Highlights

AGV scheduling in shopfloor logistics of semiconductor assembly

Júlia Bergmann, András Kovács, József Váncza

- A phase-informed AGV dispatching model for semiconductor back-end assembly, in which material flows are harmonized across process phases rather than optimizing moves individually.
- Bucketing heuristics and a speeded-up insertion procedure for constructing effective AGV tours with low runtime, which facilitates large-scale scenario testing.
- A major simulation study with dozens of configurations and capacities and multiple iterations, demonstrating steady improvement in throughput, especially when the aisle configuration and AGV capacity are optimized.
- A deployable-in-practice advice: when it is beneficial to bucket, when it is not, and how to select rules based on phase bottlenecks and facet attributes.
- Decision support prototype (web/demo) along with clear parameterization to facilitate replication and portability to other discrete-parts factories.

AGV scheduling in shopfloor logistics of semiconductor assembly

Júlia Bergmann[a,b,*], András Kovács[b], József Váncza[b]

[a] *Eötvös Loránd University, Budapest, Hungary*
[b] *HUN-REN Institute for Computer Science and Control, Budapest, Hungary*

Abstract

Automated Guided Vehicles (AGVs) are integral to indoor logistics, especially in uncertain manufacturing settings, such as semiconductor fabrication and automated warehouses. Traditional AGV dispatching strategies aim at transportation efficiency, but they overlook problem structure stemming from production process characteristics. This paper introduces a new family of logistical task scheduling algorithms that considers production phases. We propose a set of AGV dispatching strategies and low-latency scheduling heuristics. Specifically, we present a so-called bucketing scheme that groups logistical tasks based on their association with a production phase and facility layout. Through detailed simulations, we demonstrate that phase-aware scheduling significantly increases throughput, while no single strategy is ideal across all scenarios. Bucketing also becomes a practical replacement for predictive models in fitting environments. Our findings present a comprehensive scheduling framework for shopfloor logistics that captures real-time requirements of production and provides an optimization approach with reasonable computational effort.

Keywords: Semiconductor manufacturing, internal logistics, AGV, scheduling, dispatching

*Corresponding author.

Email addresses: julia.bergmann@sztaki.hu (Júlia Bergmann), andras.kovacs@sztaki.hu (András Kovács), jozsef.vancza@sztaki.hu (József Váncza)

1. Introduction

In the grand scheme of modern manufacturing and warehouse logistics, Automated Guided Vehicles (AGVs) are the unseen actors – gliding through corridors and hauling materials to perform the dance of just-in-time shopfloor logistics. AGVs have evolved from unassuming containers for carrying products to important players within an smart manufacturing system. However, as manufacturing velocity increases and choreography becomes more and more sophisticated, a question arises: How can we define optimality when directing AGVs, and also how can it be achieved?

This work rethinks AGV dispatching and scheduling as time-based orchestration, not just routing. By incorporating insight of production phases into dispatching logic, a connection between the logic of production and internal logistics is established. Consequently, scheduling becomes a real-time, dynamic negotiation between space, layout, time, demand, and capacities. To that end, we introduce a family of phase-aware and grouping heuristics and study their effect across a wide range of manufacturing scenarios, focusing on simple dispatching rules, with the main contribution lying in the resulting conceptual and computational simplicity. Our experiments reveal that the interaction between transport structure and production timing is fundamental, not peripheral. We demonstrate that optimality originates in strategies that adjust to spatial layout and the intangible cadence of production demand. Above all, though, this work provides more than algorithmic breakthroughs; it provides a new approach to indoor logistics. In this approach, intelligence is not measured by complexity, but by a system's ability to pay attention to timing, context, and the soft pulse of the factory floor.

1.1. Main findings

This study examines several aspects of AGV dispatching and task scheduling within indoor logistics, specifically in cases where transportation must be coordinated with multi-stage manufacturing phases. The main findings at the end of this research are the followings:

- *Production-phase awareness supports dispatching.*
  Dispatching strategies that consider both where materials are needed and when they are needed lead to better coordination and higher system throughput. Also, combined strategies outperform solely distance-based ones in environments with oversubscribed resource levels, or uneven or changing demand.

- *Context matters – strategy performance depends on system characteristics.*
  Universally optimal dispatching policy is not found yet. The optimality of a policy depends on the structural characteristics of the production system.

- *Simple grouping heuristics can improve performance.*
  Pre-grouping logistical tasks based on machine types or production phases improves decision-making efficiency, reduces computational overhead, and expands the horizon of task scheduler without requiring precise prediction of future demand.

- *Matching AGV capacity with facility layout improves overall facility productivity.*
  When AGV capacity matches system layout, specific strategies can perform much better, highlighting the value of "hardware-software" coordination.

1.2. Paper structure

The remainder of the paper is organized as follows: after the literature review (Section 2), Section 3 introduces the manufacturing setup and outlines the key assumptions that define the logistics system under investigation. Section 4 describes the proposed dispatching strategies, including phase-aware and load-based heuristics and the mechanism of logistical task grouping. Section 5 explains the simulation framework used for performance evaluation. Then, it presents and analyzes the experimental results, focusing on how each strategy performs under different system scenarios. Section 6 addresses practical implementation considerations offering the prototype of a decision support tool. Finally, Section 7 concludes the paper by reflecting on the main findings in relation to the initial hypotheses and suggesting ways for future research.

2. Literature review

The appropriate scheduling and dispatching of AGVs plays a critical role in the operational efficiency in semiconductor manufacturing and warehouse logistics. Effective AGV management can dramatically reduce costs, shorten cycle times, and boost productivity [1, 2, 3]. Another review, [4] highlights that scheduling in semiconductor manufacturing remains computationally challenging due to reentrant flows, batch processing, and complex resource constraints, with surveys systematically reviewing rule-based and advanced optimization methods across single-machine to job-shop settings.

AGV scheduling optimization is closely related to classical combinatorial problems, particularly the Traveling Salesman Problem (TSP) and the Vehicle Routing Problem (VRP). These problems are used as mathematical tools to model task scheduling.In the case of a single vehicle, AGV scheduling can be modeled as a variant of the TSP, where the delivery and pick up points are visited in an optimal sequence. For example, substitute delivery and pick up node formulations have helped AGV task allocation using mixed-integer programming approaches [5]. VRP models generalize these concepts for multi-AGV systems by considering task allocation among several vehicles, capacity, and service precedence. Recently, there has been an increase in survey articles demonstrating how VRP variants have been applied to industrial use cases, including AGV fleets, using constraint-based and multi-agent optimization approaches [6]. Additionally, hybrid methods combining dynamic programming and learning-based heuristics have been shown to effectively solve large TSP and VRP instances and yield scalable, real-time AGV control strategies [7]. Subsequent research aimed to enhance the control and schedulability of AGVs in dynamic production and logistics environments. [8] introduced a centralized, conflict-free control approach for AGV fleets servicing online transportation requests. This approach ensures verifiable deadlock- and livelock-freedom while minimizing request tardiness to a negligible level, and balances global optimality and computational efficiency, improving upon existing centralized schemes in difficult layout scenarios. In a more recent research, a common modeling and simulation methodology for dynamic vehicle routing problems (DVRPs) was developed on a large scale to provide a flexible medium to evaluate and compare routing policies in an uncertain environment [9]. This methodology can accommodate a wide range of problem features, such as stochastic travel times, service time windows, and decision deferral, and it serves as open-source software that can analyze decision-making in real time. Other recent work highlights that discrepancies between digital twin models and real systems can significantly impact decision quality, making calibration essential for reliable deployment in AMHS environments [10].

Basic dispatching rules for the wafer fab of semiconductor manufacturing are introduced in [11], and the paper emphasizes the necessity of adaptive dispatching in the face of complex, dynamic production demands. [12] studied AGV performance and attributes specifically in the semiconductor fab setting, identifying accurate operational profiling as a critical requirement for effective dispatching. Building on these findings, [13] developed dynamic control strategies for multimodal AGVs, addressing the integration of various transportation modes and the advantages of adaptive, multimodal dispatch strategies. Such multimodal integration enables more robust and flexible operations. Furthermore, [14] made a step further by describing look-ahead dispatching algorithms for overhead hoist transportation systems and demonstrating the potential of predictive dispatching to optimize semiconductor fab efficiency. Recent studies also explore reinforcement learning–based scheduling methods that leverage historical operational data to improve adaptability in highly dynamic semiconductor production environments [15].

In the field of warehouse logistics, [16] conducted an in-depth survey of AGV control strategies, emphasizing the importance of effective routing and dispatching in warehouse operations. [17] proposed an approach that integrates task assignment and path planning for multi-type AGVs, greatly improving efficiency in complex, multi-vehicle interactions. In a dynamic environment, [18] proposed an adaptive large neighborhood search combined with the Kuhn–Munkres algorithm to address dynamic and non-uniform task assignments. This approach ensures timely responses to unexpected events, significantly enhancing real-time operational performance. Additionally, [19] developed a model to predict scheduling for various tasks, making dynamic warehouse task management a crucial feature. Predictive scheduling may enable warehouses to respond flexibly to evolving demands while maintaining high productivity levels.

