International Refereed Journal of Engineering and Science (IRJES)
ISSN (Online) 2319-183X, (Print) 2319-1821
Volume 15, Issue 4 (Jul. – Aug. 2026), PP. 93-106

Simulation-Driven Anodizing Line Optimization: Load Sequencing and Hoist Dispatching

Taha Mankır* Sanem Aydar**
* *(B.Sc. Student, Department of Industrial Engineering, Çukurova University, Adana, Türkiye; Engineering Intern, Zahit Alüminyum San. ve Tic. A.Ş., Adana, Türkiye)*
E-mail:taha.mankir123@gmail.com, ORCID: 0009-0004-9943-4585
** *(R&D Engineer, Department of* R&D, Zahit Alüminyum San. ve Tic. A.Ş., *Adana, Türkiye)*

Abstract: *This study develops a simulation-driven decision-support framework to reduce post-nominal residence time, termed net overstay, in a Type II anodizing line constrained by tank occupancy, hoist access, and shared-rail interactions. A discrete-event model of parallel tanks and four order-preserving hoists combines the S7 operational-risk dispatching rule with Simulated Annealing for daily load-entry sequencing. Five production days comprising 549 loads were evaluated under four random seeds and two dispatching rules, resulting in 40 optimization runs and 40,000 candidate evaluations. On the baseline sequences, S7 reduced total net overstay by 12.9% but increased makespan by 2.8%. Simulated Annealing improved 12 of 20 Current runs and 15 of 20 S7 runs; mean total net overstay under S7 decreased by 17.6%. All loads were completed in the 27 improved runs, with lower makespan and no increase in critical-process net overstay. The results show that coordinated load sequencing and explainable hoist dispatching can improve operational performance without additional physical capacity.*

Keywords: *Type II anodizing; tank blockage; discrete-event simulation; simulated annealing; hoist scheduling.*

Date of Submission: 25-07-2026 | Date of acceptance: 07-08-2026

---

1. INTRODUCTION

1.1. The Anodizing Process and the Focus of This Study

Anodizing is an electrochemical surface-treatment process that improves the corrosion resistance, wear resistance, and decorative appearance of aluminum; sulfuric acid-based processing is commonly classified as Type II anodizing [1]. The line under study includes degreasing, pretreatment, neutralization, anodizing, optional coloring, sealing, rinsing, and drying, with six route families and coating-thickness classes determining the required tank sequence and occupation times. Because a tank is released only when a hoist physically removes the load, downstream occupancy, hoist availability, and shared-rail constraints may extend residence beyond the nominal process duration, defined in this study as net overstay. This research evaluates whether coordinating daily load-entry sequencing with dynamic hoist-dispatching priorities can reduce net overstay in critical tanks while completing all scheduled loads and controlling makespan through a discrete-event simulation, the S7 rule, and Simulated Annealing.

1.2. Time Overruns and Operational Risk

Process-specific residence-time limits are important because the effects of overstay vary across chemical baths. Extended exposure can influence etching, anodic-film integrity, coloring, and sealing behavior [2,3]. This study therefore treats post-nominal residence time as a process-sensitive operational-risk indicator. The available records support timing-based risk analysis, while synchronized quality data can later be used to calibrate direct defect models.

1.3. Literature Review and Research Gap

Hoist Scheduling Problem research has largely focused on deterministic and cyclic production, later extending to parallel tanks, multiple hoists, collision constraints, and joint sequence–hoist optimization [11–14]. Although simulation-based studies have addressed anodizing lines [4], existing models generally provide limited representation of noncyclic daily product mixes, endogenous tank blocking, shared-rail interference, and process-sensitive post-nominal waiting. Quality-oriented studies have also linked immersion time to quality through fuzzy satisfaction functions [15], but such functions require synchronized quality observations. To address this gap, the present study develops a field-informed discrete-event simulation of a Type II anodizing line with four order-preserving hoists, overlapping zones, parallel tanks, handover points, and real daily route–coating-thickness mixes. The framework combines the explainable S7 operational-risk dispatching rule with simulation-based Simulated Annealing for load-entry sequencing and evaluates both decision layers through matched production days, random seeds, and performance indicators. Its purpose is to identify high-performing feasible solutions and reduce net overstay and blocking under fixed physical capacity; defect prediction, closed-loop control, and global-optimality assessment remain outside the present scope.

II. THE PRODUCTION SYSTEM UNDER STUDY

2.1. Line, Processes, and Physical Resources

The system boundary extends from the entry conveyor to the exit conveyor and includes degreasing, surface preparation, neutralization, anodizing, coloring, sealing, hot-water treatment, draining, and drying. The physical tanks are arranged along a linear production line. Parallel resources, such as six anodizing tanks, three sealing tanks, and three hot-water tanks, provide nominal capacity, and each physical tank accommodates one load with an individually represented location and occupancy state. Draining is represented as a timed operation during which Hoist 4 remains loaded and holds the load at an incline for a specified duration.

Table 2.1. Process groups and physical resources of the line under study

Process area	Physical resources	Primary function
Entry	Entry conveyor and two buffers	Admission of loads to the line
Degreasing	Two parallel tanks	Removal of oil and manufacturing residues
Pretreatment	AcidMat, stripping, caustic, and rinse tanks	Route-dependent surface preparation
Neutralization	Two parallel tanks	Preparation of the surface for anodizing
Anodizing	Six parallel tanks and rinse tanks	Formation of the anodic oxide layer
Coloring	Two tin tanks and one nickel tank	Route- and color-dependent coloring
Sealing	DI tank, three sealing tanks, and rinse tanks	Closure of the pore structure
Final processes	Three hot-water tanks, draining, and two ovens	Water removal and drying
Exit	Exit conveyor	Removal of completed loads from the line

2.2. Hoists, Handover Points, and Routes

Four overhead hoists operate on the same rail, maintain their physical order, and serve overlapping responsibility zones. Load handovers take place through shared tanks: one hoist places the load in a shared tank, and the next hoist subsequently retrieves it. The neutralization, coloring, and hot-water areas therefore act simultaneously as processing resources, hoist-responsibility boundaries, and synchronization points. Occupancy of a handover tank may consequently affect two hoist zones at the same time.

