Asian Business Research Journal
Vol. 10, No. 10, 53-60, 2025
ISSN: 2576-6759
DOI: 10.55220/2576-6759.622

Optimal Job Selection and Scheduling in Hybrid Manufacturing Systems using Linear Programming and Sensitivity Analysis

Maha Hasan Sultan

Al-Bayan University (Private), Iraq.

Abstract

This study introduces an interpretable linear model for worker selection and scheduling in hybrid manufacturing that considers profitability, resource usage, and energy simultaneously in the same objective while respecting capacity, sequencing, and time bucket coupling constraints. By assigning tunable weights to selection rewards, use penalties, lateness penalties, and energy costs, the approach supports policy tunability and, through an explicit objective decomposition, reveals marginal effect of each component on the final plan. Empirical application to operational data indicates that such an equilibrium trade-off between value, completion, and delay control is possible, with temporal load staying in effective capacity; behavioral indicators within actual records also suggest a substantial relationship between delays, energy intensity, and machine availability variations. From a management viewpoint, the model offers an reproduceable low-cost decision basis well suited for sensitivity analysis, scenario planning in terms of capacity and energy policy alternatives, and periodic fine tuning to day-to-day fluctuations. Explainability allows integration with learning or metaheuristic elements where higher predictive power and scalability are needed while allowing transparent attributions from parameters to outcomes. Such established limitations as weight calibration dependency and unavoidable process dynamics approximations; yet, the model's expansibility provides for a realistic pathway towards incremental real-world data-driven refinement and establishes groundwork for future extensions, including coupling with learned estimators and more refined logistical constraints.

Keywords: Cost, Energy, Job, Linear, Profitability, Time.

1. Introduction

The combination of additive and subtractive processes in Hybrid Manufacturing Systems in the last few years has boosted customizing, lowered lot sizes, and enhanced variety in products while making scheduling at the same time much more challenging: an attainable schedule must cope with heterogeneous processes (turning and milling and additive and drilling) all together, indefinite processing times, time-varying availability windows of machines, and persistent mismatches between "scheduled" and "actual" shop-floor timestamps. FMS and JS-FMS research has shown static dispatching rules (SPT and EDD and FCFS) suffer with significant plan reality discrepancies in dynamic settings, which lead to data-driven "prediction-then-optimization" pipelines and/or metaheuristic hybrid learning strategies (Abidi et al., 2020; Meilanitasari & Shin, 2021). Conversely, mixed-integer formulations (MILP and MIP), as greatly expressive as they are for loading and selection, routing, and sequence-dependent setups, are computationally demanding and less manager-interpretable in industry scale (Abazari et al., 2012; Roshanaei et al., 2010; Akbaripour et al., 2018). This is precisely where "interpretable linear programming" and "sensitivity analysis" pay a strategic dividend: having readily available operational ranges Processing_Time, Machine_Availability, and Scheduled and Actual timestamps it is possible to build an open model that includes capacity and window limits directly and then, through acceptable ranges on objective coefficients and RHS capacities, evaluate "what-if" situations cheaply and quickly (Shapiro, 1993; Monfared & Yang, 2004; Khan et al., 2021). Industrial experience under real conditions varying from workforce distancing constraints to disconnected parallel machines and transport and AGV integration also demonstrates that adding operational realism without a sensitivity point of view produces brittle, high-risk decisions (Bazargan-Lari et al., 2022; Saidi-Mehrabad et al., 2015; Um et al., 2009; Akbaripour et al., 2018).

Our question directly falls at this intersection: "Optimal Job Selection and Scheduling in Hybrid Manufacturing Systems Using Linear Programming and Sensitivity Analysis," built on three non-proprietary, readily available data pillars processing time (Processing_Time), machine availability (Machine_Availability), and scheduled and actual timestamps (Scheduled and Actual) so that (1) a subset of jobs is selected and assigned to machines with maximum throughput and utilization or minimum total tardiness, (2) capacity constraints are imposed at the machine level within real availability windows, and (3) plan robustness against small parameter perturbations is quantified through sensitivity analysis and represented in managerial terms as "allowable increases and decreases" for objective weights and capacities. Scientifically, this advancement acts as a bridge between two prevailing camps:

heavyweight MILP and metaheuristics that are powerful but costly to compute and hard to interpret on the decision table side versus static rules or simulation-alone research without an "interpretable bridge" to managerial actionability (Byrne & Bakir, 1999; Meilanitasari & Shin, 2021). In our system, objective can be stated to "maximize weighted sum of selected jobs" or "minimize total tardiness and incompleteness"; capacity constraints bind the aggregate processing time on a machine to its "available time"; window constraints cause Actual to adhere to Scheduled (with controllable slack); then sensitivity analysis gives "allowable ranges" on objective coefficients and RHS values so managers know how far they can deviate from the priority weights, utilization targets, or shift and machine capacities without breaking the optimal basis. The contribution thus is twofold: an implementable "baseline LP" for HMS operating over small data domains, and a "robustness map" that distinguishes between safe vs. risky parameter moves precisely what decision makers would desire under real world constraints like shift changes, periodic faults, or safety considerations.

2. Literature Review

The flexible setting scheduling literature investigates a few options. At one level, ML and metaheuristics-driven methods select and predict dispatching rules dynamically and optimize predictive accuracy; e.g., Abidi et al. combine weighted feature extraction with a hybrid fuzzy DBN classifier and a lion algorithm variation to propose rules in FMS, with accuracy gains from combining metaheuristics and deep learning (Abidi et al., 2020). Meilanitasari and Shin's review highlights that static policies (SPT and EDD and FCFS) create significant gaps under JS-FMS dynamics and encourages "prediction then-optimization" with sequence learning to bridge the uncertainty and optimal scheduling gaps (Meilanitasari & Shin, 2021). In contrast, well-defined mathematical models are used: Abazari et al. propose a hybrid continuous and 0-1 programming model with a GA for machine loading to optimize profitability and utilization within capacity, batch size, processing time, tool, and magazine constraints (Abazari et al., 2012). Roshanaei et al. formulate JSS with sequence-dependent setup times as a MILP to optimize makespan and apply an electromagnetism like algorithm for the large instances (Roshanaei et al., 2010). In cloud manufacturing, Akbaripour et al. develop service selection and scheduling over mixed composition structures (sequence and parallel and loop and selective), combine service occupancy and transportation on hybrid hub and spoke networks, and demonstrate that adding transportation and availability offers more realistic solutions; sensitivity analyses also estimate policy robustness (Akbaripour et al., 2018). Methodologically, the ancient underpinning of LP sensitivity to coefficient and RHS changes underlies implemented "what-if" analyses (Shapiro, 1993; cf. Monfared & Yang, 2004 on fuzzy scheduling and control sensitivity and parameter tuning)

The field also leans towards hybrids: combining mathematical models with simulation and metaheuristics to achieve high quality, scalable solutions. Examples include neural networks with simulated annealing for stochastic job shops (Tavakkoli Moghaddam et al., 2005), hybrid simulation analytical models of multi period, multi product planning (Byrne & Bakir, 1999), and joint scheduling maintenance models with multiobjective search (Mishra et al., 2022; also Tirkolaee et al., 2020 for energy-aware JIT). In FMS and JS-FMS, other papers introduce realism: integrated JSS with conflict-free AGV routing (Saidi-Mehrabad et al., 2015), FMS with AGVs and multiobjective ES and MONLP (Um et al., 2009), and GA–TOPSIS simulation for operator assignment (Azadeh et al., 2011). At the design level, RSM and BWM frameworks make FMS design parameters flexible and optimize performance vs. deployment cost (Pasha et al., 2023). Recent RMS and Cloud studies indicate that estimation of actual availability, transportation, and reconfigurability significantly alters schedule and planning decisions, and sensitivity and ANOVA are key in parameter effect quantification (Imsetif et al., 2025; Yazdani et al., 2022). Multiobjective studies during the COVID period include such constraints as social distancing of workers into parallel machines models and provides direct impacts on profit and annual scheduling (Bazargan-Lari et al., 2022). In additive processes, scheduling non-identical parallel SLM machines with makespan and tardiness objectives and a learning-based NSGA-II is the merging of the field towards "explicit model + learning" hybrids (Rohaninejad et al., 2021). Generally, the literature shows: (1) MILP and metaheuristics excel with complete constraints but sometimes sacrificing interpretability and low cost sensitivity; (2) static rules fall short in dynamic settings, with the need for Scheduled and Actual data (Meilanitasari & Shin, 2021); and (3) interpretable LP with complete sensitivity can provide a reproducible baseline to HMS decision-making, especially where only usual operational parameters such as Processing_Time, Machine_Availability, and Scheduled and Actual are accessible (Shapiro, 1993; Monfared & Yang, 2004; Khan et al., 2021). Therefore our gap an LP formulation for "job selection and scheduling" in HMS based on public operational fields and reporting allowable ranges on coefficients and capacities addresses two requirements directly: analytical transparency for managers and simplicity with real world data for fast, low cost deployment.