Recent algorithmic advancements have increased the efficiency of AGV dispatching. [20] used reinforcement learning-based hyper-heuristics to optimize real-time task allocation, improving adaptability and robustness in dynamic warehouse environments. Reinforcement learning facilitates continuous improvement through experience-based adjustments. [21] examined decentralized task allocation, demonstrating its potential to reduce congestion and increase AGV responsiveness through decentralized dispatch strategies. Decentralized dispatching promotes scalability and reduces dependency on central control, thereby enhancing the resilience of logistics systems.

Summing up, recent research has emphasized the importance of predictive dispatching, decentralized scheduling, and advanced optimization techniques to improve the productivity of AGVs in semiconductor material flow and warehousing. However, there is a gap in the literature regarding the connection between transportation logistics and production-phase timing, particularly in situations where task urgency changes in real time. This paper bridges that gap by offering logistical task scheduling strategies that consider both the location of the materials and the time by which they are needed by the production system. The solution optimizes end-to-end flow coordination and looks beyond upcoming bottlenecks when connecting transport decisions to production-phase timing. Scalable heuristics are also used, including so-called bucketing, to enable low-latency, responsive scheduling without relying on computationally heavy predictive models or detailed system forecasts. Previous research works ([22, 23]) not only motivated the current paper, but also established a solid background knowledge on the semiconductor use case.

3. Problem statement

This section outlines the problem of scheduling AGV-based indoor logistics in a semiconductor back-end manufacturing environment. The production context, system assumptions, and operational constraints are described that influence real-time logistical decisions. Together, these elements define the optimization problem addressed in the rest of the paper.

3.1. Semiconductor wafer fabrication

The context of the work reported here is defined by semiconductor manufacturing which is a complex, multi-stage process. The flowchart in Figure 1 shows the two stages of semiconductor wafer manufacturing process, from a bare wafer to a finished, tested, and packaged chip. For the detailed phases of front-end stage refer to [24].

Manufacturing stage produces the processed wafer out of the bare silicon wafer via many phases repeated until the required quality is achieved. The phases of this stage are carried out in clean rooms where the internal logistics is supported by an overhead hoist transport (OHT) [25]. A survey of dispatching rules for operational control in this stage of wafer fabrication is presented in [11], and a rigorous digital twin calibration approach for such automated material handling system is proposed in [10]. Long rest periods are required between the phases, often extending the whole process to over 100 days, making the transport time negligible compared to the total lead time. Therefore, the focus of this study is on the subsequent assembly and

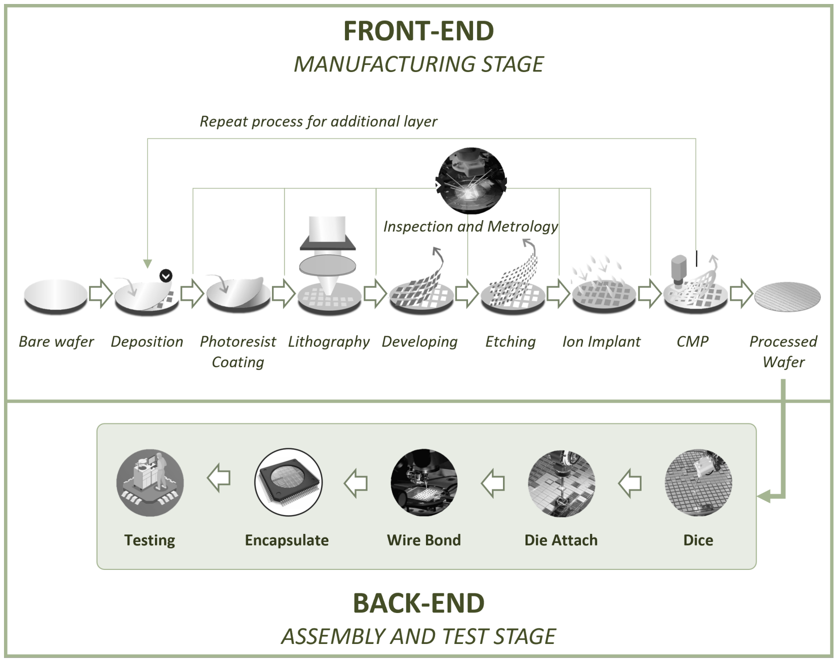


Figure 1: Semiconductor wafer production phases

test stage of the wafer manufacturing phases, where the optimization of internal logistical jobs plays a critical role in achieving efficient throughput levels.

In the assembly and testing stage, the fully processed wafer is first precisely cut into individual dies, or semiconductor chips. All dies are accurately attached to a substrate or packaging material in the die attach operation. Next, the die's electrical contacts are made by wire bonding, which involves attaching fine wires between the die contact and the package. Next, encapsulation encloses the chips in protective packages to ensure reliability and protect them from environmental conditions. Finally, the packaged chips undergo rigorous testing to confirm functionality, performance, and quality standards to ensure that fully functional components reach customers.

Efficient scheduling during assembly and testing is essential to maximize equipment utilization and reduce lead times. In these areas, processing and transportation times are comparable, so machine utilization depends heavily on the efficiency of indoor logistics. In this environment, AGVs are a key component in moving materials from origins to destinations. Without efficient scheduling of material handling, transport delays can undermine the optimized flow, resulting in equipment idle time, lost throughput, and scrap rate. These significant differences between OHT- and AGV-served transportation systems and the lack of literature about back-end stage transportation motivate the current research, and the following sections all aim to describe the behavior of such systems and propose novel methods for logistical task scheduling.

3.2. Relentless production

The fabrication and test phases outlined above require a constant flow of material to maintain maximum throughput, in contrast to the comparatively fixed nature of wafer fabrication. Back-end operations require high-frequency, dynamic internal logistical decisions. This transition requires an adaptive logistics system that fits to unforeseen events on an event-by-event basis, as outlined in the following relentless production paradigm.

The relentless production system is a stochastic production system, where operation is based on two main assumptions: (i) machines are almost always able to function, and (ii) raw parts are always available to be distributed to the machines. The stochastic property comes from real-world imperfections such as sudden machine breakdowns, varying loading and unloading times, or unexpected traffic jams. There is no fixed production schedule in this system, but the machines are constantly generating new logistical jobs. When a machine finishes a process, it immediately sends a request to the logistics center to remove the finished part. Similarly, when a machine is ready to start a new process (which is possible if a raw part is already at the machine's loading area), it immediately sends a job request to bring the next raw part to the station. In this way, the list of open logistical jobs is constantly updated by the events of the machines, and the scheduler's duty is to make the most effective assignment between these jobs and the AGVs.

A relentless production system is a stochastic, event-driven manufacturing system with the following characteristics:

1. Although there might exist different product types in the system, their process plans are identical. The difference between product types comes from the variety of raw parts.
2. There is no predefined job schedule, time-indexed production plan, or strict deadlines. Instead, jobs are generated dynamically in an event-driven way.

3. The main objective of the entire system (including machines and internal logistics) is to maximize throughput.
4. Resource availability is almost continuous and only occasionally interrupted by stochastic failure events.
5. Logistical jobs are generated exclusively by machine state-change events.

In summary, relentless production is a stochastic, event-driven manufacturing paradigm in which machines operate continuously without a fixed schedule, constantly generating transport requests as soon as they become ready, thereby requiring real-time, reactive logistics coordination to sustain throughput. Based on recent literature, the back-end stage (assembly and testing) of semiconductor manufacturing fits to the above list of criteria and the definition of relentless production [26].

3.3. Assumptions of production system

The layout is assumed to be given, i.e., the number of machines is fixed, along with information about station locations, processing times, process phases (e.g. dice, die attach, etc.), path locations, and their directions (one-way or two-way). The stations are distinguished by their function into three main modes: sources, drains and machines. All stations are equipped with two connection areas: input port (IP) at the entrance, and output port (OP) at the exit. The port capacity is equal to the number of parts the machine needs to start its process; in the current case it is 1. Any machine can start the process if and only IP is full and the machine is empty, and can finish and complete its process if and only if OP is empty. AGVs place parts on IP (loading) and pick up parts from OP (unloading). Both loading and unloading times are stochastic values due to the imprecision of the AGV's placement process. Sources and drains act as the entrance to the warehouse, raw parts come from and go to these stations, and together they are called IO (in-and-out) stations. All sources are assumed to have infinite capacity, i.e., raw parts are always available; they act as buffers of unlimited capacity between wafer processing and the assembly area of semiconductor manufacturing. All drains are also assumed to have infinite capacity, representing the input of a warehouse. Machine stations have one processing unit, and the type of machine is determined by the process phase it performs. The one-machine-one-phase rule applies, i.e., a machine cannot switch between process phases over time, its type remains unchanged. Machine availability is assumed to be represented by the percentage of working time compared to the total time. Machine failures occur stochastically, the repair time depends on the given MTTR (Mean Time To Repair) value. As usual, MTBF (Mean Time Between Failures) follows negative exponential distribution, and MTTR follows Erlang distribution [27].