Table 2.2. Hoist responsibility zones and handover points

Hoist	Speed (m/s)	Primary responsibility	Handover
Hoist 1	0.450	Entry, degreasing, and pretreatment	To Hoist 2 at neutralization
Hoist 2	0.455	Neutralization and anodizing	To Hoist 3 in the coloring area

Hoist	Speed (m/s)	Primary responsibility	Handover
Hoist 3	0.499	Coloring, DI, and sealing	To Hoist 4 in the hot-water area
Hoist 4	0.455	Hot water, draining, ovens, and exit	To the exit conveyor

Table 2.3. Product-route families represented in the model

Route	Distinctive pretreatment	Color	Primary characteristic
Bright	Direct neutralization	None	Predominantly 7 µm
Natural	Caustic treatment and rinsing	None	Natural appearance
AcidMat	AcidMat, stripping, and rinsing	None	Acid-matte finish
SAKEM	Caustic treatment and rinsing	Tin	Special color route
MAN	Caustic treatment and rinsing	Tin	Color-dependent duration
PAN	Direct neutralization	Nickel	Nickel coloring

Route and coating-thickness mix determines the regional workload of the line. Route families concentrate demand on different hoists and coloring areas, while coating thickness increases anodizing and sealing occupancy. Consequently, the bottleneck is dynamic and depends on the daily route–coating-thickness sequence rather than on a permanently fixed station.

III. SOLUTION FRAMEWORK AND STRATEGY

3.1. Two-Level Decision Structure The framework keeps physical capacity fixed and integrates two decision layers. At the planning level, Simulated Annealing generates alternative daily load-entry sequences through two-load swaps, while at the operational level, S7 ranks physically feasible hoist tasks using operational-risk-weighted net overstay, downstream blockage, and task age. Each candidate sequence is evaluated by discrete-event simulation under tank-occupancy, hoist-movement, handover, and shared-rail constraints using completed loads, makespan, critical and total net overstay, P90 cycle time, and average WIP. Figure 4.1 summarizes the interaction between the planning layer, operational layer, simulation model, and performance evaluation.

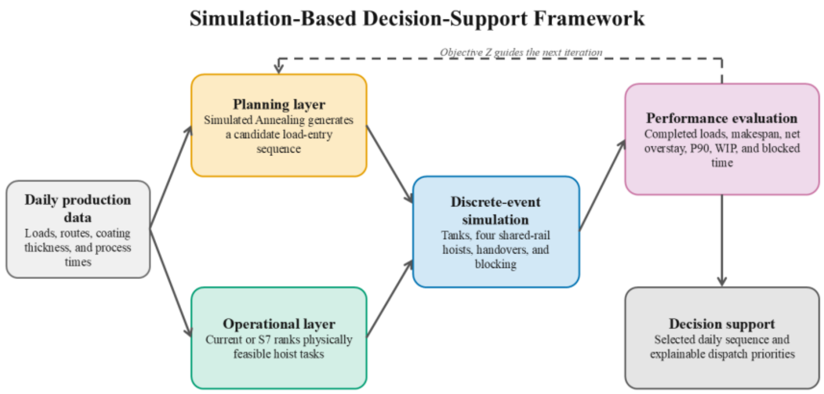

Figure 3.1. Two-level simulation-based decision-support framework

The framework provides explainable decision support for planning and execution. Before production begins, planners may receive one or more candidate load sequences, while operators may receive a reasoned task recommendation at each dispatching decision. Hardware safety interlocks, mandatory recipe requirements, and operator authority retain precedence over algorithmic recommendations. Persistent excess demand, equipment events, and additional process constraints can be addressed through complementary capacity planning, maintenance response, and model extensions.

IV. DATA PREPARATION AND USE

4.1. Data Preparation, Simulation Inputs, and Scope

The dataset combined operational records and field observations covering process durations, temperatures, hoist movements, and load arrivals. Column names, units, route labels, coating-thickness classes, and time formats were standardized, with routes consolidated into six product families and coating thickness grouped into 7, 13, and 25 µm classes. After separating structural blanks, missing values, and event-related notes, group-specific 95th-percentile thresholds were used to distinguish intrinsic process durations from historical blockage. Candidate probability distributions were evaluated using AICc, BIC, bootstrap goodness-of-fit tests, and train–test performance, and simulated durations were bounded by the observed filtered ranges [6–8]. Five production days representing different route and coating-thickness mixes were selected, comprising 549 loads modeled as independent scheduling problems. For paired Current and S7 comparisons, daily process durations were held constant, while Simulated Annealing modified only the load-entry sequence. Automated checks and run metadata preserved row-level traceability, random seeds, dispatching rules, and iteration counts. The available data support interpreting net overstay as an operational-risk indicator, although broader synchronized quality and cost data would be required to estimate defect and economic outcomes directly.