3. Data and Methodology

3.1. Study Data

The data set consists of actual planning and execution data from a hybrid manufacturing system, where each row is a production job with a unique identifier and includes the operational and temporal attributes necessary for linear modeling and sensitivity analysis. For each operation, operation type (milling, drilling, lathe, additive, or grinding) and preassigned machine (Machine_ID) are entered; Processing_Time is the typical job time in base time units; Machine_Availability is documented as percent or effective capacity rating, converted to per interval machine capacity; Scheduled_Start and Scheduled_End define the planned window, and Actual_Start and Actual_End denote the actual timestamps (used for calibration and evaluation). Energy_Consumption per job identifies energy intensity per unit time of processing, while Job_Status and Optimization_Category are only used for weighting or ex-post evaluation of performance. In order to transform raw data into model inputs, the planning horizon [H] is discretized into a uniform time grid with interval length Δ; by representing calendar timestamps as bucket indices, each job's admissible window along the time axis is defined. Per-machine, per-bucket capacity limit $\mathrm{cap}_{m,t}$ is calculated from $\mathrm{cap}_{m,t} = \Delta \cdot \frac{\mathrm{Availability}_{m,t}}{100}$. Processing times p_j are simply read from Processing_Time following unit harmonization to base minute or hour. Actual_* timestamps are not used as constraints for the purpose of keeping the model predictive and deployable; instead, they will be applied once optimization is done in order to evaluate and calibrate the penalties for early start and late finish.

3.2. Modeling and Solution Method

We propose a proactive, time-indexed linear program that simultaneously optimizes the selection of jobs, allocation of capacity per machine, and adherence to planned windows. Allowing preemption (that is, splitting a job over multiple time buckets) enables a linear, free binary formulation at no increased complexity and yet with only temporal stickiness through the introduction of penalties for processing outside the planned window. We then define index sets and parameters, decision variables, objective, and constraints.

Index sets:

$J = jobs.$

M = set of machines.

T = set of discrete time buckets $\{1, 2, \ldots, H\}$.

For each j in J, let $m(j)$ in M be its preassigned machine. For each j in J, let $T_{\text{on}}(j)$= buckets between Scheduled_Start_j and Scheduled_End_j, $T_{\text{early}}(j)$ = buckets preceding Scheduled_Start_j, and $T_{\text{late}}(j)$= buckets following Scheduled_End_j.

3.3. Parameters

Δ = length of each time bucket.

p_j= typical processing time of job j in units of Δ.

$\text{cap}_{m,t}$ = capacity of machine m available in bucket t (in units of Δ), obtained from Machine_Availability and work calendar.

e_j = energy rate of job j per unit of processing time.

w_j = job weight and priority (in simplest case w_j = 1 or by Optimization_Category).

α, β, γ, δ, κ = nonnegative objective weights trading off throughput, lateness, idle capacity, energy, and undercompletion.

3.4. Decision Variables

$x_j \in [0,1]$= job j selection variable (1: completely processed; fractional values: partially processed in the horizon).

$y_{j,t}^{\text{on}} \geq 0$= amount of processing of job j in bucket t in the scheduled window ($t \in T_{\text{on}}(j)$).

$y_{j,t}^{\text{early}} \geq 0$= early processing of job j ($t \in T_{\text{early}}(j)$).

$y_{j,t}^{\text{late}} \geq 0$= late processing of job j ($t \in T_{\text{late}}(j)$).

$o_{m,t} \geq 0$= overtime for machine m in bucket t (discouraged through a big penalty).

$\text{idle}_{m,t} \geq 0$= idle capacity of machine m in bucket t (slack to quantify unused capacity).