In the back-end stage of semiconductor manufacturing, factories are primarily organized around a series of dedicated, single-function process steps, such as die attach, bonding, molding, and testing. Academic and industrial models typically reflect this setup, focusing on linear assembly lines rather than highly flexible, multifunctional stations. While some multifunctional integration exists, it is usually limited and tailored to specific processes. A prominent example is test handlers, which integrate device handling, temperature control, sorting, and electrical testing but do not switch between fundamentally different types of assembly tasks. Even when assembly and testing are integrated for certain package types, these are rare exceptions. Overall, modular specialization remains the most dominant and practical approach for modeling back-end manufacturing logistics.[28, 29, 30])

Although the assumption of the sources' infinite capacity may seem too idealistic, in this scenario, it reflects the functional scope of the manufacturing (front-end) stage being modeled rather than suggesting that the entire factory operates without limits. The assembly and testing (back-end) stage in question runs continuously and is not driven by individual orders. Therefore, it relies on a steady supply of raw parts from upstream processes. If those materials are not available, then the prior stage stops operating. This is beyond the scope and control of the system being analyzed. Similarly, drains refer to the transfer of materials to a downstream warehouse or logistics system whose capacity constraints are not examined in this study. From a control standpoint, periods when input material is missing can be managed with IO-machine jobs that remain open but inactive. These jobs are suspended until materials arrive, at which point AGVs can resume transporting them. Thus, assuming infinite capacity does not ignore material availability; it merely shifts the concern to the edges of the system. This allows the focus to remain on internal flows, machine usage, and AGV dispatching without distractions from upstream or downstream planning complexities.

The size of the AGV fleet and the speed of the AGVs are also fixed for a scenario. AGVs can transport more than one part at a time, up to their capacity. AGVs follow the collect-all-distribute-all rule: after the assignment, the AGV completes all the pick up tasks, and starts the delivery tasks after all of the previous ones have been accomplished. A general assumption is that all AGVs try to carry out as many logistical jobs as possible, meaning that if there are enough open jobs then they are assigned to an AGV up to its capacity.

The shopfloor layout consists of straight track segments arranged in a grid-like pattern with crossings and aisles. Parallel or diagonal aisle orientation is assumed, and each aisle has sufficient room for two parallel tracks to enable bidirectional AGV traffic. AGVs travel along discrete track segments, and overtaking is not possible. When an AGV stops to load or unload at a station on the side of an aisle, it occupies the corresponding track segment, which stops all following AGVs traveling in the same direction until the operation is finished. Stations can be placed on one or both sides of an aisle. Crossings are locations where AGVs can turn and perform U-turns. Crossings are considered exclusive-access areas: when an AGV enters a crossing, it occupies the intersection until it leaves, having completed transit. AGVs arriving while a crossing is occupied must queue in first-in, first-out order. AGVs queue along the track when obstructed by another vehicle or a congested crossing. A minimum safety distance exists between AGVs, and collision avoidance can be enforced using external control systems or onboard sensors.

A snapshot of the five-phased layout used in the simulation model as a use-case is presented in Figure 2. The ”S”- and ”D”-labeled stations at the top and bottom rows are the IO stations, the middle ones are the machines colored by their production phase.

3.4. Model of indoor logistics

In a relentless production system, where the layout is more complex than usual and the logistical tasks are continuously generated, AGVs can be the best servers to carry out the jobs. Each job consists of a sequence of two tasks: the first is to pick up a part from the origin station, and the second is to place it at the destination station. Naturally, jobs can be separated based on the type of their origin and destination. A job is called an IO-Machine or a bringing job, if its origin is an IO station, and its destination is a machine station. Machine-IO or taking jobs are defined exactly the other way around: their origin is a machine, and their destination is an IO. Machine-Machine logistical jobs are not included in this work due to the characteristics of the use-case which is the assembly and testing phase of semiconductor wafer manufacturing where in-between-phase storage locations are commonly used.

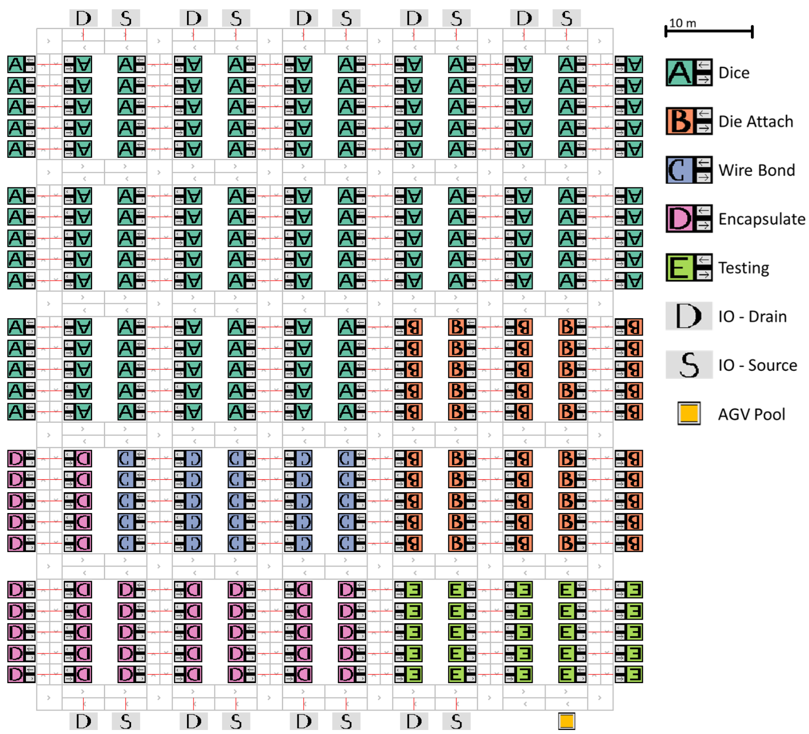


Figure 2: Layout snapshot of the simulation model of the use-case. The track width is 1.5 m and the total width of the aisles is 3 m. The machines and IO stations have uniform dimensions: 2.5 m × 5 m. The entire shop floor is rectangular with an area of over 6,000 m^2 (85.5 m × 78 m). IO stations are located in the top and bottom rows. There is an AGV area in the bottom right corner that can be used as a parking and/or charging station for the fleet.

As stated earlier, the logistical jobs are generated by the machines. Machines have four states: (1) processing when the machine actively works on a part, (2) waiting when IP is empty and the machine waits for a new raw part, (3) blocked when OP is full and the machine cannot place its finished part onto OP, and (4) failed when some unexpected failure event has stopped its process. Jobs are generated during certain state-change events. Figure 3 summarizes all feasible transitions from one machine state to another. Each change event is characterized by three properties: the trigger, the condition, and the generated job type.

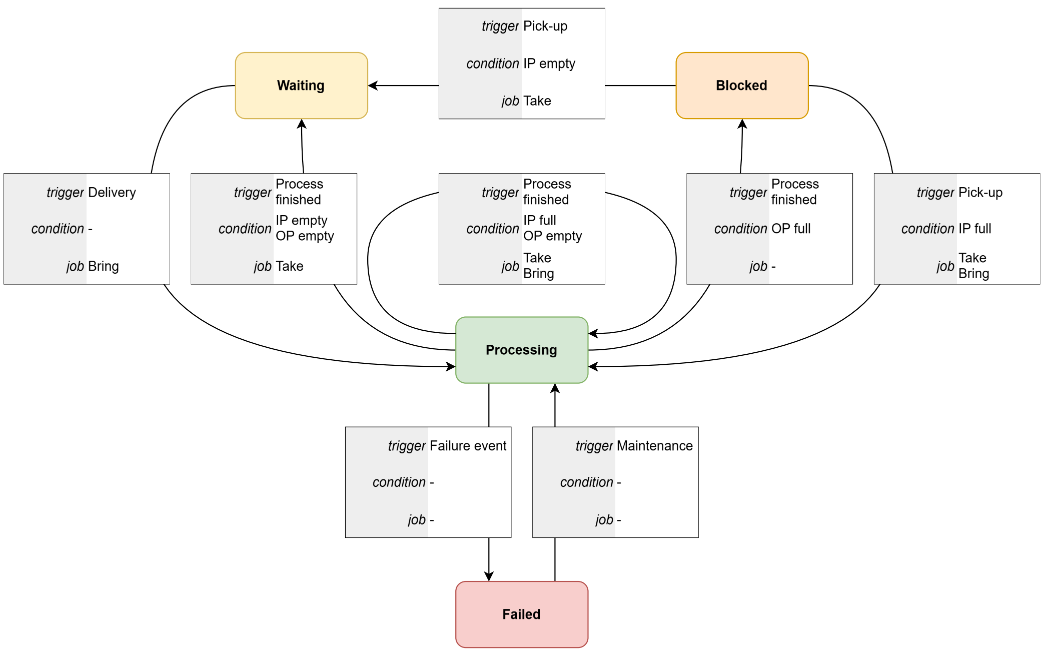


Figure 3: Machine state-change events. Each feasible change is described by its trigger, the environmental conditions, and the generated job's type.