V. DISCRETE-EVENT SIMULATION METHOD

5.1. Method Selection and Model Purpose

The line was modeled using discrete-event simulation because load progression depends on tank occupancy, hoist availability and position, shared-rail interference, and handover status. The model was implemented in Python with SimPy's process-interaction approach [9], using one second as the base time unit and advancing the clock between scheduled events. It evaluates tank utilization, blocking, residence times, and production performance under alternative load sequences and hoist-dispatching rules, while matched quality data can support future product-quality extensions.

5.2. System Boundary, Entities, and Resources

The simulation boundary begins with load generation at the entry conveyor and ends when the load is placed on the exit conveyor after completing the oven operations. Each load stores its identifier, route family, coating thickness, arrival time, next process step, and current station. At the system level, the model tracks tank occupancies, the positions and availability states of the four hoists, pending transport tasks, active loads, and recorded station visits.

Loads follow their specified routes in the forward direction and may be assigned to eligible parallel tanks. Each physical tank accommodates one load; consequently, the six anodizing tanks and three sealing tanks are represented and occupied separately. The entry and exit conveyors are modeled as virtual stations. Draining is represented as a timed activity during which Hoist 4 holds the load in an elevated and inclined position and dedicates its capacity to this operation.

Table 5.1. Main entities and resources represented in the simulation model

Component	Representation	Main attributes	Function
Load	Job object and route-step sequence	Route, coating thickness, current station, next step, and process times	Represents the production order and its process flow
Physical tank	Capacity-one resource	Occupancy status, current load, nominal completion time, and reservation status	Prevents simultaneous assignment of more than one load
Hoist	Continuous autonomous process	Position, empty and loaded speeds, availability state, and assigned task	Transfers loads between stations
Dispatcher	Shared task queue and selection mechanism	Pending, feasible, claimed, and requeued tasks	Selects an eligible task for an idle hoist

Component	Representation	Main attributes	Function
Rail controller	Position- and path-control mechanism	Hoist order, zone boundaries, active movements, and neighboring hoists	Prevents passing and conflicting rail movements
Event and station-visit log	Time-stamped records	Entry time, nominal duration, exit time, source, destination, and waiting reason	Supports KPI calculation and model verification

5.3. Load Generation and Process Progression
The model supports stochastic generation and CSV replay; in the main experiment, loads followed the order specified in the daily CSV files and waited whenever the entry position was occupied. Each load entered with validated route and coating-thickness data, and only process steps with positive duration values were included. Processing began when the load was placed in its assigned tank; after nominal completion, a transport task was generated, while the load continued to occupy the source tank until physical pickup by a hoist. Post-nominal waiting therefore emerged endogenously from destination occupancy, hoist position and availability, and shared-rail constraints.

5.4. Transport-Task Generation and Tank Reservation
Each transport task contains the source station, the next process step, its creation time, the primary hoist, and any additional hoists eligible to perform the task. The dispatcher considers only tasks that are within the relevant hoist's operating range, whose source remains occupied, for which at least one destination is available, and whose movement is feasible under the rail constraints.
When multiple parallel destinations are available, the nearest unoccupied physical tank to the source is selected and reserved before pickup. When all eligible destinations are occupied, the task remains in the queue and source occupancy continues until a destination becomes available.

Once a task has been assigned, the hoist travels empty to the source, picks up the load, travels loaded to the destination, and drops off the load. The process at the destination begins after drop-off. The travel time m is calculated using Equation (1), where x_h denotes the destination position, x_s denotes the source position, and v is the applicable loaded or empty hoist speed.

$$m = \frac{|x_h - x_s|}{v} \quad (1)$$

Pickup and drop-off durations are sampled within the intervals of 20–30 seconds and 10–15 seconds, respectively. These durations represent physical handling operations and are modeled separately from chemical processing times.
A fixed random seed supports the reproducibility of repeated runs. Changing the load sequence also changes the order of simulation events and random-number calls. Each candidate sequence therefore produces a reproducible, sequence-specific assignment of random values to loads.

5.5. Hoist Zones and Rail-Movement Control
Each hoist operates primarily within its assigned responsibility zone. The overlapping sections of these zones enable handovers at the neutralization, coloring, and hot-water areas. Some rinse-related transport tasks may also be performed by an eligible neighboring hoist. Under the Current rule, the model retains a preference for the primary hoist while allowing flexible task assignment where physically feasible.

The rail controller continuously monitors the physical order of Hoists 1–4. Before a movement begins, the requested destination is checked against the hoist's operating zone and the current or projected positions of neighboring hoists. A safety clearance of 0.05 m is maintained between adjacent hoists.

If the requested path overlaps with an active neighboring movement, the requesting hoist waits until a position-change event makes the path feasible. This logic prevents hoists from passing one another or occupying conflicting sections of the shared rail. The start and end times of rail-related waiting events are recorded for performance analysis.

5.6. Current and S7 Dispatching Rules
Under the Current rule, feasible tasks are sorted in ascending order according to effective priority, entry or urgency subpriority, primary-hoist ownership, distance, ready time, and creation order. A penalty of 1,000 is applied to

tasks assigned to a flexible rather than primary hoist, thereby preserving the preference for primary-hoist ownership. The Current rule extends beyond first-in, first-out by incorporating ownership, distance, and timing criteria, while S7 adds the composite operational-risk score.

Under the S7 rule, the same physical-feasibility checks are applied, but feasible tasks are assigned a high-score–high-priority value composed of operational-risk, blockage, and age components. Net overstay o_i is defined in Equation (2), the process-sensitivity-based risk component r_i in Equation (3), and the task-age component a_i in Equation (4).

$$o_i = \max(0, t - f_i - 45) \tag{2}$$
$$r_i = w_i o_i \tag{3}$$
$$a_i = 0.20 u_i \tag{4}$$

In Equation (2), t denotes the current simulation time and f_i denotes the nominal completion time. In Equation (3), w_i is an expert-defined operational-risk weight rather than an empirically observed defect probability. In Equation (4), u_i is the time for which the task has remained in the dispatching queue.

