p\} \quad (10)$$

and the machine-availability uncertainty set is equation 11:

$$\mathcal{U}_a = \{\eta: 0 \leq \eta_i \leq 1, \sum_{i \in M} \eta_i \leq \Gamma_a\} \quad (11)$$

The complete uncertainty region is therefore equation 12:

$$\mathcal{U} = \mathcal{U}_p \times \mathcal{U}_a \quad (12)$$

Γ_p and Γ_a are responsible for determining the amount of uncertainty involved in the model. Higher values allow for a higher number of possible deviations and hence lead to a more conservative scheduling strategy.

2.5 Minimax optimization principle

The worst-case loss associated with any scheduling decision x is represented in equation (13), based on the uncertainty set as defined in equation (12):

$$G(x) = \max_{\xi \in \mathcal{U}} F(x, \xi) \quad (13)$$

Based on this, the robust scheduling problem can be formulated as

$$\min_{x \in \mathcal{X}} \max_{\xi \in \mathcal{U}} F(x, \xi) \quad (14)$$

Equation (14) is the main minimax construction of the proposed model. The inner maximization problem finds the worst admissible instance of uncertainty in production process, while the outer minimization problem picks up the feasible schedule with minimum possible worst-case loss.

2.6 Problem definition

The study focuses on the issue of computing a feasible manufacturing schedule for a smart manufacturing system, considering processing times and machine availability under uncertainty. The jobs should be assigned to an appropriate machine and sequenced such that there are no violations of temporal and capacity constraints.

As opposed to the deterministic scheduling problem, where the solution is optimized for nominal parameters only, the problem formulated here assesses each feasible schedule within a limited region of uncertainty and selects the one that has minimal worst-case scheduling loss. Hence, the formulated problem turns out to be a mixed-integer minimax optimization problem combining discrete decision-making and adversarial uncertainty.

3. Proposed minimax mathematical programming framework

According to the suggested approach, robust scheduling of production is modeled as a two-level minimax optimization problem. Specifically, the upper level defines machine-job assignment, production schedule, and planned start times, while the lower adversarial level finds the worst possible realization of the uncertainty in processing time and machine availability. In contrast to a deterministic scheduling problem, the suggested modeling approach explicitly accounts for the propagation of processing time deviations to completion times, tardiness, and makespan.

3.1 Integrated assignment and sequencing model

Let $x_{ij} \in \{0,1\}$ indicate whether job j is assigned to machine i , and let $y_{jki} \in \{0,1\}$ indicate whether job j precedes job k on machine i . The assignment requirement is:

$$\sum_{i \in M_j} x_{ij} = 1, \quad \forall j \in J \quad (15)$$

Equation (15) assures that each job is assigned to exactly one feasible machine. Non-overlapping sequencing for two jobs assigned to the same machine is enforced using:

$$S_k \geq S_j + p_j(\xi)x_{ij} - M_0(1 - y_{jki}), \quad \forall j, k \in J, j \neq k \quad (16)$$

and

$$S_j \geq S_k + p_k(\xi)x_{ik} - M_0 y_{jki}, \quad \forall j, k \in J, j \neq k \quad (17)$$

Equations (16) and (17) prevent the simultaneous processing of two jobs on the same machine and incorporate the uncertainty of processing time. M_0 is chosen to be large enough so that the infeasible disjunctive constraint is disabled. The sequencing binary variable is turned ON if both jobs use the same machine i :

$$y_{jki} + y_{kji} \geq x_{ij} + x_{ik} - 1, \quad \forall i \in M, j, k \in J, j \neq k \quad (18)$$

Equation (18) makes sure that one processing sequence is chosen if there are two jobs on the same machine.

3.2 Propagation of processing-time uncertainty

The nominal processing duration of job j is $\bar{p}_j$. Its uncertain realization is:

$$p_j(\xi_j) = \bar{p}_j(1 + \delta_j \xi_j), \quad 0 \leq \xi_j \leq 1 \quad (19)$$

Equation (19) provides a direct definition of the source of uncertainty that exists in the scheduling model, where δ_j is the maximum relative deviation.