Although this framework describes single-part logistical jobs, in real life, machines may require multiple parts. Fortunately, the definition can be extended easily: if n parts are required by a single machine then n different logistical jobs are generated and assigned. In this case, machines that already access some delivered parts should have priority. However, if the required multiple parts can be transported in sets, then the presented methodology is applicable as it is. There is a natural upper limit for the number of the open jobs at a time, which is twice the sum of required parts of all machines. It is reached only if all AGVs stop working, and all the machines have requested both bringing and taking jobs. Of course, this extreme and undesired event is not realistic to happen in a well-balanced production environment. The definitions above entail that there is an injection (many-to-one correspondence) between jobs and machines, which property is used later for job grouping (Section 4.6).

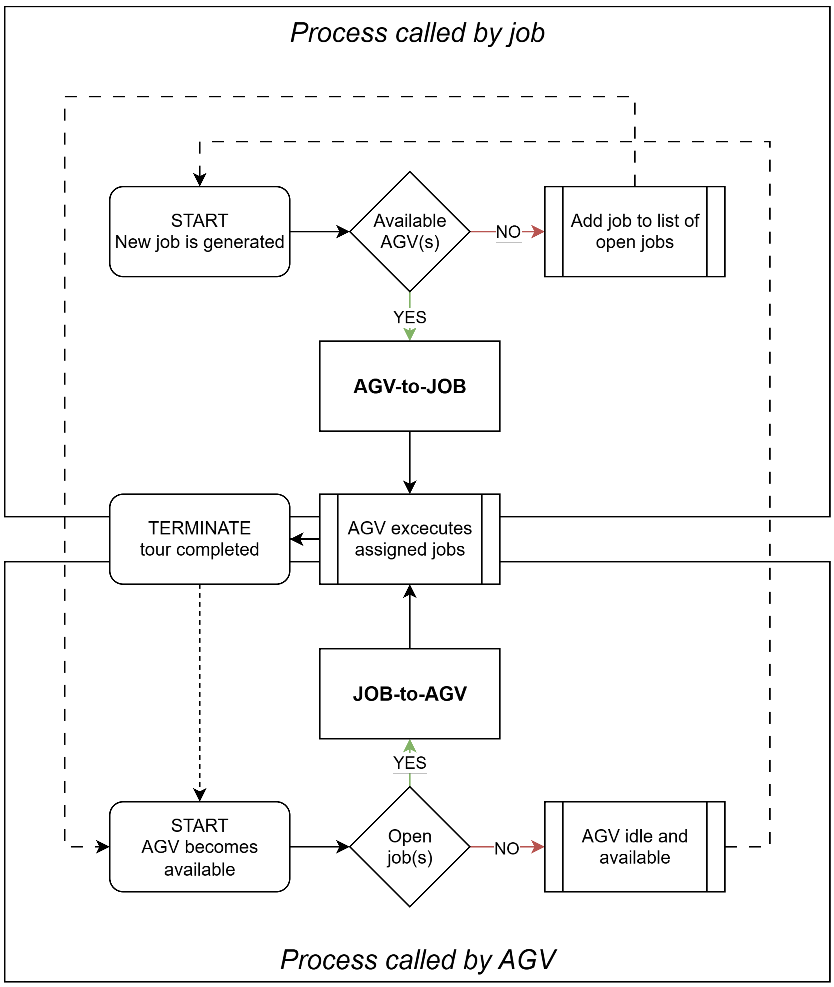


Figure 4: Model of internal logistics flow

The internal logistics flow shown in Figure 4 presents real-time logistical job scheduling in a closed-loop material handling system. The flow presents the scheduling process as a whole consisting of two separate process loops.

The first loop is activated by logistical jobs, which are continuously generated and trigger the central scheduler, who evaluates order factors such as origin and destination stations. After each and every job generation (when a machine requires a new part or need finished part to be taken away), the scheduler tries to find the AVG that is available (has free capacity and entitled to perform the job) and minimizes the cost of assignment. If there are no available AGVs in system, then the job is put in the list of open jobs and the system waits for the next trigger event. In parallel, the scheduler observes all AGV statuses, such as location, availability, load, and activity. The second process is activated by the AGVs. When any AGV becomes available (meaning that it has just executed all previously assigned jobs), the scheduler browses among the open jobs and selects the set that minimizes the assignment cost. The aim is to fully load the AGV and when it happens, the AGV becomes unavailable. However, if there are less open jobs than the capacity, the AGV starts to execute them while staying available until it reaches its full capacity. When the AGV has finished the last assigned pick up task it becomes unavailable (regardless of whether it reached its full capacity), and immediately starts to execute its delivery tasks. If there are no open jobs in the system at all, then the AGV remains available without any task to do. In such cases it can either wander around the shopfloor randomly, or go to a charging station. As assignments are issued, AGVs start moving the materials and live operational data is fed back to the scheduler. The two essential algorithms of the flow, AGV-to-JOB and JOB-to-AGV, are formulated and detailed in the next section.

Technically, this flow represents an already running system, but the absolute start can also be interpreted. At the very beginning, it is assumed that all AGVs are available, and when the factory starts operating, the machines immediately start generating bring-type jobs, then AGVs are assigned to them.

In such a system setup, where the serving AGVs have the capacity to transport more than one part at a time, the dispatcher's rule is not only to find feasible AGV-job pairs, but also to determine the order of task completion. In other words, the scheduler solves the scheduling problem: the allocation between AGVs and logistical jobs, and the sequencing of the tasks of the jobs to be executed. Traditional scheduling strategies include the well-known first-in-first-out, shortest or longest processing time, and earliest due date prioritization rules. Planning is not reasonable due to the system's continuous, event-driven job generation and the lack of a fixed production schedule. This requires real-time responsiveness: reactive dispatching replaces static planning, as the latter would quickly become obsolete because of equipment failures or transport delays. Look-ahead is not used because future jobs in this stochastic, event-driven environment are unpredictable, where stochasticity is caused by machine breakdowns, varying placement times, and traffic situations. This leaves only current state data as the basis of decision-making. Prioritizing real-time dispatching prevents the unnecessary computational cost of ineffective anticipatory planning.

3.5. Key performance indicators

The key performance indicator (KPI) used throughout this work is throughput, defined as the minimum amount of finished goods produced per production phase in a given time interval. This criteria maximizes the number of completed products in a given time period. This measure defines the critical throughput bottleneck of a system and quantitatively reflects the rate-limiting phase of the overall production flow. By maximizing the throughput, the scheduling strategy prevents the slowest phase from affecting the performance of the entire system, thereby increasing overall production efficiency.

Secondary metrics are always useful in such investigations to support and test the found outcomes. After analyzing the main KPI, two other point of views are considered: the total traveled distance by AGVs, and the non-value-added-time (NVAT) of the machines. The fist one is measured in meters and proportional to the energy consumed by the AGVs. NVAT of a machine is defined as the portion of time when the machine is either waiting for a new part to arrive or blocked by a finished part because the previous has not been taken yet. In such cases, the machine could theoretically work, but stops due to the deficiency of the transportation system. Machine failures are not included in the NVAT value. These metrics supplement the examination in order to reach better understanding of the system's behavior.

4. Methodology

In this section, a general scheduling framework is proposed, which can be applied with any kind of job prioritization rule, and applies the flow model in Figure 4. Later, dedicated dispatching strategies, such as distance minimization and balancing of machine utilization and production phases, are proposed and described. An acceleration technique is also presented to tackle the increasing computational complexity of logistical task sequencing.

Then, the combination of such strategies is defined with a discussion on the weights of their combination. Lastly, the concept of bucketing is introduced, which is a type of disjoint grouping of logistical jobs. Table 1 presents all notations used in this section.

