# Evolutionary Algorithm-Based Real-Time Scheduling via Simulation-Optimization for Multiproduct Batch Plants

Scheduling in the process industry determines the sequence and timing of operations to optimize objectives such as minimizing order tardiness and improving plant performance. Traditionally, scheduling is performed offline and infrequently, with production data manually transferred to the scheduling system. This delayed approach can lead to discrepancies between the plant’s actual state and the scheduling model. For example, disturbances during this time may render the solution ineffective due to mismatches between the plant and its model. Other problems may arise in environments where frequent disturbances and fast responses are necessary, as the traditional scheduling approach often lacks the flexibility to adapt in real time. Because of this inertia, traditional methods often fail to respond quickly enough, making them unsuitable for dynamic production environments.

To address this problem, real-time scheduling establishes a continuous exchange of information between the scheduling system and the production plant's control system. This ensures the model is regularly updated to reflect the current production state, generate accurate predictions, and facilitate fast decision-making. For effective real-time integration, the scheduling system must accurately model the plant's behavior and quickly generate solutions to prevent disturbances from propagating, which would otherwise lead to further discrepancies.

We present a simulation-optimization approach tailored for real-time requirements, combining an evolutionary algorithm with a discrete-event simulator. The simulation model is continuously updated with the latest plant data, enabling the optimizer to make well-informed decisions in real time.

We validate our approach using a multiproduct, multistage batch plant in the pharmaceutical industry, demonstrating that it can generate high-quality solutions quickly through continuous synchronization and rapid disturbance responses. We further test the system by simulating disturbances, showing that with appropriate parameter tuning and parallelization, our approach achieves near-optimal solution quality. We compare our results with those of an idealized scheduler, demonstrating that our method performs competitively under time constraints and effectively manages unforeseen disturbances.

**Keywords:** Modelling and Simulation, Large Scale Design and Planning/Scheduling

# Introduction

Production scheduling is a decision-making process that determines the sequence and allocation of order operations and resources in production processes, improves plant efficiency, and reduces operational costs. Scheduling is typically performed offline and generates static production plans in advance which operators of the production plant execute on the shop floor.

Real-world production environments are dynamic with frequent unforeseen events such as machine breakdowns, new order arrivals, or uncertain processing times and operation costs. Such events render offline schedules impractical or infeasible without further revisions or full reschedules.

Online scheduling systems generate production plans in real-time and maintain a continuous information exchange between the scheduling system and the control system of the production plant. The requirements are different from offline scheduling systems with high expectations on responsiveness to unforeseen events, modifiability to adapt to evolving constraints and objectives, and accuracy in its predictions, to continual align with the plant state and minimize the impact of disruptions. Many existing approaches fall short to deliver timely responses and to generate schedules that are interpretable and executable by plant operators (Henning et al. 2023).

*Online and offline scheduling complement each other in which offline scheduling produce plans for a period of time and online scheduling dynamically adapts the plan if disruptive events occur.*

Simulation-optimization is well suited to fill this gap by using high-fidelity simulation models to represent the production process in detail and generates production plans, the plant operators can follow without much modification. Simulation models can automatically adapt to changing plant states and production plans, e.g., which orders to produce first, and are simple to modify when the plant structure and parameters of the process change, e.g., resource availability and operation uncertainties, and thus fit the requirements for online scheduling of dynamic production processes.