For a fixed assignment, the uncertain completion time for job j is:

$$C_j(\xi) = S_j + \bar{p}_j(1 + \delta_j \xi_j) \quad (20)$$

The relationship between uncertain processing times and completion time is provided in equation (20). The associated job tardiness is expressed through:

$$T_j(\xi) \geq C_j(\xi) - d_j, \quad T_j(\xi) \geq 0, \quad \forall j \in J \quad (21)$$

Substituting equation (20) into equation (21) gives:

$$T_j(\xi) \geq S_j + \bar{p}_j - d_j + \bar{p}_j \delta_j \xi_j \quad (22)$$

Equation (22) clearly shows how the uncertainty term ξ_j gets transmitted into job lateness. So, the sensitivity coefficient related to ξ_j becomes clear from $\bar{p}_j \delta_j$, together with its impact on sequencing and processing time.

The makespan is specified by equation 23:

$$C_{\max}(\xi) \geq C_j(\xi), \quad \forall j \in J \quad (23)$$

or equivalently:

$$C_{\max}(\xi) \geq S_j + \bar{p}_j + \bar{p}_j \delta_j \xi_j, \quad \forall j \in J \quad (24)$$

Equation (24) clearly relates processing time uncertainty to the makespan part of the objective function.

3.3 Uncertain scheduling objective

Uncertain scheduling loss includes job tardiness, total makespan, and machine idle time:

$$F(x, y, S, \xi) = \lambda_1 \sum_{j \in J} T_j(\xi) + \lambda_2 C_{\max}(\xi) + \lambda_3 \sum_{i \in M} I_i(\xi) \quad (25)$$

$\lambda_1, \lambda_2, \lambda_3 \geq 0$ are non-negative weights. equation (25) is tightly related to equations (20)–(24). So, uncertainty in processing times alters completion times, which, in turn, affect tardiness and makespan.

For a given scheduling scenario (x, y, S) , the worst-case performance estimation is defined by:

$$\Psi(x, y, S) = \max_{(\xi, \eta) \in \mathcal{U}} F(x, y, S, \xi) \quad (26)$$

This equation 26 is the precise formulation of inner maximization within the minimax paradigm.

3.4 Machine-availability uncertainty and robust capacity

Machine availability is modeled under capacity reduction as equation 27:

$$a_i(\eta_i) = \bar{a}_i - \Delta a_i \eta_i, \quad 0 \leq \eta_i \leq 1 \quad (27)$$

The uncertain machine-capacity requirement is:

$$\sum_{j \in J} \bar{p}_j (1 + \delta_j \xi_j) x_{ij} \leq \bar{a}_i - \Delta a_i \eta_i, \quad \forall i \in M \quad (28)$$

Equation (28) represents joint uncertainty in processing time and machine availability. By arranging the terms appropriately, have equation 29.

$$\sum_{j \in J} \bar{p}_j x_{ij} + \sum_{j \in J} \bar{p}_j \delta_j x_{ij} \xi_j + \Delta a_i \eta_i \leq \bar{a}_i, \quad \forall i \in M \quad (29)$$

For each machine i , the robust capacity counterpart subject to budget parameters Γ_p and Γ_a is:

$$\max_{\substack{0 \leq \xi_j \leq 1, \sum \xi_j \leq \Gamma_p \\ 0 \leq \eta_i \leq 1, \eta_i \leq \Gamma_a}} \{ \sum_{j \in J} \bar{p}_j \delta_j x_{ij} \xi_j + \Delta a_i \eta_i \} \leq \bar{a}_i - \sum_{j \in J} \bar{p}_j x_{ij}, \quad \forall i \in M \quad (30)$$

Equation (30) represents the exact constraint wise robust counterpart of the uncertain capacity constraint.

3.5 Dual reformulation of robust capacity

For each machine i , introduce dual variables $\rho_i \geq 0$, $\mu_{ij} \geq 0$, and $\sigma_i \geq 0$. The linear programming dual counterpart of Equation (30) is written as:

$$\Gamma_p \rho_i + \sum_{j \in J} \mu_{ij} + \Gamma_a \sigma_i \leq \bar{a}_i - \sum_{j \in J} \bar{p}_j x_{ij}, \quad \forall i \in M \quad (31)$$

subject to:

$$\rho_i + \mu_{ij} \geq \bar{p}_j \delta_j x_{ij}, \quad \forall i \in M, j \in J \quad (32)$$

$$\sigma_i \geq \Delta a_i, \quad \forall i \in M \quad (33)$$

$$\rho_i \geq 0, \quad \mu_{ij} \geq 0, \quad \sigma_i \geq 0, \quad \forall i \in M, j \in J \quad (34)$$

The equations (31)-(34) provide an explicit dualization of the joint uncertainty in capacity, allowing for independent budget monitoring while maintaining the constraint-level effects simultaneously.