Symbol	Type	Meaning
S	set	set of stations
s	object	single station
$d_{s_1 \to s_2}$	matrix	distance between stations
δ	float	mean distance between machines and IO stations
M	set	set of machines
m	object	single machine
P	set	set of process phases
p	object	single phase
p_m	string	process phase of machine (A, . . . , E)
t_p	time	cycle time of process phase p
k_p	integer	count of machines serving process phase p
K_p	integer	count of finished parts from process phase p
A	set	set of AGVs
a	object	single AGV
C	integer	AGV capacity (uniform)
v	float	AGV speed (uniform)
L_a	integer	current load of an AGV (always $\leq C$)
$o_\uparrow$	list	order of picking up assigned jobs
$o_\downarrow$	list	order of delivering assigned jobs
$\mathcal{P}(n)$	group	the set of all permutations of $\{1, 2, \ldots, n\}$
J	set	set of jobs
j	object	single job (includes pick up and delivery tasks)
m_j	object	machine of job
T_j	datetime	deadline of job
ϕ_j	string	state of job (new, ongoing, or done)
f	function	cost of assignment

Table 1: Table of notations

4.1. General dispatching framework

The overall model of scheduling mechanism consists of two main algorithms: the job selection is called JOB-to-AGV in Algorithm 1, and the AGV selection is called AGV-to-JOB in Algorithm 2. The objective of these algorithms is to optimally allocate logistical jobs to available AGVs in order to achieve maximum efficiency while minimizing transport costs. Both of them return not only the optimal assignment but also the best sequencing of the tasks of jobs that belong to a specific AGV.

Algorithm 1 JOB-to-AGV

input a AGV
output J_a^* list of assigned jobs
output $o_\uparrow^*$ pick up order
output $o_\downarrow^*$ delivery order

$J_a^* \doteq \emptyset$
while $L_a < C$ (a has free capacity) **do**
  $J_a \doteq \{j \in J | \phi_j = \text{"new"}\}$
  if $|J_a| > 0$ **then**
    $n \doteq |J_a^*| + 1$

$$j^*, o_\uparrow^*, o_\downarrow^* \doteq \mathop{\arg\min}_{\substack{j \in J_a \\ o_\uparrow \in \mathcal{P}(n) \\ o_\downarrow \in \mathcal{P}(n)}} f(a, J_a^* \cup \{j\}, o_\uparrow, o_\downarrow)$$

    $J_a^* \doteq J_a^* \cup \{j^*\}$
    $\phi_j \doteq \text{"ongoing"}$
  else
    if $|J_a^*| > 0$ **then**
      a executes already assigned jobs
    else
      a becomes idle and available
    end if
  end if
end while
return $J_a^*, o_\uparrow^*, o_\downarrow^*$

Algorithm 1 assigns jobs to an individual AGV by sequentially selecting and ordering jobs based on the AGV's available capacity and current operational constraints. The algorithm iteratively selects jobs from the list of new jobs (J_a) and assigns them according to an associated cost function f that weights possible task sequences based on logistical efficiency. Whenever the job j^* that minimizes f is found, it is added to the optimal set J^* which is the return value of the algorithm along with the optimal pick up and delivery orders.

Algorithm 2 AGV-to-JOB

input j job
output a^* assigned AGV
output $o^*_\uparrow$ pick up order
output $o^*_\downarrow$ delivery order

$A_j \doteq \{a \in A | L_a < C \text{ } (a \text{ has free capacity})\}$
if $|A_j| > 0$ **then**
  $J_a \doteq \{j \in J | j \text{ is already assigned to } a\}$
  $n_a \doteq |J_a| + 1$
  get AGV with minimal f cost:

$$a^*, o^*_\uparrow, o^*_\downarrow \doteq \underset{\substack{a \in A_j \\ o_\uparrow \in \mathcal{P}(n_a) \\ o_\downarrow \in \mathcal{P}(n_a)}}{\arg\min} f(a, J_a \cup \{j\}, o_\uparrow, o_\downarrow)$$

  $\phi_j \doteq$ "ongoing"
else
  $\phi_j \doteq$ "new"
end if
return $a^*, o^*_\uparrow, o^*_\downarrow$

Algorithm 2 complements the first one by assigning a specific job to the most suitable AGV among the eligible candidates. It compares the available AGVs (A_j), calculates the costs associated with assigning the job to each AGV in A_j, and selects the AGV with the minimum operating cost, thus optimizing the use of resources.

Note that due to task sequencing, the computational demand of both algorithms may be too high for AGVs with higher capacities, so an accelerated version is presented later.

4.2. Dispatching strategies

Dispatching strategies are described by logistical job prioritization rules, which effect not only the assignment between jobs and AGVs but also the sequencing of individual job tasks. Different strategies have been developed, each with a unique objective, to be optimized according to an appropriate cost function. In the followings, after the description of the baseline dispatching strategy come the novel strategies. Their main advantage comes from the recognition of not only the traveling distance but also the time and phase component that is constantly changing during operation. They bring a clean and smart solution on how to prioritize jobs that may have large transportation cost but are important and urgent from the production system's point of view.

The cost function of the baseline Distance Minimization (DM) approach, f_{DM}, as given by Equation (1), represents the total travel distance of an AGV a assigned to a given subset of logistic jobs J_a. The function considers both pick up and delivery tasks of the assignment, where $o_\uparrow$ and $o_\downarrow$ are permutations of tasks in their respective execution sequences. The domain of this function covers the AGV a, the set J^n of jobs, where $n \doteq L_a$ is the load of a, and all permutations of pick up and delivery sequences, expressed by the power set $\mathcal{P}(n)$. The cost is defined as the sum of the travel distances from one task to the next consecutive task during pick up, from the last pick up station to the first delivery station, and from consecutive delivery stations to the next one. The objective function captures the total route cost of completing a multi-job order and serves as the sequencing optimization objective within the model for the dispatcher's decision making.

$$\begin{aligned} &f_{DM} : \\ A \times J^n \times \mathcal{P}(n) \times \mathcal{P}(n) &\rightarrow \mathbb{R} \\ (a, J_a, o_\uparrow, o\downarrow) &\mapsto \sum_{i=2}^{n} \left(d_{J_a\left[o_\uparrow(i-1)\right] \longrightarrow J_a\left[o_\uparrow(i)\right]} \right) + \\ &+ d_{J_a\left[o_\uparrow(n)\right] \longrightarrow J_a\left[o_\downarrow(1)\right]} + \\ &+ \sum_{i=2}^{n} \left(d_{J_a\left[o_\downarrow(i-1)\right] \longrightarrow J_a\left[o_\downarrow(i)\right]} \right) \end{aligned} \tag{1}$$

The cost function of the Equalize Phase (EPh) approach f_{EPh}, as introduced in Equation (2), measures the imbalance in throughput across phases of a given AGV assignment. The function acts on an AGV a, a subset of jobs assigned to that AGV, i.e., set of jobs J_a, along with their pick up $o_\uparrow$ and delivery $o_\downarrow$ sequences. Each job j has a manufacturing phase p_{m_j}. For each phase $p \in P$, K_p, refers to the number of parts completed in that phase. The formula calculates the variance across all phases, i.e., across all $p \in P$, in the estimated total finished parts by phase, i.e., in $K_p + \sum_{\substack{j \in J_a \\ p_{m_j} = p}} 1$. This variance captures how a given current assignment would affect the phase-level throughput balance if it were implemented. The expression is scaled by a factor δ, where δ refers specifically to the mean machine-to-IO station distance, effectively anchoring load balancing optimization to the physical organization of the system. The inclusion of δ is a technicality for later combined models. This cost function is useful for solving production equity by evenly distributing completed tasks across production phases. This approach reduces the variance between phases and distributes the production workload evenly.

$$
\begin{aligned}
f_{EPh} :& \\
A \times J^n \times \mathcal{P}(n) \times \mathcal{P}(n) \to& \mathbb{R} \\
(a, J_a, o_\uparrow, o\downarrow) \mapsto& \delta \cdot \mathrm{Var}_{p \in P} \left[K_p + \sum_{\substack{j \in J_a \\ p_{m_j} = p}} 1 \right]
\end{aligned}
\tag{2}
$$

Although Equation (2) is invariant to pick up and delivery orders ($o_\uparrow$ and $o\downarrow$), the formulation is recommended due to its general form in Algorithms 1 and 2.

The cost function of Equalize Machine (EM) approach, f_{EM}, given by Equation (3), estimates the time-based execution cost of an AGV a performing a set of logistic jobs J_a. It takes as input the AGV, the subset of jobs of size n, and ordered pick up and delivery sequences $o_\uparrow$ and $o_\downarrow$. It estimates the cumulative job execution time by aggregating time slices $T_j - T_0$ across jobs $j \in J_a$, where T_j refers to each individual job's deadline and T_0 refers to a reference start time (e.g., the time of the decision or the start of production). The deadline of a job is set by its machine's cycle time at the time when the job is generated. The whole expression is scaled by AGV speed v, This is again a technical convenience, just as the inclusion of δ in f_{EPh}. It accounts for the latency or delay penalty of performing multiple jobs at once and proves useful in applications where job lead time minimization is important. In multi-objective optimization contexts, f_{EM} serves as a complement to distance-based cost functions (e.g., f_{DM}) by focusing exclusively on temporal efficiency over spatial aspects, and is also an indirect way of balancing machine utilization in the factory.

$$\begin{aligned} f_{EM}:& \\ A \times J^n \times \mathcal{P}(n) \times \mathcal{P}(n) \to& \mathbb{R} \\ (a, J_a, o_\uparrow, o\downarrow) \mapsto& v \cdot \sum_{j \in J_a} (T_j - T_0) \end{aligned} \tag{3}$$

Equation (3) is, similarly to EPh cost function, invariant of pick up and delivery orders ($o_\uparrow$ and $o\downarrow$).