The raw downstream-blockage score b_i^r is calculated using Equation (5). It combines indicator variables for an occupied destination I_i^d, an alternative-buffer constraint I_i^b, a waiting upstream task in the same hoist zone I_i^u, and a critical-chain condition I_i^c.

$$b_i^r = 150 I_i^d + 150 I_i^b + 200 I_i^u + 250 I_i^c \tag{5}$$

Equation (6) applies the blockage component b_i only when net overstay is positive and caps it at 800 points.

$$b_i = \min(b_i^r, 800) I(o_i > 0) \tag{6}$$

The final S7 task score p_i is calculated using Equation (7); a higher value indicates a higher dispatching priority.

$$p_i = r_i + b_i + a_i \tag{7}$$

Table 6.2. Comparison of the Current and S7 dispatching rules

Decision component	Current rule	S7 rule
Physical feasibility	Reachability, destination occupancy, and rail constraints	Same feasibility checks; physical constraints retain precedence over the score
Primary ranking basis	Effective priority, primary/flexible ownership, distance, and task time	Operational-risk-weighted net overstay, blockage, and task age
Critical-task protection	Priority class and timeout mechanisms	General threshold of 180 s, hard threshold of 600 s, and process-specific early thresholds
Process-specific treatment	Priority inherited from the general ranking criteria	AcidMat/Caustic: 30 s and +300; sealing: 60 s and +200
Parallel hot-water tanks	Feasibility and proximity	Feasibility and proximity with additional load balancing
Tie-breaking	Distance, ready time, and creation order	Ready time and creation order

The protection mechanism restricts the candidate set when tasks exceeding the specified thresholds remain sufficiently close in score to the highest-ranked task. The general threshold is 180 seconds, and the hard threshold is 600 seconds. Earlier protection is applied after 30 seconds for AcidMat and caustic-treatment tasks and after 60 seconds for sealing tasks.

These thresholds are operational decision parameters that promote timely selection of critical tasks. Field calibration can align them with manufacturer guidance and plant-specific process tolerances before operational deployment.

5.7. Event Flow, Termination Condition, and Performance Indicators

For each load, the principal event sequence consists of arrival, source-station assignment, destination reservation, empty hoist movement, pickup, loaded movement, drop-off, process completion, and generation of the next transport task. After the final process step, the load is transferred to the exit conveyor, and the active-load count is reduced by one.

A simulation run terminates only when the arrival generator has completed and the active-load count has reached zero. This condition ensures that alternative scenarios are compared after the same complete set of loads has finished processing.

For every tank visit, the model records the tank-entry time, nominal processing duration, and pickup time. The actual residence time τ_i, gross overstay o_i^g, and net overstay o_i are calculated using Equations (8), (9), and (2), respectively. Here, n_i denotes the nominal duration of the visit, while t_i^{in} and t_i^{out} denote the tank-entry and pickup times, respectively.

$$\tau_i = t_i^{\text{out}} - t_i^{\text{in}} \tag{8}$$
$$o_i^g = \max(0, \tau_i - n_i) \tag{9}$$

The main performance indicators are the number of completed loads, tank-level net overstay, number of hoist movements, rail-wait events, makespan, individual cycle times, mean cycle time, and average work-in-process. Makespan, cycle time, mean cycle time, and average WIP are calculated using Equations (10)–(13).

$$M = \max_i(C_i) - \min_i(A_i) \tag{10}$$
$$c_i = C_i - A_i \tag{11}$$
$$\bar{c} = \frac{1}{n}\sum_{i=1}^{n} c_i \tag{12}$$
$$\bar{W} = \frac{1}{M}\int_0^M W\,(t)dt = \frac{\sum_{i=1}^{n} c_i}{M} \tag{13}$$

Here, A_i and C_i denote the arrival and completion times of load i, respectively; $W(t)$ denotes the number of loads in the system at time t; and n is the number of completed loads.

5.8. Model Verification, Validation, and Scope

Verification compared the implementation with the documented line layout, six route families, parallel tanks, handover points, hoist zones, draining operation, and shared-rail order constraints. Automated checks enforce valid routes and durations, destination reservation before pickup, source release after pickup, monotonic simulation time, and completion of every scheduled load. Detailed event and station-visit logs provide load-level traceability.

Validation follows comparisons of empirical and simulated residence times, loaded and empty travel times, makespan, and route-specific flow patterns, supported by operator and process-engineer review [10]. The current records support comparative evaluation of sequencing and dispatching alternatives; additional production days, stochastic replications, and independent observations can strengthen deployment-oriented validation.

VI. SIMULATED ANNEALING METHOD

6.1. Definition of the Optimization Problem

For each production day, the characteristics of the assigned loads are fixed, and only their load-entry sequence is treated as a decision variable. For a day comprising n loads, a candidate solution is represented by the permutation defined in Equation (14). Each load j_i retains its identity, route, coating thickness, inter-arrival time, and load-specific processing times as an indivisible record. Across the five experimental days, n ranges from 100 to 122. Because exhaustive enumeration of a solution space of size $n!$ is computationally impractical, Simulated Annealing was employed [5]. In Equation (14), r denotes the candidate sequence, and j_i denotes the load occupying sequence position i.

$$r = (j_1, j_2, \dots, j_n) \tag{14}$$

6.2. Initial Solution and Neighborhood Structure

The initial solution for each day was constructed from the chronological order of the Entry Time field in the empirical anodizing-production records. This sequence was first simulated using the relevant Current or S7 dispatching rule to obtain a run-specific reference objective value.