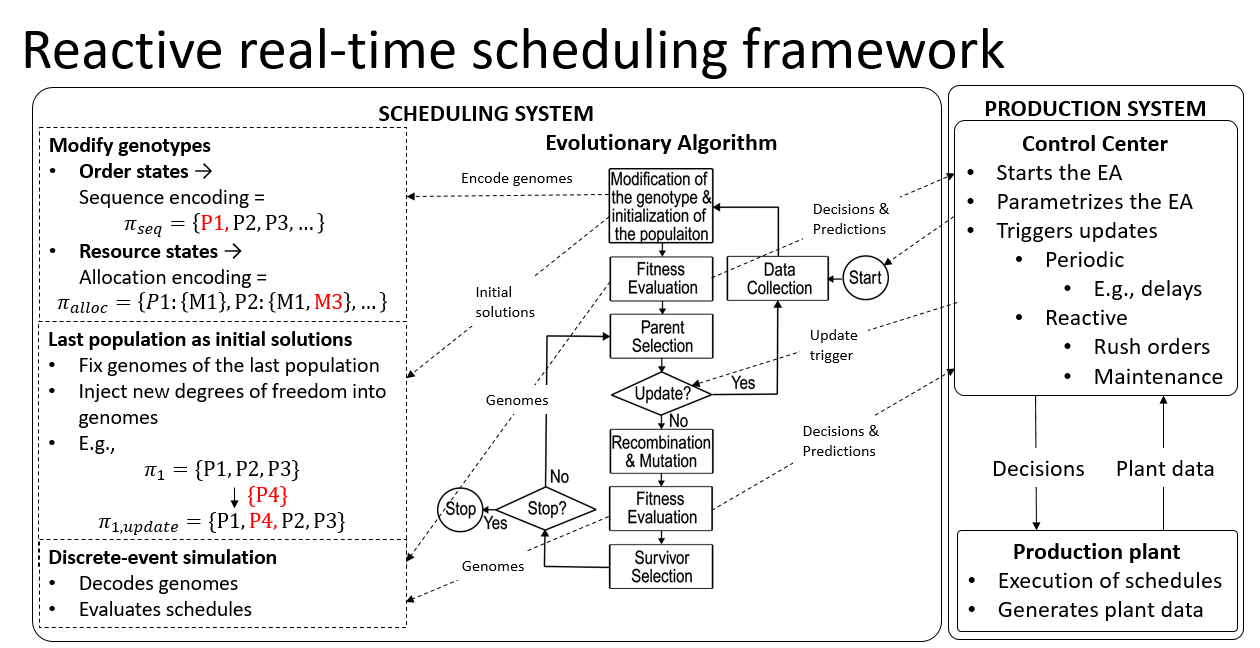
Simulation-optimization methods that combine complex models with metaheuristic methods are inherently inefficient for optimization tasks. This inefficiency arises primarily from the computational demands of simulation models, where processing time increases proportionally with model complexity. Additionally, simulations often involve a significant initial overhead for execution. Metaheuristics, as guided search algorithms, further contribute to these limitations. Their iterative nature makes them less suitable for scenarios requiring real-time decision-making capabilities.

The study uses a modified rolling horizon evolutionary algorithms (RHEA), a type of evolutionary algorithm designed for real-time decision-making for general video game playing (GVGA), where dynamic changes in the environment and fast responses are crucial. RHEA evolves sequences of decisions at each decision point and continuously optimizes solutions based on real-time feedback. We modified RHEA for online production scheduling, e.g., model complexity, robustness, and adaptability to disturbances and uncertainties.

We answer the question of how to bring simulation-optimization to a real plant where schedules might become infeasible or suboptimal due to disruptions. We show that simulation-optimization with RHEA is suited for reactive real-time scheduling systems by continuously optimizing the schedules and updating the model parameters and optimization instances with the plant state.

# Approach

Our online scheduling framework uses simulation-optimization with real-time feedback from the production process to refine decisions and update the simulation model to align with the state of the production. Fig. X illustrates the components and interactions of our online scheduling framework.



The framework is designed to operate in real time, ensuring that scheduling decisions remain both feasible and efficient as production conditions evolve. It consists of two tightly interconnected components: the Scheduling System and the Production System.

## Scheduling system

**The scheduling system** leverages an evolutionary algorithm to generate and refine scheduling solutions, represented as genotypes. These genotypes encode both the order of tasks (πseq={P1,P2,P3,...}πseq​={P1,P2,P3,...}) and the allocation of resources to tasks (πalloc={P1:{M1},P2:{M1,M3},...}πalloc​={P1:{M1},P2:{M1,M3},...}). By encoding decisions in this way, the framework can systematically explore and optimize schedules through evolutionary operators, including selection, recombination, and mutation. The initial population of solutions is derived from the last generation of schedules, ensuring continuity and leveraging historical data. When necessary, new degrees of freedom are injected, such as adding new tasks or adjusting allocations, to address changing production requirements. To evaluate the quality of the schedules, the system employs a discrete-event simulation that decodes genotypes and simulates their execution under realistic production conditions.