4.3. Acceleration of DM sequencing

As discussed in Section 3.4, every logistical job contains one pick up and one delivery task, and the complete operating cycle of the AGV consists of two distinct phases: first, it performs all pick up tasks, then, all delivery tasks.

Algorithms JOB-to-AGV and AGV-to-JOB plan a complete operating cycle iteratively, inserting the two tasks of a new job into the earlier planned route at a time, while keeping the task sequence for the earlier jobs unchanged. During this, the DM cost function determines the optimal locations for insertion and calculates the cost of the extended route.

However, explicit enumeration is computationally prohibitive at this case of practical relevance with several hundreds of machines. The naive implementation of DM considers every possible combination of the pick up insertion position and the delivery insertion position, corresponding to n^2 candidate sequences when inserting the nth task into the earlier route of length $(n-1)$.

An accelerated implementation stems from the observation that the two optimal insertion positions are independent of each other, unless the new pick up task is inserted last or the new delivery task is inserted first into the route. Hence, it is sufficient to consider the following candidate routes:

- The $(n-1)$ candidate not-last-pick up and $(n-1)$ candidate not-first-delivery sequences can be built independently, and the two optimal sub-sequences can be merged;

- All the n candidate pick up sequences can be coupled with a delivery sequence starting with the new delivery task;
- Finally, all the n candidate delivery sequences can be coupled with a pick up sequence ending with the new pick up task.

Since the cost of a new sequence after insertion can be calculated in constant time, the above method leads to an $O(n)$ implementation of the DM function instead of the naive $O(n^2)$ implementation.

4.4. Combined cost function

To achieve multiple strategic objectives, combined cost functions are formulated that merge distance minimization with equity-based approaches. EPh* (Equation 4) integrates distance minimization with equalized phase servicing, attempting to find an optimum between efficient AGV movements and stabilized production. The value of parameter w controls the trade-off between these conflicting objectives. EM* (Equation 5) balances minimizing AGV travel distances with evenly distributed machine load.

$$\begin{aligned} &f_{EPh^*}: \\ A \times J^n \times \mathcal{P}(n) \times \mathcal{P}(n) &\rightarrow \mathbb{R} \\ (a, J_a, o_\uparrow, o_\downarrow) &\mapsto w \cdot f_{DM}(a, J_a, o_\uparrow, o_\downarrow) + \\ &\quad + (1-w) \cdot f_{EPh}(a, J_a, o_\uparrow, o_\downarrow) \end{aligned} \tag{4}$$

$$\begin{aligned} &f_{EM^*}: \\ A \times J^n \times \mathcal{P}(n) \times \mathcal{P}(n) &\rightarrow \mathbb{R} \\ (a, J_a, o_\uparrow, o_\downarrow) &\mapsto w \cdot f_{DM}(a, J_a, o_\uparrow, o_\downarrow) + \\ &\quad + (1-w) \cdot f_{EM}(a, J_a, o_\uparrow, o_\downarrow) \end{aligned} \tag{5}$$

The choice and adjustment of the parameter w is discussed in more detail in the following experimental analysis.

4.5. Best weights of combined cost functions

In Equations (4) and (5), the weight of the DM component w is assumed to be a given parameter, but it should be refined for each fab scenario. The return value of both cost functions is highly dependent on this parameter, as it weighs the importance of DM over EM or EPh prioritization rules. Traditionally, the optimal value of w can be estimated by a grid search, where the cost function is evaluated for a set of w values and the one that minimizes the cost is selected. To find the optimal w value of the combined cost functions, simulation experiments can be evaluated, but performing a full grid search for each possible scenario is computationally expensive. Therefore, to reduce the evaluation cost, golden section search [31] is used, as it is an efficient optimization algorithm for unimodal functions over bounded intervals. This method uses the golden ratio to identify two interior points within the current interval. At each step, the cost function is evaluated at these points, and the interval is updated in a way that retains the subinterval likely to contain the optimum. One of the previously evaluated points is reused in the next iteration, minimizing the number of function evaluations. This process continues until the interval length falls below a defined threshold, and the w value that minimizes the cost function in the last iteration is selected as the optimal combined cost function weight.

Figure 5 shows the result of a series of experiments to compare grid search and golden section search. The grid is defined as fifteen equally spaced values between 0 and 1, and the bounds. Examining the results, it seems like that there is only one global peak at $w \approx 0.95$, as the metric has a single peak. Continuity and smoothness are guaranteed by the shape of the boxes in the plot as the parameter varies. By definition, the weight is bounded between 0 and 1, and the boxes show only a little noise in the experiments. Meeting these criteria ensures that the golden section search approach will reliably guide the system to an optimal parameter value. The horizontal lines (I_i) on the plot mark the selected region (where the maximum lies) after the nth iteration of the golden section search. The numbers around the boxes indicate the iteration, where only in the first iteration must be two values evaluated (f_1^A and f_1^B), in other iterations evaluation is at only one place (f_j). The effectiveness of the golden search method is evident: after only five iterations, or evaluations at six different weights, the suspected optimum is found. The interval in which the optimum is known to exist is less than 6% of the original interval defined by the bounds. In comparison, a grid search requires seventeen evaluation points, and the length of the same interval is 12.5% of the entire region.

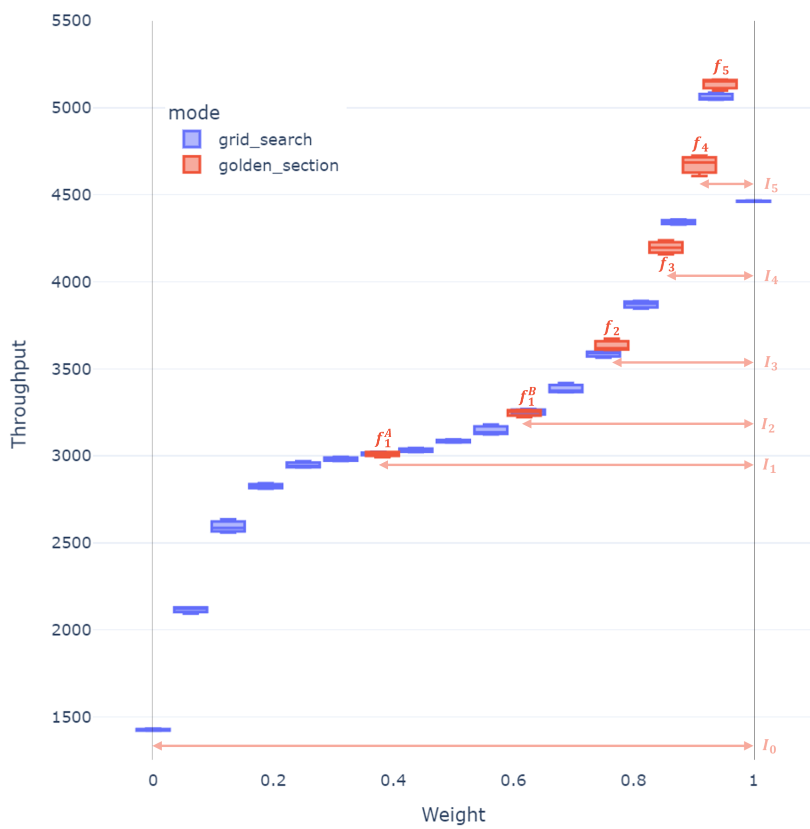


Figure 5: Grid search (blue) and golden section search (red) for optimal weight estimation. Numbering of the boxes indicates the golden search iteration, and the horizontal double-sided arrows show the selected region after the iteration.

4.6. Bucketing

The last proposed strategy, called bucketing, clusters physically close machines of the same production phase into buckets of size matching the AGV capacity, with one vehicle completing all jobs in a bucket in one operation. Although this grouping is effective on machines, it also acts as a bucketing of logistical jobs with a minor modification. Each machine bucket creates two job buckets, the first one being the IO-Machine jobs, and the second one being the Machine-IO jobs. During operation, whole job buckets are assigned to AGVs (and the other way around) instead of individual logistical jobs.