At each iteration, two distinct positions in the current permutation were selected with equal probability, and the loads at those positions were exchanged to generate a single-swap neighbor. Because all load-specific attributes remain attached to the corresponding load record, the swap operation changes only the sequence in which loads are introduced into the simulated line.

6.3. Evaluation of Candidate Solutions by Simulation
Each candidate permutation was converted in memory into validated load-arrival records while preserving load identity, route, coating thickness, inter-arrival time, and positive-valued T_* durations. A new SimPy environment and anodizing-line model were created for each candidate, and the simulation continued until all loads reached the exit conveyor. Load-specific process durations were held constant within each run, and the same simulation seed was used to support reproducibility. During optimization, only the state information required for KPI calculation was retained to reduce computational overhead, while the best sequence was subsequently re-simulated in detailed-reporting mode.

6.4. Objective Function and Priority Hierarchy
The objective function is a penalty-based composite measure that compares the KPIs of a candidate sequence with those of the selected reference run. Let M denote makespan, N the number of completed loads, O total net overstay, P_{90} the 90th percentile of cycle time, and W average WIP. The subscript 0 denotes the corresponding reference value.
Makespan and completed-load violations are defined in Equations (15) and (16), respectively. The critical-process penalty is defined in Equation (17), and the normalized secondary performance indicators are given in Equations (18)–(20). The critical-process set H_c comprises AcidMat, stripping, caustic treatment, anodizing, tin coloring, nickel coloring, and sealing. In Equation (17), O_h and $O_{h,0}$ denote the candidate and reference total net overstays for critical process group h, respectively.

$$V_M = \max\left(0, \frac{M}{M_0} - 1\right) \tag{15}$$
$$V_T = \max(0, N_0 - N) \tag{16}$$
$$V_C = \sum_{h \in H_c} \max\left(0, \frac{O_h}{O_{h,0}} - 1\right) \tag{17}$$
$$N_O = \frac{O}{O_0} \tag{18}$$
$$N_P = \frac{P_{90}}{P_{90,0}} \tag{19}$$
$$N_W = \frac{W}{W_0} \tag{20}$$

If a reference denominator equals zero, the implementation applies a division-by-zero safeguard. The contribution is set to zero when the corresponding candidate value is also zero; otherwise, the absolute candidate value is used. The composite objective function is defined in Equation (21).

$$\min Z(r) = 1000000V_M + 100000V_T + 10000V_C + 100N_O + 10N_P + N_W \tag{21}$$

Table 6.1. Objective-function components and their associated weights

Component	Weight	Calculation	Methodological purpose
Makespan violation (V^M)	1,000,000	Positive relative increase over the reference	Strongly penalizes deterioration in total completion time
Completed-load violation (V^T)	100,000	Positive shortfall in completed loads relative to the reference	Prevents sequences that appear favorable by leaving loads incomplete
Critical-overstay violation (V^C)	10,000	Sum of positive relative deteriorations in critical process groups	Avoids deterioration in areas with high operational risk
Normalized total net overstay (N^O)	100	Candidate total net overstay divided by reference total net overstay	Reduces system-wide net overstay

Component	Weight	Calculation	Methodological purpose
Normalized P_{90} cycle time (N^P)	10	Candidate P_{90} divided by reference P_{90}	Reduces the upper tail of the cycle-time distribution
Normalized WIP (N^W)	1	Candidate average WIP divided by reference average WIP	Balances the accumulation of loads within the system

The weights encode a strong priority hierarchy rather than mathematically hard constraints. A makespan improvement retains the baseline contribution, whereas a higher makespan incurs a large penalty. The completed-load component promotes sequences that finish all scheduled loads.

After these primary performance requirements have been addressed, candidate solutions are differentiated according to critical-process overstay, total net overstay, P_{90} cycle time, and WIP. This hierarchy discourages substantial extensions of the production day in exchange for marginal reductions in net overstay. The weights are configurable model parameters representing operational preferences and should be evaluated through sensitivity analysis before industrial deployment.

6.5. Acceptance Rule and Escape from Local Minima

The objective value of the candidate solution $Z(r')$ is compared with the objective value of the current solution $Z(r)$. Their difference is calculated using Equation (22).

$$\varDelta = Z(r') - Z(r) \tag{22}$$

According to the Metropolis acceptance rule in Equation (23), a candidate with $\Delta \leq 0$ is accepted unconditionally. When $\Delta > 0$, the worse candidate may still be accepted with a temperature-dependent probability.

$$P_a = \begin{cases} 1 & \text{if} \varDelta \leq 0 \\ \exp\left(-\frac{\varDelta}{T_k}\right) & \text{if} \varDelta > 0 \end{cases} \tag{23}$$

If a random number drawn uniformly from $[0,1)$ is smaller than P_a, the current sequence is replaced by the worse candidate. The best sequence found up to that iteration is retained separately. This mechanism allows the search to leave a local neighborhood during its early, high-temperature stages.

The magnitude of the objective-function penalties directly affects the acceptance probability. At the same temperature, a makespan violation weighted by 1,000,000 produces a substantially lower acceptance probability than a small deterioration in normalized WIP. The cooling schedule and objective weights are therefore calibrated jointly. When the ratio Δ/T_k becomes excessively large, assigning zero acceptance probability maintains numerical stability.