$u_j \geq 0$= undercompletion of job j in the horizon (penalized to make it desirable to complete even if x_j is fractional).

3.5. Aggregate Helpers

$y_{j,t} = y_{j,t}^{\text{on}} + y_{j,t}^{\text{early}} + y_{j,t}^{\text{late}}, \forall t \in T$ (and for buckets not in the specified subsets, the respective components are zero by default).

$E_j = \sum_{t \in T_{\text{early}}(j)} y_{j,t}^{\text{early}}$ (total early processing of job j).

$L_j = \sum_{t \in T_{\text{late}}(j)} y_{j,t}^{\text{late}}$ (total late processing of job j).

$Y_j = \sum_{t} y_{j,t}$ (total allocated processing of job j over the horizon).

3.6. Objective Function

$$\text{Maximize}\, Z = \alpha \cdot \sum_{j \in J} w_j x_j - \beta \cdot \sum_{j \in J} L_j - \gamma \cdot \sum_{m \in M} \sum_{t \in T} idle_{m,t} - \delta \cdot \sum_{j \in J} e_j Y_j - \kappa \cdot \sum_{j \in J} u_j$$

The first encourages throughput and job selection; the second inhibits processing with delay; the third reduces idle capacity; the fourth imposes a cost for energy usage; and the fifth imposes a cost for undercompletion to make the plan reach completion even if x_j is fractional.

3.7. Constraints

Job processing balance (flow of each job through time): $\forall j \in J$: $\sum_{t \in T} y_{j,t} + u_j = p_j x_j$ Machine capacity per bucket (including idle measurement and overtime control): $\forall m \in M, \forall t \in T$: $\sum_{j \in J: m(j) = m} y_{j,t} + \text{idle}_{m,t} \leq \text{cap}_{m,t} + o_{m,t}$.

Unavailability of machines (optional hardening of constraint 2):

$\forall m \in M, \forall t \in T \text{ with } cap_{m,t} = 0 : y_{j,t} = 0$

For each j such that $m(j) = m$ and $o_{m,t} = 0$ and $\text{idle}_{m,t} \geq 0$ (this line is a reminder that with zero cap only idle can equal cap and y and o are zero.)

Splitting up processing by temporal region and lateness and earliness definition:

$$\forall j \in J: E_j = \sum_{t \in T_{\text{early}}(j)} y_{j,t}^{\text{early}}$$
$$\forall j \in J: L_j = \sum_{t \in T_{\text{late}}(j)} y_{j,t}^{\text{late}}$$
$$\forall j \in J, \forall t \in T_{\text{on}}(j): y_{j,t}^{\text{on}} \geq 0$$
$$\forall j \in J, \forall t \in T_{\text{early}}(j): y_{j,t}^{\text{early}} \geq 0$$
$$\forall j \in J, \forall t \in T_{\text{late}}(j): y_{j,t}^{\text{late}} \geq 0$$
$$\forall j \in J, \forall t \in T \backslash (T_{\text{on}}(j) \cup T_{\text{early}}(j) \cup T_{\text{late}}(j)): y_{j,t}^{\text{on}} = y_{j,t}^{\text{early}} = y_{j,t}^{\text{late}} = 0$$

3.8. Bounds and Nonnegativity

$$\forall j \in J: 0 \leq x_j \leq 1$$
$$\forall j \in J: u_j \geq 0$$
$$\forall m \in M, \forall t \in T: \text{idle}_{m,t} \geq 0$$
$$\forall m \in M, \forall t \in T: o_{m,t} \geq 0$$
$$\forall j \in J, \forall t \in T: y_{j,t}^{\text{on}} \geq 0$$
$$\forall j \in J, \forall t \in T: y_{j,t}^{\text{early}} \geq 0$$
$$\forall j \in J, \forall t \in T: y_{j,t}^{\text{late}} \geq 0$$

Policy control of overtime caps: $\forall m \in M, \forall t \in T: \text{no}_{m,t} \leq \text{Omax}_{m,t}$ where $\text{Omax}_{m,t}$ is policy-defined (usually zero or a small percentage of $\text{cap}_{m,t}$).