The goal of bucketing is to define groups of machines that are close to each other and operate synchronously. This strategy focuses on reducing travel time, streamlining scheduling, and minimizing AGV idle time. Buckets, involving groupings of jobs, can make the scheduling process simpler and potentially reduce computational complexity. AGVs tend to operate continuously because they can perform multiple tasks per trip, minimizing downtime between jobs. However, implementing such a bucketing scheme requires careful attention to potential problems, which can include reduced flexibility, complications in prioritizing jobs, and complexity in determining the ideal machine buckets.

The shopfloor (of a single machine type) is represented as a grid-like two-dimensional structure in which the placement of the machines is discrete, and there are tracks that restrict movement between them. In order to build contiguous buckets of machines, a snake-like bucketing order is proposed to ensure that each group consists of exactly C machines in spatial continuity, with C representing the capacity of a single AGV in the fleet. The bucketing starts in the top-left corner, then extends in a selected primary direction – either to the right along rows or downwards along columns – until there is a full bucket of C machines. When a barrier, e.g., a track or grid boundary, blocks further movement in the direction of travel, the bucketing order adjusts by turning to the next viable direction, in a zigzag pattern. This is repeated for all production phases separately ensuring that the machines of each bucket are of one type. Figure 6 shows a small example for the execution of the proposed method.

This strategy ensures that the machines are allocated in a continuous, orderly fashion, maximizing the adjacency of each bucket and responding to traffic constraints. The algorithm repeatedly processes all the machines in turn until the entire factory floor is divided into approximately equal-sized buckets. The resulting structure has logical order and spatial consistency,

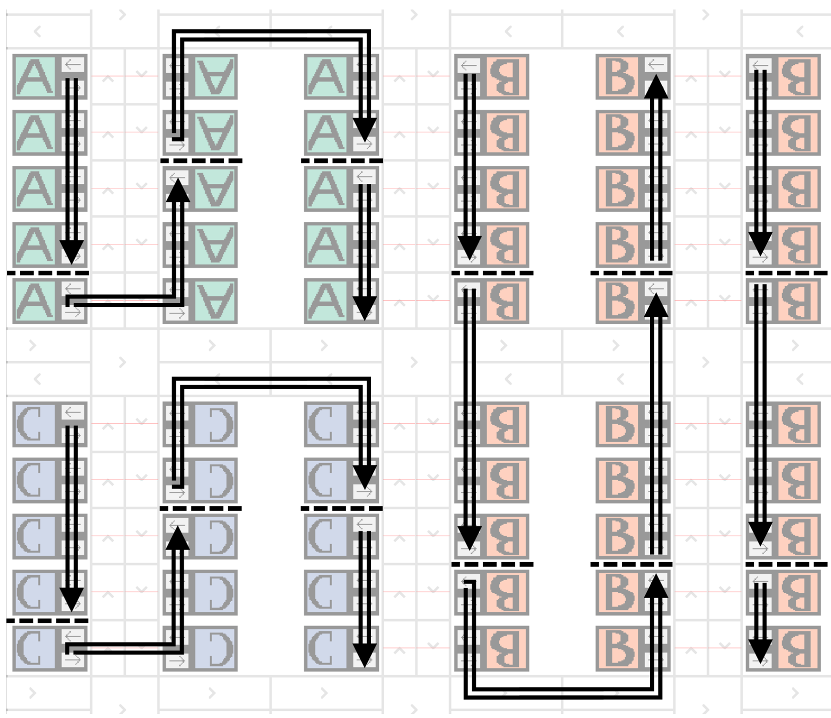


Figure 6: Snake-like bucketing order. Machines are processed in the predefined row-wise zigzag shape (left-to-right, then right-to-left) in all phase groups, and consecutive machines are grouped into buckets of size equal to the AGV capacity which is set to 4 in this example. Dashed lines indicate bucket borders.

making it well suited to support applications that require localized machine allocation.

5. Experiments and results

The following section presents a comprehensive simulation study to evaluate the performance of the proposed AGV dispatching strategies in various production scenarios. The experiments will evaluate the effects of system parameters such as capacity, the number of AGVs, and production structures.

5.1. Experiment setup

A simulation framework was developed to analyze and measure the performance of the proposed dispatching strategies. The plans of experiments were designed to replicate a wide range of realistic production scenarios in a semiconductor back-end plant[1]. Some of the most significant factors were randomized across situations to observe how the logistics configurations influenced the system's behavior.

One of the primary variables was the number of the production phases. The simulated scenarios included a setup with a single type of machine, reflecting a homogeneous setup, as well as setups with three or five different phases, reflecting various phases in the assembly process, including dicing, die attach, wire bonding, encapsulation, and testing. This enabled the study to test how production phases' complexity affects scheduling performance.

Additionally, the transport system was diversified by scaling the AGV capacity. Vehicles with various capacities, from small to large (5, 7, 10, 11, 13, 15), were tested to observe the interaction between transport batching and total throughput. At the same time, the size of the AGV fleet was tested under optimal conditions and with plus-or-minus twenty percent deviations to mimic over- and under-provisioned logistics systems. A simulation-based method was used to determine the most appropriate AGV fleet size by incrementally increasing the number of AGVs and measuring throughput until it plateaued, indicating system constraints. In these simulation runs the base dispatching strategy, DM without bucketing was applied. The smallest fleet

[1]The short presentation of the underlying industrial project which motivated the research can be found at `https://epicinnolabs.com/cases/digital-twining-examples-1/`. The simulation model used in this paper is an abstracted version of the developed digital twin of the real production.

size that achieved maximum throughput was marked to avoid resource over-provisioning, ensuring cost efficiency and optimal performance. However, from the perspective of current research, the interesting case is when the AGV fleet becomes a bottleneck. This occurs when the maximum throughput is not reached using the base strategy, but better performance can be achieved by applying smarter dispatching. Therefore, the final AGV fleet sizes have been fixed at 5, 7, and 10.

Another significant factor was the availability of the machines. The machines experienced stochastic breakdowns and had 80%, 80%, and 100% availability rates in various scenarios. The purpose of this part of the experiment was to assess the resilience of the dispatching strategies in realistic operating scenarios involving interruptions. Several factors remained unchanged throughout all the simulations. The factory floor layout was predetermined to consist of 300 machines arranged in a spatially consistent manner as shown in Figure 2. The machines were allocated across phases in inverse order of their respective average processing times to balance the workload and avoid phase-dependent bottlenecks. IO station allocations were based on a rule derived from the fab size to ensure proper connectivity for transport operations. The speed at which the AGV moved was preset to 0.5 meters per second to eliminate the impact of variations in transport speed on the dispatching logic. Although the AGV fleet could theoretically consist of various types and models of AGVs, common practice shows that single logistical systems are served by a fleet of uniform vehicles [32].

In the used simulation environment, the default AGV transporter follows a fixed path consisting of nodes and connectors. It begins at a designated home or parking spot and automatically departs when a set of logistical jobs and the orders of execution is assigned. The AGV moves at a constant speed, which is set as a parameter, and it stops automatically when arriving at loading or unloading areas. The AGV only changes direction at network nodes, aligning itself with the next path segment. Each AGV carries loads up to its capacity at a time by default and won't accept a new task until the current one is finished. Loading and unloading use typical station logic, which may include a stochastic processing time. When multiple AGVs use the same track, the system reserves track segments to prevent collisions; vehicles will wait if a segment is already in use. Unless extra control logic is added, the AGV doesn't change its route on the fly. Once its jobs are complete, the AGV waits for the next job or returns to its parking spot. This built-in behavior is rule-driven, and it is designed to simulate a straightforward yet realistic internal transport setup.

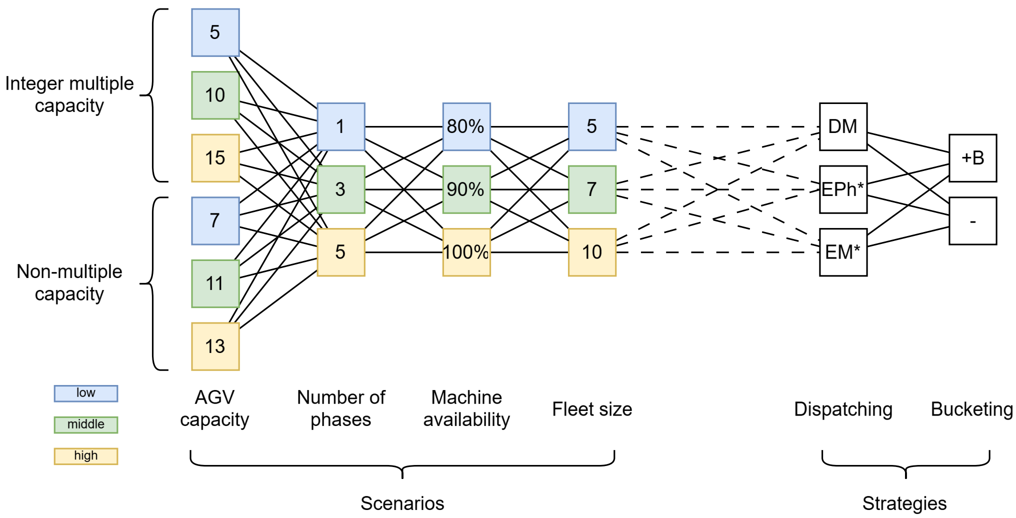


Figure 7: Network of experiment setups. The set of experimental scenarios is defined as the Cartesian product of all scenario variables (AGV capacity, number of phases, machine availability, and fleet size values) and strategy combinations.