6.6. Temperature and Cooling Schedule

A geometric cooling schedule was used. The initial temperature was set to $T_0 = 10$, the target final temperature to $T_f = 1$, and the iteration limit to $K = 1{,}000$. The cooling coefficient was calculated automatically using Equation (24), giving approximately $\alpha = 0.997700$.

$$\alpha = \left(\frac{T_f}{T_0}\right)^{\frac{1}{K}} \tag{24}$$

The temperature at iteration k is defined in Equation (25).

$$T_k = T_0\alpha^k = 10(0.997700)^k \tag{25}$$

The same temperature schedule and candidate-evaluation budget were applied across all production days, random seeds, and dispatching rules. The selected temperature scale provides a consistent basis for controlled comparison across five empirical workloads, while further parameter tuning can pursue additional performance gains.

6.7. Stopping Criterion, Random Seeds, and Experimental Design

Each optimization run terminated after the predefined limit of 1,000 iterations, and early stopping was disabled. A moving acceptance window covering the final 100 iterations, corresponding to 10% of the search, was monitored only as a convergence diagnostic.

The swap selections, Metropolis acceptance decisions, and remaining stochastic simulation activities were controlled using the random seeds 42, 43, 44, and 45. Each of the five production days was optimized four times under both the Current and S7 dispatching rules. The resulting experimental design comprised

$$5 \text{ days} \times 4 \text{ seeds} \times 2 \text{ rules} = 40 \text{ optimization runs}$$

corresponding to 40,000 candidate evaluations.

6.8. Outputs, Solution Selection, and Methodological Scope

For each run, the reference-run metrics, KPI and acceptance records for all 1,000 candidates, best-sequence CSV file, final simulation summary, and run configuration were stored. To support the resumption of interrupted experimental batches at the completed-run level, metadata files recorded the date, input-file SHA-256 hash, number of loads, random seed, dispatching rule, and iteration count.

The final evaluation jointly considers Z, the number of completed loads, makespan, critical-process net overstay, total net overstay, P_{90} cycle time, WIP, and total blocked time. Run-specific normalization of Z supports within-rule optimization, while unscaled raw KPIs support absolute comparisons between the Current and S7 dispatching rules.

VII. EXPERIMENTAL DESIGN

7.1. Experimental Inputs and Product Mix

The five-day experimental dataset comprises 549 loads. Daily workload ranges from 100 to 122 loads. Each production day is treated as a separate daily scheduling problem. The empirical load records for that day, including the original sequence, route, coating thickness, and load-specific process information, are retained.

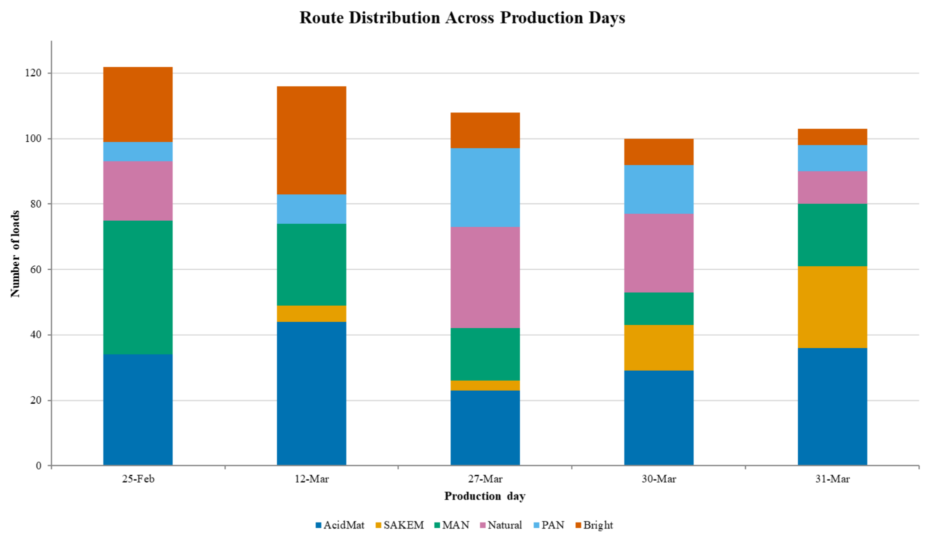

Figure 7.1. Route distribution across the five production days

Table 7.1. Aggregate route and coating-thickness mix over five production days

Dimension	Class	Number of loads	Share of total (%)
Route	AcidMat	166	30.2
Route	SAKEM	47	8.6
Route	MAN	111	20.2
Route	Natural	83	15.1
Route	PAN	62	11.3
Route	Bright	80	14.6

Dimension	Class	Number of loads	Share of total (%)
Coating thickness	7 µm	79	14.4
Coating thickness	13 µm	439	80.0
Coating thickness	25 µm	31	5.6
Total	—	**549**	**100.0**

7.3. Scenarios and Paired Comparison Structure

Table 7.2. Experimental scenarios applied to each production day

Scenario	Load sequence	Hoist rule	Comparison purpose
A	Empirical baseline sequence	Current	Day–seed reference for Current
B	Empirical baseline sequence	S7	S7 reference and A–B rule comparison
C	Simulated Annealing sequence	Current	Change relative to Scenario A for the same day and seed
D	Simulated Annealing sequence	S7	Change relative to Scenario B for the same day and seed

For each day–seed combination, the empirical baseline sequence is first simulated under the relevant dispatching rule. Simulated Annealing is then executed for 1,000 iterations using the same dispatching rule and simulation seed. This paired structure ensures that each candidate sequence is evaluated only against the reference corresponding to the same production day, seed, and dispatching rule.