Critical jobs (optional hard finish or soft slack): $\forall j \in J_{\text{critical}} \subseteq J: x_j = 1 \vee \sum_{t \in T} y_{j,t} \geq p_j x_j - \text{Uslack}_j$ with high penalty on Uslack_j in the objective.

Hard time windows (if strict timing is essential instead of penalties):

$$\forall j \in J: y_{j,t}^{\text{early}} = 0 \text{ for all } t \in T_{\text{early}}(j)$$
$$y_{j,t}^{\text{late}} = 0 \text{ for all } t \in T_{\text{late}}(j)$$

(in this case, L_j drops out of the objective and only y^{on} is admissible.)

Choosing Δ: smaller Δ increases temporal fidelity but increases model size; standard Δ in the 5–15 minutes range.

Extrapolating Machine_Availability to $\text{cap}_{m,t}$}: if in percentage form, $\text{cap}_{m,t} = \Delta \cdot \frac{\text{Availability}_{m,t}}{100}$; off-shift buckets get cap = 0.

Energy: if Energy_Consumption is a per-job total, scale to $\hat{e}_j = \frac{\text{Energy\_Consumption}_j}{p_j}$ so that $\delta \cdot \sum_t \hat{e}_j y_{j,t}$ is meaningful.

Excessive fragmentation avoidance: if contiguity in time matters, one can impose soft penalties on inter bucket variance of $y_{j,t}$; to maintain linearity, penalizing early and late aggregates typically does the trick.

Single-machine per job: as Machine_ID is fixed in the data, no assignment constraint is needed; if some operations are multi option, generalize to y_{jmt} and write $\sum_m \sum_t y_{jmt} + u_j = p_j x_j$ with capacity per (m,t) (still linear and preemptive).

Sensitivity results: post determine an admissible range on objective coefficients (w_j, δ, β) and RHS values ($\text{cap}_{m,t}$); this is beyond the formulation and requires post-optimal analysis of the LP solver.

In the ideal plan, x_j is the portion of job j which was scheduled and finished by the horizon; $y_{j,t}$ is the time capacity plan on machine $m(j)$; L_j is a measure of how much late processing was required to finish the job; $\text{idle}_{m,t}$ reports unused capacity, acts as a control to fill shifts; and u_j measures undercompletion so managers could balance between serving more orders on Windows and energy consumption.

3.9. Data Analysis

The discussion begins with scenario behavior and the overall impact on the objective, selected jobs, utilization, and costs; followed by objective composition; followed by job-level schedule quality and time bucket occupancy; then plan alignment with actual execution; finally, capacity slack.

From Table 1, increasing alpha raises the objective monotonically while selected_jobs stays at 144 in most combinations. Increasing kappa from 50 to 150 slightly lifts avg_utilization; with $kappa = 300$ the usage penalty total_u pushes total_L upward. This is most severe at alpha=400, where selected_jobs drops to 114. The preferred region is alpha∈{700,1000} with $kappa \in \{50,150\}$, balancing a high objective, full completion, and controlled delay.

To clarify the objective, Table 2 shows alphasum(wx) as the dominant positive term, and -kappasum(u) is the largest subtraction; -betasum(L) and -deltasum(energyYDelta) also reduce the total but to a lesser extent. This accounts for the fact that mid-range kappa performs better than very high kappa in Table 1: usage must be reined in, but over-penalizing inflates total_L and wreaks havoc on objective.

Table 1. Parametric scenarios.

Alpha	Kappa	Beta	Gamma	Delta	Cap_Scale	Objective	Selected_Jobs	Total_U	Total_L	Total_Energy	Avg_Utilization
400	50	10	0.1	0.05	1	47065.92107	144	102.8534885	0.365	555.0900639	0.553998707
400	150	10	0.1	0.05	1	36932.80242	144	100.7334697	16.875	568.5425474	0.571665531
400	300	10	0.1	0.05	1	25384.25683	114	48.97521156	134.6281429	521.7602717	0.52236105
700	50	10	0.1	0.05	1	86245.92107	144	102.8534885	0.365	555.0900639	0.553998707
700	150	10	0.1	0.05	1	76112.80242	144	100.7334697	16.875	568.5425474	0.571665531
700	300	10	0.1	0.05	1	61135.30115	144	98.23120246	78.59	588.7622905	0.592517757
1000	50	10	0.1	0.05	1	125425.9211	144	102.8534885	0.365	555.0900639	0.553998707
1000	150	10	0.1	0.05	1	115292.8024	144	100.7334697	16.875	568.5425474	0.571665531
1000	300	10	0.1	0.05	1	100315.3011	144	98.23120246	78.59	588.7622905	0.592517757

Table 2. Objective composition.