3.6 Exact adversarial subproblem

For a given schedule $(x^{(r)}, y^{(r)}, S^{(r)})$, the adversarial subproblem provides the worst possible value of uncertainty. The explicit linear program can be stated as

$$\max_{\xi, \eta, T, C, C_{\max}} [\lambda_1 \sum_{j \in J} T_j + \lambda_2 C_{\max} + \lambda_3 \sum_{i \in M} I_i] \quad (35)$$

subject to:

$$C_j = S_j^{(r)} + \bar{p}_j (1 + \delta_j \xi_j), \quad \forall j \in J \quad (36)$$

$$T_j \geq C_j - d_j, \quad T_j \geq 0, \quad \forall j \in J \quad (37)$$

$$C_{\max} \geq C_j, \quad \forall j \in J \quad (38)$$

$$0 \leq \xi_j \leq 1, \quad \sum_{j \in J} \xi_j \leq \Gamma_p \quad (39)$$

$$0 \leq \eta_i \leq 1, \quad \sum_{i \in M} \eta_i \leq \Gamma_a \quad (40)$$

Since the scheduling decision variables are frozen to $(x^{(r)}, y^{(r)}, S^{(r)})$, equations (35)-(40) become an LP in continuous space. Therefore, binary variables do not exist in the inner worst-case evaluation process.

3.7 Minimax master problem

Let us use $\mathcal{K}$ to denote the set of possible adversary scenarios found iteratively. Then, the restricted master problem is (Equation 41)

$$\min_{x, y, S, z} z \quad (41)$$

subject to:

$$z \geq F(x, y, S, \xi^k), \quad \forall k \in \mathcal{K} \quad (42)$$

subject to the assignment constraint in (15), sequencing constraints in (16)-(18), temporal logic, and robust capacity constraints in (31)-(33).

Equation (42) makes the epigraph variable z dominate the scheduling loss in all accumulated scenarios in $\mathcal{K}$.

4. Theoretical analysis and solution method

The approach to solve the formulation is via a column-and-constraint generation procedure. The master problem solves for the scheduling decisions, and the adversarial linear program computes the worst-case uncertainties.

4.1 CCG initialization

The initial scenario is selected as the nominal operating state equation 43:

$$\xi^0 = 0, \quad \eta^0 = 0 \quad (43)$$

The initial scenario set is constructed as equation 44:

$$\mathcal{K}_0 = \{(\xi^0, \eta^0)\} \quad (44)$$

Solving the master problem over $\mathcal{K}_0$ yields an initial feasible production schedule (x^0, y^0, S^0) .

4.2 Master problem step

At iteration r , the restricted master problem solves equation 45:

$$\min_{x, y, S, z} z \quad (45)$$

subject to equation 46:

$$z \geq F(x, y, S, \xi^k), \quad \forall k \in \mathcal{K}_r \quad (46)$$

along with all problem-specific feasibility and robust capacity constraints. The resulting optimal objective value will give a lower bound LB_r of the entire minimax problem.

4.3 Adversarial subproblem step

Fixing the incumbent schedule $(x^{(r)}, y^{(r)}, S^{(r)})$, the adversarial LP in equations (34)–(39) is solved to determine the maximum scheduling loss UB_r (Equation 47).

If:

$$UB_r - LB_r \leq \varepsilon \quad (47)$$

where $\varepsilon > 0$ is a predefined convergence tolerance, the procedure terminates with the incumbent schedule. Otherwise, the new worst-case realization (ξ^{r+1}, η^{r+1}) is generated via (Equation 48):

$$\mathcal{K}_{r+1} = \mathcal{K}_r \cup \{(\xi^{r+1}, \eta^{r+1})\} \quad (48)$$

4.4 Finite-convergence theorem

Theorem 2. *If it is known that the scheduling feasible region consists of a finite number of integer assignment sequences and the uncertainty set $\mathcal{U}$ forms a bounded polytope, then the proposed CCG method is guaranteed to stop after a finite number of iterations and, on termination (at zero optimality gap $\varepsilon = 0$), yields an optimal minimax schedule.*

Proof. The uncertainty set $\mathcal{U}$ is a bounded polytope with a finite number of extreme points. The adversarial subproblem in equations (34)–(39) achieves its optimal value at an extreme point of the set of all possible points within its feasible region. It means that there exist only finitely many different worst-case scenarios.