7.4. Experimental Parameters and Replication

Table 7.3. Standardized parameters used in the five-day experiment

Parameter	Value	Role in the experiment
Production days	5	Introduces variation in product mix and workload across days
Daily loads	100–122; total: 549	Preserves the load set belonging to each day
Dispatching rules	Current and S7	Compares two task-selection approaches under the same physical model
Random seeds	42, 43, 44, and 45	Generates four stochastic search trajectories for each day–rule combination
Optimization runs	40	5 days × 4 seeds × 2 rules
Iterations per run	1,000	Provides an equal search budget for every run
Candidate evaluations	40,000	Simulation-based evaluation of candidate sequences
Neighborhood structure	Two-load swap	Exchanges the loads at two positions in each iteration
Initial/final temperature	10/1	Defines the geometric cooling schedule, with $\alpha \approx 0.997700$
Early termination	Disabled	Allows every optimization run to reach 1,000 iterations
Acceptance window	Final 100 iterations	Used only for convergence monitoring

7.5. Response Variables and Performance Metrics

Table 7.4. Performance indicators monitored in the experiment

Metric	Unit	Methodological definition	Preference
Completed loads	count	Number of scheduled loads reaching the exit conveyor	Must equal the daily total
Makespan	seconds	Elapsed time between the first load entry and final load exit	Lower is better
Total blockage (gross overstay)	seconds	Aggregate positive tank residence beyond nominal completion	Lower is better
Total net overstay	seconds	Aggregate positive overstay after the 45-second allowance for each visit	Lower is better

Metric	Unit	Methodological definition	Preference
Critical-process net overstay	seconds	Net overstay accumulated in the seven critical process groups	Lower is better
P_{90} cycle time	seconds	90th percentile of individual load cycle times	Lower is better
Average WIP	loads	Time-weighted number of loads within the system boundaries	Lower is better
Objective value Z	-	Composite objective normalized to the run-specific baseline	Lower is better

A run is classified as improved in objective-function terms when its final Z value is below the baseline value of 111.0.

7.6. Comparison and Summary Method

For each optimization run, the Simulated Annealing effect is calculated by comparing the final best KPI with the corresponding day–seed–rule baseline. The S7–Current effect is calculated only for the empirical baseline sequence by comparing the S7 and Current results obtained for the same day and seed.

Equation (26) defines the percentage relative change r_K for performance indicator K, where K^e is the experimental value and K^r is the reference value.

$$r_K = 100\left(\frac{K^e}{K^r} - 1\right) \tag{26}$$

The experiment uses a purposively selected set of five production days and four stochastic seeds. Aggregate results are therefore summarized descriptively using the arithmetic mean, minimum–maximum range, and number of successful runs. The four seeds represent internal algorithmic replications using the same empirical daily input. Confirmatory hypothesis testing becomes appropriate after expanding the dataset with statistically independent production-day observations.

VIII. RESULTS

Table 8.1. Summary of the main experimental results

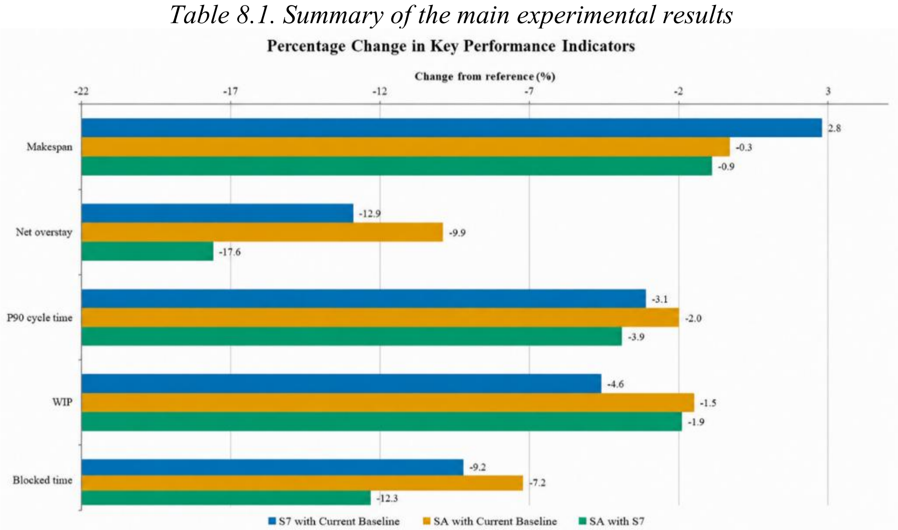

Figure 8.1. Percentage changes in key performance indicators across comparison scenarios

In the best Current run, all 122 loads were completed, while makespan decreased by 1.1%, total net overstay by 34.7%, and total blocked time by 26.2%. In the best S7 run, all 100 loads were completed, while makespan decreased by 1.0%, total net overstay by 34.1%, and total blocked time by 22.8%. These values provide descriptive evidence for the purposively selected experimental dataset and define the empirical scope of inference.

IX. DISCUSSION AND CONCLUSION

All baseline and optimized simulations completed every scheduled load. For the empirical baseline sequences, S7 reduced total net overstay, cycle-time variability, WIP, and blocked time, although it increased makespan. Simulated Annealing improved 12 of 20 Current runs and 15 of 20 S7 runs, while critical-process net overstay decreased across the improved solutions. These findings indicate that load-entry sequencing and dynamic hoist dispatching provide complementary mechanisms for reducing post-nominal waiting and congestion under fixed physical capacity.

Table 10.1 compares the present framework with selected hoist-scheduling studies representing closely related industrial and methodological settings [4,11,12,14]. Unlike studies centered primarily on cyclic scheduling, fleet-size selection, makespan minimization, or deterministic task–hoist optimization, the present study addresses noncyclic daily product mixes and models tank blocking, shared-rail interference, and process-sensitive net overstay endogenously through discrete-event simulation.