The evolutionary algorithm operates iteratively, starting with the modification and initialization of genotypes. At each iteration, the algorithm evaluates the fitness of each solution, selects the most promising candidates, and applies recombination and mutation to generate new solutions. A survivor selection process ensures that only the best-performing solutions are carried forward. This iterative process continues until a stopping criterion is met. However, what distinguishes this framework is its ability to adapt in real time: periodic and reactive triggers from the production system prompt updates to the algorithm, ensuring that the schedules remain aligned with the current state of the production plant.

## Production system

The production system plays a critical role in providing real-time feedback to the scheduling system. It comprises a control center and the production plant itself. The control center initiates and parametrizes the evolutionary algorithm while monitoring real-time data from the production plant. Updates to the scheduling process are triggered either periodically—such as when delays occur—or reactively in response to unforeseen events, such as rush orders or maintenance requirements. The production plant executes the schedules generated by the algorithm and provides detailed operational data, including task progress and resource availability, which is continuously fed back into the control center. This feedback loop ensures that the scheduling system remains responsive and accurate, adapting to both predictable and unexpected changes in the production environment.

By integrating these components, the reactive scheduling framework achieves a balance between optimization and adaptability. The evolutionary algorithm ensures that the schedules are efficient, while the real-time feedback mechanism guarantees that the decisions remain feasible and responsive to the current state of the production system. This synergy enables the framework to address dynamic and complex scheduling problems in real-world manufacturing environments effectively.

# Case-Study

The case study addressed in this work is taken from Kopanos, et al., (2010). It is a multiproduct batch plant with 17 units (machines) that are organized in 6 stages. The problem, which is a variant of a hybrid flow shop problem, comprises 12 instances that vary in the number of orders, the objectives, and in the storage policy. The features of the problem include limited product-unit flexibility, machine-dependent processing times, sequence-dependent changeover times, and product-specific recipes, meaning that certain jobs are not processed on some of the available stages. This paper considers three problem instances with 30 orders, unlimited intermediate storage (UIS), and the objectives Makespan (Cmax), Overall & Changeover Cost (O.&C.C.) and Weighted Lateness (W.L.), which are minimized. The objectives are defined as 𝐶𝐶𝑚𝑚𝑚𝑚𝑚𝑚 = max􀵫𝐶𝐶1, . . , 𝐶𝐶|𝐼𝐼|􀵯 [ℎ] (1) 𝑂𝑂.&𝐶𝐶. 𝐶𝐶 = 𝜔𝜔𝜔𝜔𝑚𝑚𝑚𝑚𝑚𝑚 + Σ𝑖𝑖∈𝐼𝐼 𝑐𝑐𝑐𝑐𝑖𝑖 [ℎ], with 𝜔𝜔 = 0.9 ⋅ 103 (2) 𝑊𝑊. 𝐿𝐿.= 𝛴𝛴𝑖𝑖∈𝐼𝐼 𝛼𝛼𝐸𝐸𝑖𝑖 + 𝛽𝛽𝑇𝑇𝑖𝑖 [ℎ],with 𝛼𝛼 = 0.9 and 𝛽𝛽 4.5 (3) 𝐶𝐶𝑖𝑖 denotes the completion time of job 𝑖𝑖, 𝑐𝑐𝑐𝑐𝑖𝑖 denotes the sum of all changeover times multiplied with a sequence dependent impact factor associated with job 𝑖𝑖, and 𝐸𝐸𝑖𝑖 and 𝑇𝑇𝑖𝑖 denote the earliness and the tardiness of job 𝑖𝑖.

# Experiment

* 30 & 60 & 90 orders?
* Initilialize with a set of good solutions
* Disturbances

# Results