At each iteration, except for the terminal one, the gap condition in equation (46) does not hold, and the adversarial scenario (ξ^{r+1}, η^{r+1}) is appended to $\mathcal{K}_{r+1}$. The new scenario creates an additional valid constraint in the master problem, and the current sub-optimal solution of the master problem is strictly excluded from the feasible region of the whole uncertainty set.

Since the number of the extreme points of the uncertainty set is finite, it is impossible to generate infinitely many scenarios. Hence, in a finite number of iterations, the convergence condition in equation (46) holds. If $\varepsilon = 0$, $LB_r = UB_r$ and it ensures the optimality of the incumbent schedule.

4.5 Algorithmic procedure

Algorithm 1: minimax CCG scheduling procedure

1. **Initialize:** Set iteration index $r \leftarrow 0$. Define tolerance $\varepsilon > 0$. Initialize scenario set $\mathcal{K}_0 \leftarrow \{(0,0)\}$.
2. **Master Step:** Solve the restricted master problem in equations (44)–(45) to obtain schedule $(x^{(r)}, y^{(r)}, S^{(r)})$ and lower bound LB_r .
3. **Subproblem Step:** Fix $(x^{(r)}, y^{(r)}, S^{(r)})$. Solve the adversarial LP in equations (34)–(39) to obtain worst-case realization (ξ^{r+1}, η^{r+1}) and upper bound UB_r .
4. **Convergence Check:** Compute Optimality Gap = $UB_r - LB_r$.
 - **IF** $UB_r - LB_r \leq \varepsilon$ **THEN**
 - Terminate and output optimal schedule $(x^*, y^*, S^*) \leftarrow (x^{(r)}, y^{(r)}, S^{(r)})$.
 - **ELSE**
 - Update scenario set $\mathcal{K}_{r+1} \leftarrow \mathcal{K}_r \cup \{(\xi^{r+1}, \eta^{r+1})\}$.
 - Set $r \leftarrow r + 1$ and return to Step 2.

4.6 Computational complexity

Computational complexity is dictated by the master problem due to the combinatorial characteristic of the binary assignment (x_{ij}) and sequence (y_{jki}) decision variables. For a scheduling problem that has n jobs and m machines, the potential number of sequence decision variables increases at $O(n^2m)$.

On the other hand, the adversarial subproblem becomes a continuous linear programming problem when the decision from the master problem is determined, and the complexity of the problem is polynomial in the number of continuous uncertainty decision variables and constraints. The CCG algorithm does not enumerate all the possible continuous uncertainty space $\mathcal{U}$.

5. Numerical results and engineering application

This numerical assessment extends from a particular small-scale scheduling problem into a wider range to evaluate the quality of the solutions, as well as scalability, reliability, and computational efficiency. There are four benchmark scales included in the analysis: 12×4 , 30×10 , 50×20 , and 100×20 . In each pair of numbers, the first one denotes jobs, and the second one—machines. The smallest scale is used for mathematical proof.

5. Numerical experiments and computational analysis

This section outlines the experimental validation, benchmark testing, sensitivity analysis of uncertainties, scalability analysis, and applications of the proposed minimax Column-and-Constraint Generation (CCG) approach.

5.1 Experimental configuration

Numerical experiments are conducted on heterogeneous machines with job-related eligibility restrictions. Processing times are simulated around nominal values with deviations limited up to 20%, and machine availability can be reduced up to 15%. Uncertainty budgets are changed systematically to assess the effect of robustness. The parameters of the objective function are given as follows:

$$\lambda_1 = 0.45, \quad \lambda_2 = 0.35, \quad \lambda_3 = 0.20 \quad (49)$$

Equation (49) determines the largest weight on total tardiness, followed by makespan and machine idle time.

The CCG method stops if:

$$\varepsilon = 10^{-4} \quad (50)$$

Equation (50) acts as a termination criterion for all numerical experiments.

To ensure reproducibility, all benchmark instances are solved using the same mathematical model and solver settings. The following performance indicators are measured: nominal value of the objective function, worst-case value of the objective function, total tardiness, makespan, total number of iterations of the CCG algorithm, and CPU time.