Table 9.1. Comparison with selected hoist-scheduling studies

Study	Production setting	Method and primary objective	Relation to the present study
Alsaffar and Suliman [4]	Real anodizing plant with multiple overlapping hoists	Simulation-based heuristic; reduce makespan and increase production	Closest industrial context; performance centers on makespan and production
Laajili et al. [12]	Multi-hoist surface-treatment line with cyclic operation	Variable neighborhood search; minimize cycle time and select hoist count	Optimizes cyclic schedules and fleet size; the present study fixes four hoists and uses daily noncyclic mixes
Ramin et al. [11]	Multi-recipe, multi-stage line with overlapping hoists	Logic-based MILP; optimize makespan and hoist movement under collision constraints	Uses a logical scheduling model; the present study uses discrete-event simulation to measure blocking and net overstay
Chen et al. [14]	Multi-variety electroplating production line	Improved Salp Swarm Algorithm; jointly optimize task order and hoist schedule	Uses coupled planning optimization; the present study combines Simulated Annealing sequencing with dynamic S7 dispatching
Present study	Empirical anodizing line with four hoists on a shared rail	Discrete-event simulation, Simulated Annealing, and S7; reduce net overstay and blocking while completing all loads	Models endogenous blocking, rail interference, and operational-risk indicators under fixed physical capacity

The findings are descriptive and limited to five purposively selected production days, four algorithmic seeds, a 1,000-iteration search budget, and a two-load swap neighborhood. Breakdowns, maintenance, rush orders, initial WIP, and direct defect and economic outcomes were outside the modeled scope. Nevertheless, the results show that simulation-based load sequencing combined with explainable hoist dispatching can reduce post-nominal waiting and congestion without additional tanks or hoists. Future research should extend the model using broader synchronized production and quality data, alternative neighborhood structures, and replicated field validation.

REFERENCES

[1] U.S. Department of Defense (2020), Anodic Coatings for Aluminum and Aluminum Alloy, MIL-PRF-8625F with Amendment 2, Naval Air Systems Command, Lakehurst, USA.

[2] QUALANOD (2026), Specifications for the QUALANOD Quality Label for Sulfuric Acid-Based Anodizing of Aluminium, Edition 01.01.2026, Zurich, Switzerland.

[3] Ofoegbu, S. U., Fernandes, F. A. O. and Pereira, A. B. (2020), "The Sealing Step in Aluminum Anodizing: A Focus on Sustainable Strategies for Enhancing Both Energy Efficiency and Corrosion Resistance", Coatings, 10(3), 226. doi: 10.3390/coatings10030226.

[4] Alsaffar, A. J. and Suliman, S. M. A. (2020), "Hoist Scheduling for an Anodizing Plant at an Extrusion Company", Arab Journal of Basic and Applied Sciences, 27(1), 93–104. doi: 10.1080/25765299.2020.1729931.

[5] Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P. (1983), "Optimization by Simulated Annealing", Science, 220(4598), 671–680. doi: 10.1126/science.220.4598.671.

[6] Akaike, H. (1974), "A New Look at the Statistical Model Identification", IEEE Transactions on Automatic Control, 19(6), 716–723. doi: 10.1109/TAC.1974.1100705.

[7] Schwarz, G. (1978), "Estimating the Dimension of a Model", The Annals of Statistics, 6(2), 461–464. doi: 10.1214/aos/1176344136.

[8] Efron, B. (1979), "Bootstrap Methods: Another Look at the Jackknife", The Annals of Statistics, 7(1), 1–26. doi: 10.1214/aos/1176344552.

[9] SimPy Team (2026), "SimPy Documentation: Overview", https://simpy.readthedocs.io/en/latest/ (Accessed: July 19, 2026).

[10] Sargent, R. G. (2013), "Verification and Validation of Simulation Models", Journal of Simulation, 7(1), 12–24. doi: 10.1057/jos.2012.20.

[11] Ramin, D., Fraizzoli, D., Ballarino, A. and Brusaferri, A. (2023), "Dynamic Hoist Scheduling for Multi-Recipe and Multi-Stage Production Lines: A Logical Framework", Computers & Industrial Engineering, 182, 109360. doi: 10.1016/j.cie.2023.109360.

[12] Laajili, E., Lamrous, S., Manier, M.-A. and Nicod, J.-M. (2021), "An Adapted Variable Neighborhood Search Based Algorithm for the Cyclic Multi-Hoist Design and Scheduling Problem", Computers & Industrial Engineering, 157, 107225. doi: 10.1016/j.cie.2021.107225.

[13] Yan, P., Wang, G., Che, A. and Li, Y. (2016), "Hybrid Discrete Differential Evolution Algorithm for Biobjective Cyclic Hoist Scheduling with Reentrance", Computers & Operations Research, 76, 155–166. doi: 10.1016/j.cor.2016.06.011.

[14] Chen, X., Yang, B., Pang, Z., Zhou, P. and Fu, G. (2024), "Coupled Optimization of Task Sequence and Hoist Scheduling for Electroplating Production Lines Based on an Improved Salp Swarm Algorithm", CIRP Journal of Manufacturing Science and Technology, 53, 34–47. doi: 10.1016/j.cirpj.2024.07.002.

[15] Fargier, H. and Lamothe, J. (2001), "Handling Soft Constraints in Hoist Scheduling Problems: The Fuzzy Approach", Engineering Applications of Artificial Intelligence, 14(3), 387–399. doi: 10.1016/S0952-1976(01)00008-2.