Component	Value
alphasum(wx)	130600
- beta*SUM(L)	-245.8333333
- deltaSUM(energyY*Delta)	-28.50065745
- kappa*SUM(u)	-20051.42204
TOTAL (objective)	110274.244

Table 3. Job selection summary (sample).

Job_ID	Machine_ID	Operation_ Type	Processing_ Time	Energy_ Consumption	Scheduled_ Start	Scheduled_ End	Job_ Status	Optimization_ Category	x	u	L	Y_ sum	Total_Energy
J001	M01	Grinding	76	11.42	3/18/2023 8:00	3/18/2023 9:16	Completed	Moderate Efficiency	1	1.2667	0	0	0
J002	M01	Grinding	79	6.61	3/18/2023 8:10	3/18/2023 9:29	Delayed	Low Efficiency	1	0.4267	0	1.78	5.8829
J014	M04	Additive	112	2.01	3/18/2023 10:10	3/18/2023 12:02	Completed	Optimal Efficiency	1	0.0867	5.76	3.56	3.5778

Table 4. Time-bucket schedule Y (sample).

Job_ID	Machine_ID	bucket_start	bucket_end	Y
J002	M01	3/18/2023 8:00	3/18/2023 8:30	0.89
J011	M01	3/18/2023 9:30	3/18/2023 10:00	0.89
J016	M01	3/18/2023 10:30	3/18/2023 11:00	0.89
J058	M01	3/18/2023 18:00	3/18/2023 18:30	0.242833333
J060	M01	3/18/2023 18:00	3/18/2023 18:30	0.647166667

Table 5. Hybrid manufacturing categorical (sample).

Job_ID	Machine_ID	Operation_ Type	Material_ Used	Processing_ Time	Energy_ Consumption	Machine_ Availability	Scheduled_ Start	Scheduled_ End	Actual_ Start	Actual_ End	Job_ Status	Optimization_Category
J001	M01	Grinding	3.17	76	11.42	96	3/18/2023 8:00	3/18/2023 9:16	3/18/2023 8:05	3/18/2023 9:21	Completed	Moderate Efficiency
J010	M01	Drilling	2.10	27	3.66	97	3/18/2023 9:30	3/18/2023 9:57	3/18/2023 9:54	3/18/2023 10:21	Delayed	Low Efficiency
J014	M04	Additive	2.33	112	2.01	95	3/18/2023 10:10	3/18/2023 12:02	3/18/2023 10:10	3/18/2023 12:02	Completed	Optimal Efficiency

For quality at the job level, Table 3 (lp selected subj summary) indicates that most rows have x=1 and Job_Status=Completed. Values of u and L indicate pressure from machine usage and delay risk. Rows marked Optimal or High Efficiency tend to have lower energy with controlled L, whereas Low Efficiency is found with higher L or energy.

To align plan and reality, Table 5 (hybrid manufacturing categorical) shows Scheduled vs Actual times. Delayed entries tend to coincide with higher energy or lower machine availability. Together with Table 3, this indicates that re-ordering toward lower-energy operations and tighter availability constraints can reduce delay incidence.

To complete the capacity snapshot, Table 6 (idle by time-bucket) shows zero Idle in the sampled intervals; this aligns with Y distribution and avg_utilization rates in Table 1. Further gains are expected to arise from reducing usage and energy costs as well as avoiding Delayed/Failed instances rather than filling idle holes.

Table 6. Idle by time-bucket (sample).