5.2 Benchmark methods

The suggested Minimax-CCG method is compared with four other scheduling approaches:

1. **Deterministic Mathematical Programming (D-MP):** Provides an optimal schedule under nominal values of the process time and machine availability without any protection against uncertainty.
2. **Scenario-Based Stochastic Programming (SP):** Determines an optimal schedule that minimizes the average scheduling cost under a finite number of scenarios of uncertainty.
3. **Traditional Robust Scheduling (TRS):** Guarantees the system's operation for bounded parameter changes with a static robust approach without iterations and CCG decomposition.
4. **Dynamic Model Predictive Control (MPC):** Implements receding horizon scheduling to adapt the production schedule to the occurring perturbations (Bozzi et al., 2023).

5.3 Small-Scale Mathematical Verification

The 12×4 case study (12 jobs, 4 machines) is used to test the underlying mathematical performance of the proposed framework. In normal operating conditions, the deterministic model has a solution value of 34.72, while the robust model has one of 36.15.

In cases of worst-case uncertainties, the deterministic solution deteriorates into 48.73, while the robust solution has a worst-case value of 41.26.

The resulting price of robustness ("POR") is

$$\text{POR} = \frac{36.15 - 34.72}{34.72} \times 100 = 4.12\% \quad (51)$$

The Worst-Case Deterioration Reduction (WDR) is:

$$\text{WDR} = \frac{48.73 - 41.26}{48.73} \times 100 = 15.33\% \quad (52)$$

From equations (51) and (52), it can be seen that there is a reduction of 15.33% in the worst-case cost deterioration with a nominal cost increase of only 4.12%.

5.4 Benchmark Comparison

The benchmarking comparison is done based on identical uncertainty realizations and scheduling constraints among all competing approaches. In fact, the D-MP model provides the best nominal objective since it does not take uncertainty into consideration at all; yet, it fails dramatically when the operation environment turns hazardous. The SP approach offers better uncertainty protection in multiple scenarios, but it seeks to minimize the expected objective instead of the worst case. The TRS approach guarantees the best worst-case protection among all three approaches but with a higher nominal cost due to its more conservative way of handling uncertainty. The MPC strategy can make decisions adaptively according to varying conditions, yet it does not guarantee the minimum worst-case objective explicitly. The proposed Minimax-CCG framework tries to minimize the worst-case objective directly while generating the minimum number of adversarial scenarios necessary for the CCG process.

The comparison results for the 12×4 production problem is shown in table 1.

Table 1: Benchmark comparison of scheduling methods for the 12×4 manufacturing instance.

Method	Nominal Objective	Worst-Case Objective	Robustness Cost (%)
D-MP	34.72	48.73	0.00
SP	35.46	44.91	2.13
TRS	37.02	42.18	6.63
MPC	36.81	43.57	6.02
Proposed Minimax-CCG	36.15	41.26	4.12

Table 1 shows that the proposed Minimax-CCG algorithm has the best worst-case objective of 41.26 compared to other algorithms. Despite having an objective that is 4.12% greater than the deterministic model's objective, the proposed approach has a much better worst-case objective. The TRS approach yields an objective slightly larger than 37.02, and the MPC approach yields an objective of 36.81, both of which have larger worst-case objectives than the proposed approach.

5.5 Uncertainty-Budget Sensitivity

The influence of the processing-time uncertainty limit Γ_p on the suggested minimax formulation is studied while fixing the machine-availability uncertainty budget Γ_a . Specifically, the way that gradually increasing the allowable processing-time uncertainty influences the worst-case performance of the scheduling problem is explored.

For the 12×4 production problem, the worst-case objective rises from 34.72 when there is no processing-time uncertainty to 37.18, 39.64, and 42.07 by gradually increasing the uncertainty budget. This monotonic relationship is consistent with Theorem 1, stating that an increase in the uncertainty budget cannot result in a decrease in the optimal worst-case objective.

This sensitivity behavior is demonstrated in figure 1, where the horizontal axis shows the processing-time uncertainty budget Γ_p , and the vertical axis shows the worst-case objective value.