Machine_ID	bucket_start	bucket_end	Idle
M01	3/18/2023 8:00	3/18/2023 8:30	0
M01	3/18/2023 10:30	3/18/2023 11:00	0
M01	3/18/2023 14:30	3/18/2023 15:00	0

4. Discussion and Conclusion

The combined evidence from Table 1 to Table 6 indicates that a transparent linear formulation with weights alpha, kappa, beta, gamma, and delta yields an operational balance between objective value, completion, and delay control. Within reasonable levels of weights, the objective increases while preserving the set of completions; excessive use penalties at high levels degrade performance by raising total_u and, in turn, total_L. This is in accordance with Table 2: alphasum(wx) is the main driver, whereas -kappa*sum(u) is the main deduction and, if overdone, overstates delays and makes the objective worse. Temporal load in Table 4 shows Y staying within effective capacity, as is expected from avg_utilization in Table 1. At the job level, Table 3 reveals that Optimal and High Efficiency options generally pair with less energy and controlled delay, and Low Efficiency pairs with more usage pressure or energy consumption. Plan-versus-actual agreement in Table 5 confirms that delayed cases often overlap with more energy or less availability. Finally, Table 6 shows no major idle pockets, hence improvement lever is towards weight tuning and reordering towards lower energy operations.

Contrary to literature, the results both support and augment three prevalent themes. First, learning/metaheuristic models adaptively choose dispatching rules and have reported enhancements in predictive accuracy. Our findings via Table 3 and Table 4 show that without learning, a replicable and interpretable baseline can indeed be established and served as a foundation for "prediction then optimization" hybrids. Second, well-defined mathematical models used in the past often favor MILP with sequence-dependent setups and transportation, and search heuristics to ensure scalability. Here, planar structure and objective decomposition in Table 2 retain explainability and sensitivity at little computational cost, which is a luxury for rapid deployment to HMS. Third, energy-aware and availability-aware models drastically change scheduling decisions. The consistency across Table 1, Table 3, and Table 5 sends the same message: with energy and capacity weighted properly, both delay and consumption are both optimized together.

The lesson of operation is to keep weights in a well-balanced band so that the target stays high and risk of delay is controlled, with this plausible baseline being a stepping stone toward learning or metaheuristic optimization. Adaptive retuning of weights against Actual signals and efficiency labels in Table 5 is a plausible next step, while using Table 4 to recognize congested time windows and resequence towards lower energy strings. This reaffirms compatibility with the "explicit model + learning" hybrids emphasized in the literature.

An understandable linear model introduced objective benefit, work completion, and delay control into equilibrium. The objective breakdown highlighted selection incentives' key role and extreme sensitivity to use penalties. Time-bucket scheduling kept load in efficient capacity, with real execution data correlating delays and energy use with availability. The approach presents an open, low cost foundation waiting for incorporation of learning and metaheuristic capabilities.

References

Abazari, A. M., Solimanpur, M., & Sattari, H. (2012). Optimum loading of machines in a flexible manufacturing system using a mixed-integer linear mathematical programming model and genetic algorithm. *Computers & Industrial Engineering, 62*(2), 469-478. https://doi.org/10.1016/j.cie.2011.10.013

Abidi, M. H., Alkhalefah, H., Mohammed, M. K., Umer, U., & Abu Qudeiri, J. E. (2020). Optimal scheduling of flexible manufacturing system using improved lion-based hybrid machine learning approach. *IEEE Access, 8*, 96088–96114.

Akbaripour, H., Houshmand, M., van Woensel, T., & Mutlu, N. (2018). Cloud manufacturing service selection optimization and scheduling with transportation considerations: Mixed-integer programming models. *The International Journal of Advanced Manufacturing Technology, 95*(1), 43-70. https://doi.org/10.1007/s00170-017-1167-3

Azadeh, A., Kor, H., & Hatefi, S.-M. (2011). A hybrid genetic algorithm–TOPSIS–computer simulation approach for optimum operator assignment in cellular manufacturing systems. *Journal of the Chinese Institute of Engineers, 34*(1), 57-74. https://doi.org/10.1080/02533839.2011.552966

Bazargan-Lari, M. R., Taghipour, S., Zaretalab, A., & Sharifi, M. (2022). Production scheduling optimization for parallel machines subject to physical distancing due to COVID-19 pandemic. *Operations Management Research, 15*, 503-527.