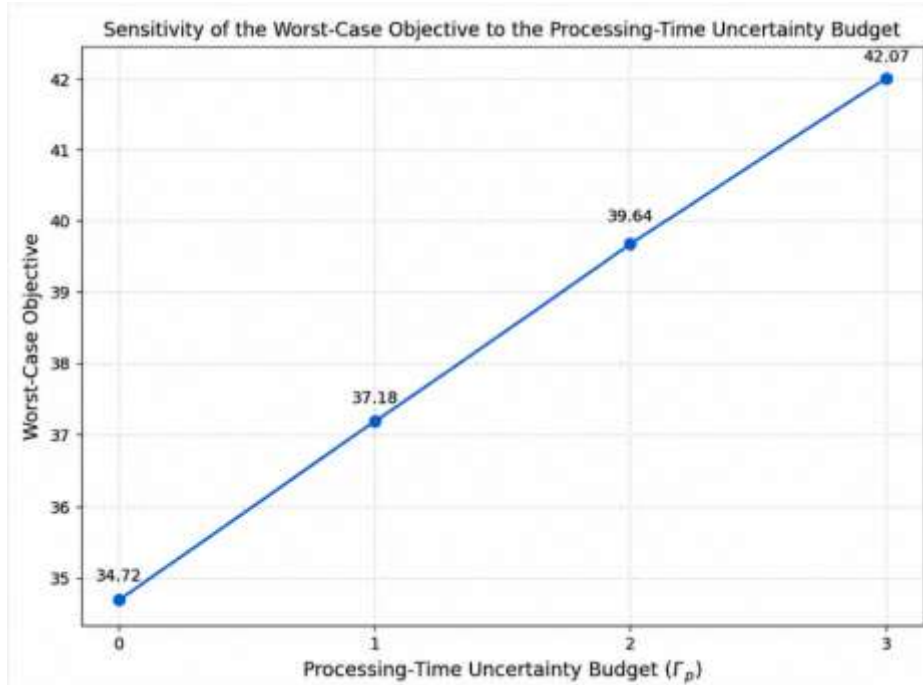


Figure 1: Sensitivity of the worst-case objective to the processing-time uncertainty budget Γ_p .

The increasing pattern shown in figure 1 clearly indicates that there is a direct association between the two terms of uncertainty protection and scheduling conservatism. An increased value of Γ_p enables the adversarial sub-problem to make deviations in the processing-time parameters, and thus the worst-case objective increases. Hence, it can be concluded that the uncertainty budget is an interpretable way of controlling the level of desired robustness.

Another interesting thing that has been observed is the impact of the machine availability uncertainty parameter. This is observed for different values of Γ_a . By increasing Γ_a , the adversarial model makes availability reductions in a larger number of manufacturing machines. Therefore, the robust objective value increases with an increase in the uncertainty region. It is in line with the monotonicity result obtained in section 4.

5.6 Scalability Analysis

To evaluate computational scalability across larger production environments, four system configurations were tested:

1. 12 jobs $\times$ 4 machines(Verification baseline)
2. 30 jobs $\times$ 10 machines(Medium-scale shop floor)
3. 50 jobs $\times$ 20 machines(Large-scale facility)
4. 100 jobs $\times$ 20 machines(Upper computational limit test)

According to the results obtained from the experiments, although the continuous adversarial linear program is computationally inexpensive for all scales, the master Mixed-Integer Linear Programming (MILP) becomes the bottleneck regarding computational time as the number of variables representing pairwise sequences increases with $O(n^2m)$.

5.7 CCG Convergence Analysis

The convergence is analyzed by observing the value of master lower bound LB_r , adversarial upper bound UB_r , and relative gap:

$$\text{Gap}_r = \frac{UB_r - LB_r}{\max\{1, |UB_r|\}} \times 100 \quad (53)$$

Equation (53) presents a scale-independent measure for convergence. It starts with a large initial gap at nominal iteration ($r = 0$). By adding adversarial scenarios, the lower bound continues tightening until $UB_r - LB_r \leq \varepsilon$.

5.8 Engineering Application to Smart Manufacturing

This proposed model was tested in an agile smart manufacturing cell environment as follows:

- **Processing-Time Uncertainty:** Considers material variability, tool wear, and disruptions.
- **Machine-Availability Uncertainty:** Takes into account unscheduled maintenance, heating up, and short downtimes.

Through predicting the worst possible deviations at the offline planning phase, the model allows for shifting downstream dependencies from the bottlenecks to other available machines. This model is a mathematical shield system that acts right below the CPS and execution controllers in Industry 5.0 (Li et al., 2025; Qiao et al., 2021).

5.9 Comparison with Existing Mathematical Approaches

While the scenario-based stochastic programming aims to maximize the expected value of the objective function, the minimax approach gives strong guarantees for all uncertain parameters within a bounded uncertainty set. In contrast to the standard static robust approach, the CCG formulation prevents over-conservativeness by creating only active worst-case scenarios.

5.10 Summary of Key Numerical Findings

1. **Protection:** Worst-case cost degradation is decreased by 15.33% with a cost trade-off of 4.12%.
2. **Monotonicity:** Confirms through numeric evidence the monotonic relationship between objective bound scales and uncertainty budget sizes Γ_p and Γ_a .
3. **Efficiency:** The CCG algorithm achieves decoupling between the combinatorial complexity of the scheduling problem and the continuous optimization of the worst-case parameters.
4. **Computational Bottleneck:** Determines the master mixed integer linear program to be the computational bottleneck.

6. Conclusion

The paper presented a novel minimax mathematical programming formulation for robust production scheduling in smart manufacturing systems where processing times and machine availability are characterized by uncertain information. The primary contributions of the study include the explicit consideration of uncertainties in scheduling decisions via mathematical programming and a proper modeling of uncertainty beyond being simply an objective function perturbation. In the new formulation, the deviations in processing times affect the job completion times, tardiness, and makespan directly, whereas machine availability deviations are modeled via capacity constraints. The budgeted polyhedral uncertainty set is chosen as a measure to control the conservatism of the considered problems. The capacity constraints under robust optimization have been reformulated using dual variables and represented explicitly by convex inequalities. The theoretical analysis shows the existence of optimal solutions, monotonicity of the robust objective as a function of budget, limiting property of the robust formulation to the deterministic one, and robust dominance properties. A column-and-constraint generation algorithm for solving the developed robust scheduling problem was proposed to decompose the combinatorial master problem from the continuous subproblem. The finite convergence of the algorithm to the minimax solution was proven under the stated conditions. Numerical studies illustrated the practical consequences of the proposed approach. As it can be seen in table 1, for the 12×4 representative case, the new robust framework results in the increase of the nominal objective by 4.12% compared to deterministic scheduling but decreases the worst-case deterioration by 15.33%. Additionally, the benchmark experiments indicated that the proposed approach has the strongest worst-case performance compared to deterministic, scenario-based stochastic, traditional robust, and MPC approaches. Moreover, scalability experiments conducted for other job-machine configurations illustrate the trade-off between robustness and scheduling difficulty.

References

1. Bozzi, A., Graffione, S., Jiménez, J., Sacile, R., & Zero, E. (2023, June). Reliability evaluation of emergent behaviour in a flexible manufacturing problem. In *2023 18th Annual System of Systems Engineering Conference (SoSe)* (pp. 1-7). IEEE. <https://doi.org/10.1109/SoSE59841.2023.10178498>