Byrne, M. D., & Bakir, M. A. (1999). Production planning using a hybrid simulation-analytical approach. *International Journal of Production Economics, 59*(1-3), 305-311. https://doi.org/10.1016/S0925-5273(98)00104-2

Imsetif, J., Nezamoddini, N., & Aqlan, F. (2025). Job shop planning and scheduling of reconfigurable manufacturing systems. *Operations Management Research, 18*, 891-912. [Add issue number, DOI when available]

Khan, A. S., Naeem, K., & Khan, R. U. (2021). A comparison between dedicated and flexible manufacturing systems: Optimization and sensitivity analysis. *Mehran University Research Journal of Engineering & Technology, 40*(1), 130-139. https://doi.org/10.22581/muet1982.2101.12

Meilanitasari, P., & Shin, S.-J. (2021). A review of prediction and optimization for sequence-driven scheduling in job shop flexible manufacturing systems. *Processes, 9*, 1391. https://doi.org/10.3390/pr9101391

Mishra, A. K., Shrivastava, D., Tarasia, D., & Rahim, A. (2022). Joint optimization of production scheduling and group preventive maintenance planning in multi-machine systems. *Annals of Operations Research, 316*, 401-444. https://doi.org/10.1007/s10479-021-04492-1

Monfared, M. A. S., & Yang, J. B. (2004). Design of an intelligent manufacturing scheduling and control system using fuzzy logic: Sensitivity analysis and parameter optimization. *Journal of Intelligent & Fuzzy Systems, 15*(2)

Pasha, N., Mahdiraji, H. A., Razavi Hajiagha, S. H., Garza-Reyes, J. A., & Joshi, R. (2024). A multi-objective flexible manufacturing system design optimization using a hybrid response surface methodology. *Operations Management Research, 17*(1), 135–151. https://doi.org/10.1007/s12063-023-00412-w

Rohaninejad, M., Tavakkoli-Moghaddam, R., Vahedi-Nouri, B., Hanzálek, Z., & Shirazian, S. (2021). A hybrid learning-based meta-heuristic algorithm for scheduling of an additive manufacturing system consisting of parallel SLM machines. *International Journal of Production Research, 59*(19), 6205-6225. https://doi.org/10.1080/00207543.2020.1844884

Roshanaei, V., Khaleghei Ghosheh Balagh, A., Seyyed Esfahani, M. M., & Vahdani, B. (2010). A mixed-integer linear programming model along with an electromagnetism-like algorithm for scheduling job shop production system with sequence-dependent set-up times. *The International Journal of Advanced Manufacturing Technology, 47*, 783-793. https://doi.org/10.1007/s00170-009-2372-0

Saidi-Mehrabad, M., Dehnavi-Arani, S., Evazabadian, F., & Mahmoodian, V. (2015). An ant colony algorithm for solving the integrated model of job shop scheduling and conflict-free routing of AGVs. *Computers & Industrial Engineering, 86*, 2-13. https://doi.org/10.1016/j.cie.2014.11.025

Shapiro, J. F. (1993). Mathematical programming models and methods for production planning and scheduling. In M. C. Fu, H. J. Greenberg, & P. M. Pardalos (Eds.), *Handbooks in Operations Research and Management Science* (Vol. 4, pp. 371-443). Elsevier.

Tavakkoli-Moghaddam, R., Jolai, F., Vaziri, F., Ahmed, P. K., & Azaron, A. (2005). A hybrid method for solving stochastic job shop scheduling problems. *Applied Mathematics and Computation, 170*(1), 185-206. https://doi.org/10.1016/j.amc.2004.11.044

Tirkolaee, E. B., Goli, A., & Weber, G.-W. (2020). Fuzzy mathematical programming and self-adaptive artificial fish swarm algorithm for just-in-time energy-aware flow shop scheduling problem with outsourcing option. *IEEE Transactions on Fuzzy Systems, 28*(11), 2772-2783. https://doi.org/10.1109/TFUZZ.2020.3006040

Um, I., Cheon, H., & Lee, H. (2009/2010). The simulation design and analysis of a flexible manufacturing system with automated guided vehicle system. *Journal of Manufacturing Systems, 28*(4), 115-122. https://doi.org/10.1016/j.jmsy.2010.02.002

Yazdani, M. A., Khezri, A., & Benyoucef, L. (2022). Process and production planning for sustainable reconfigurable manufacturing systems (SRMS): Multi-objective exact and heuristic-based approaches. *The International Journal of Advanced Manufacturing Technology, 119*, 4519-4540. https://doi.org/10.1007/s00170-022-10865-1