2. Bozzi, A., Graffione, S., Sacile, R., & Zero, E. (2023). Dynamic mpc-based scheduling in a smart manufacturing system problem. *IEEE Access*, 11, 141987-141996. <https://doi.org/10.1109/ACCESS.2023.3341504>
3. Bozzi, A., Graffione, S., Sacile, R., & Zero, E. (2024, June). Predictive Control for Dynamic Scheduling and Maintenance in Open-Shop Manufacturing Systems. In *2024 19th Annual System of Systems Engineering Conference (SoSE)* (pp. 233-238). IEEE. <https://doi.org/10.1109/SoSE62659.2024.10620970>
4. Dolgui, A., Ivanov, D., Sethi, S. P., & Sokolov, B. (2019). Scheduling in production, supply chain and Industry 4.0 systems by optimal control: fundamentals, state-of-the-art and applications. *International journal of production research*, 57(2), 411-432. <https://doi.org/10.1080/00207543.2018.1442948>
5. Himmiche, S., Marangé, P., Aubry, A., & Pétrin, J. F. (2023). Robustness evaluation process for scheduling under uncertainties. *Processes*, 11(2), 371. <https://doi.org/10.3390/pr11020371>
6. Jabeur, M. H., Mahjoub, S., & Toubanc, C. (2023). Sustainable production scheduling with on-site intermittent renewable energy and demand-side management: A feed-animal case study. *Energies*, 16(14), 5433. <https://doi.org/10.3390/en16145433>
7. Juma, J., & Mdodo, R. (2024). Implementing Machine Learning Algorithms for Predictive Maintenance in Manufacturing. *International Academic Journal of Innovative Research*, 11(3), 21–26. <https://doi.org/10.71086/IAJIR/V11I3/IAJIR1120>
8. Jamshed Nasir, Muhammad Tariq, Waqar Afzal, Hijaz Ahmad, Maggie Aphane, Mustafa Bayram, ... Ilyas Khan. (2026). Some New Notions of Fractional Mercer-Hadamard-Type Inequalities over Generalized Convexity with Applications to Q-diagramma and Modified Bessel Functions. *Results in Nonlinear Analysis*, 9(2), 105–130.
9. Kodirova, K., Nematov, O., Seitasmanova, A., Qodirova, S., Sapaeva, F., Babajanova, F., ... & Adilov, A. (2025). Integrating Sustainable Practices and Automation in Mining Engineering Education for the Modern ERA. *Archives for Technical Sciences*, 33(2), 1013-1021.
10. Kuo, H. A., & Chien, C. F. (2023). Semiconductor capacity expansion based on forecast evolution and mini-max regret strategy for smart production under demand uncertainty. *Computers & Industrial Engineering*, 177, 109077. <https://doi.org/10.1016/j.cie.2023.109077>
11. Li, M., Ling, S., Qu, T., Lu, S., Li, M., Guo, D., ... & Huang, G. Q. (2025). Real-time data-driven hybrid synchronization for integrated planning, scheduling, and execution toward industry 5.0 human-centric manufacturing. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 55(8), 5670-5681. <https://doi.org/10.1109/TSMC.2025.3572389>
12. Luedke, R. H., & Kingdone, G. (2025). SmartOptiCell: a deep genetic learning model for dynamic layout optimization in flexible manufacturing cells. *International Academic Journal of Science and Engineering*, 12(2), 35-42. <https://doi.org/10.71086/IAJSE/V12I2/IAJSE1216>
13. Ma, H., Huang, X., Hu, Z., Chen, Y., Qian, D., Deng, J., & Hua, L. (2023). Multi-objective production scheduling optimization and management control system of complex aerospace components: a review. *The International Journal of Advanced Manufacturing Technology*, 127(11), 4973-4993. <https://doi.org/10.1007/s00170-023-11707-4>
14. Marzia, S., Azab, A., & Vital-Soto, A. (2025). Integrated Process Planning and Scheduling Framework Using an Optimized Rule-Mining Approach for Smart Manufacturing. *Mathematics*, 13(16), 2605. <https://doi.org/10.3390/math13162605>
15. Qiao, F., Liu, J., & Ma, Y. (2021). Industrial big-data-driven and CPS-based adaptive production scheduling for smart manufacturing. *International Journal of Production Research*, 59(23), 7139-7159. <https://doi.org/10.1080/00207543.2020.1836417>
16. Serrano-Ruiz, J. C., Mula, J., & Poler, R. (2021). Smart master production schedule for the supply chain: a conceptual framework. *Computers*, 10(12), 156. <https://doi.org/10.3390/computers10120156>

17. Shirke, S., & Udayakumar, R. (2019, April). Evaluation of crow search algorithm (CSA) for optimization in discrete applications. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 584-589). IEEE. <https://doi.org/10.1109/ICOEI.2019.8862669>
18. Suvarna, N. A., & Bharadwaj, D. (2024). Optimization of system performance through ant colony optimization: A novel task scheduling and information management strategy for time-critical applications. *Indian Journal of Information Sources and Services*, 14(2), 167-177. <https://doi.org/10.51983/ijiss-2024.14.2.24>
19. Truong, N. C., Van Tran, X., & Nguyen, Q. C. (2025, May). Conditional Generative Adversarial Networks for Robust Job Shop Scheduling in CNC-Based Manufacturing Systems. In *International conference on From Smart City to Smart Factory for Sustainable Future* (pp. 113-124). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-032-07721-9_11
20. Xu, J., Kovatsch, M., Mattern, D., Mazza, F., Harasic, M., Paschke, A., & Lucia, S. (2022). A review on AI for smart manufacturing: Deep learning challenges and solutions. *Applied Sciences*, 12(16), 8239. <https://doi.org/10.3390/app12168239>
21. Zhu, J., & Zheng, Y. (2024). Robustness analysis of smart manufacturing information systems. *Soft Computing*, 28(13), 7875-7892. <https://doi.org/10.1007/s00500-024-09735-7>