The test setup offered a total of 162 scenarios, each characterized by a unique combination of fleet size, transport capacity, and machine setup (types and availability), summarized in Fig. 7. The series of experiments consisted of 10 repetitions for all scenarios and all dispatching strategies, and each run simulated 10 days of production. That resulted in a total of 9,720 runs. This systematic design enabled fair comparisons of the performance of different strategies under various adverse production conditions.

5.2. Overall results

Different experiment scenarios – even with the same algorithms – have different throughput values due to varying system capabilities. Therefore, relative throughput difference (RTD) has been introduced to make dispatching strategies comparable. In all scenarios, the baseline strategy is plain DM (without bucketing), and the throughput difference of the other strategies is normalized by the baseline strategy's throughput. Formally, the throughput is $K = \min_p K_p$, and if DM's throughput is K_{DM}, then the RTD of strategy f is defined as $\frac{K_f - K_{DM}}{K_{DM}} \cdot 100$. RTD shows the effect of different strategies as a percentage rather than an absolute value.

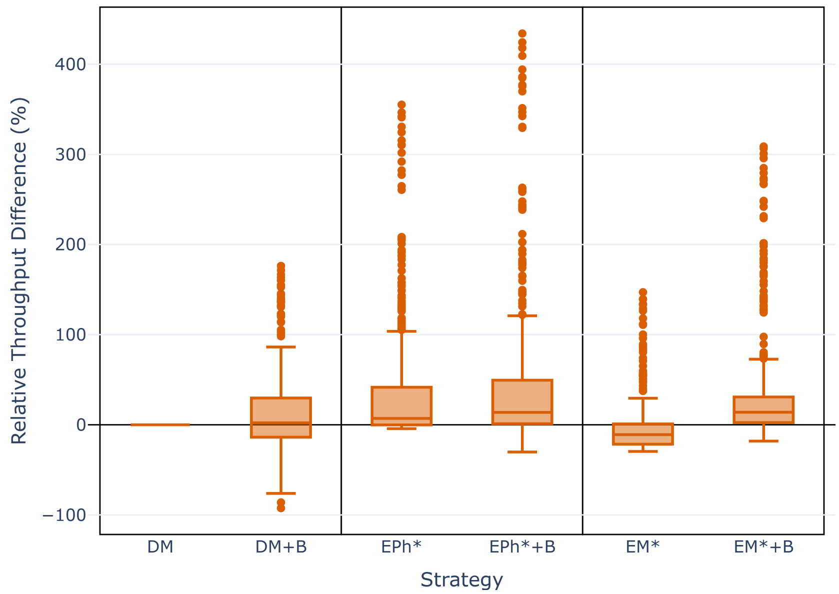
All the plots below show the results of simulation experiments and consist of two subplots. The first subplot always shows results when the AGV capacity is set as an integer multiple of the aisle size ($C \in \{5, 10, 15\}$). Meanwhile, the second subplot includes scenarios when the capacity is a non-multiple number ($C \in \{7, 11, 13\}$). The plots have a similar structure, separated into boxes based on dispatching strategies (DM, EPh*, and EM*), and each strategy is split into non-bucketed and bucketed (+B) versions.

Figure 8 illustrates the RTD for all experiments and dispatching strategies. Even at first glance, the difference between scenarios in which the AGV capacity is an integer multiple of the aisle size is clear. In the first case, bucketing has a positive effect for all three strategies. Summarizing all RTDs results in an overall average increase of throughput by 14.28% compared to the same strategy's non-bucketed version. Meanwhile, in the other set of scenarios, bucketing seemingly deteriorates the results. The EM* strategy has unpredictable effects in Figure 8b (throughput is below the baseline DM strategy in half of the cases), but the EPh* strategy improves throughput in nearly all experiment runs.

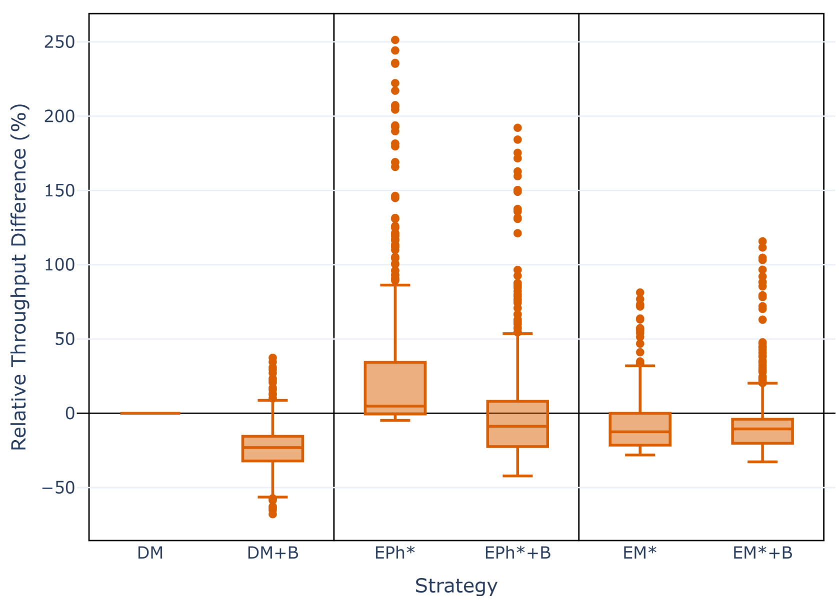
5.3. Throughput in different scenarios

The figures of this section present the effect of different dispatching strategies in system setups with different characteristics. The groups of the following charts are labeled as low, middle, and high, corresponding to the values that are discussed in Section 5.1. For better visibility and more intuitive understanding, the boxes of the previous chart are changed to bars, which always denote the mean of the RTDs.

As stated before, based on the heterogeneity of production, three different setups are evaluated: fabs operating one, three, or five phases. Analyzing the RTD values of Figure 9, a conclusion about the dispatching strategies and bucketing is noticeable: in case of single-phase fabs, bucketing improves throughput slightly when AGV capacity is an integer multiple of the aisle size. However, in other cases, the baseline DM strategy outperforms every other candidate apart from noise. When multiple phases are present at the fab, both of the alternative strategies, EPh* and EM* increase the throughput significantly. Again, in case of integer multiple capacities, bucketing tends to outperform its non-bucketed pair consistently, as visible from the uniformly positive bars across most configurations in Figure 9a. In contrast, Figure 9b highlights the challenges of non-multiple AGV capacities, where bucketing's benefits become inconsistent. In these cases, even the advanced

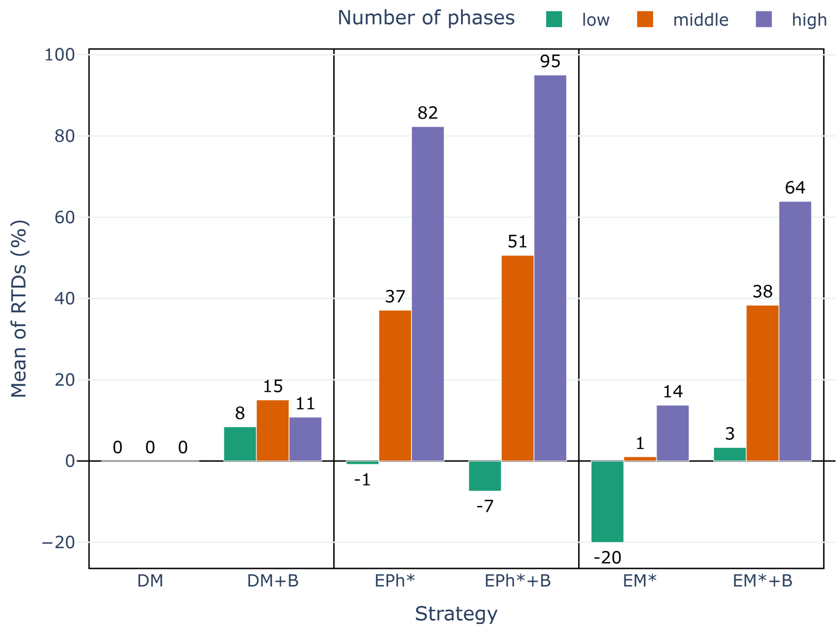


(a) Integer multiple capacity, $C \in \{5, 10, 15\}$

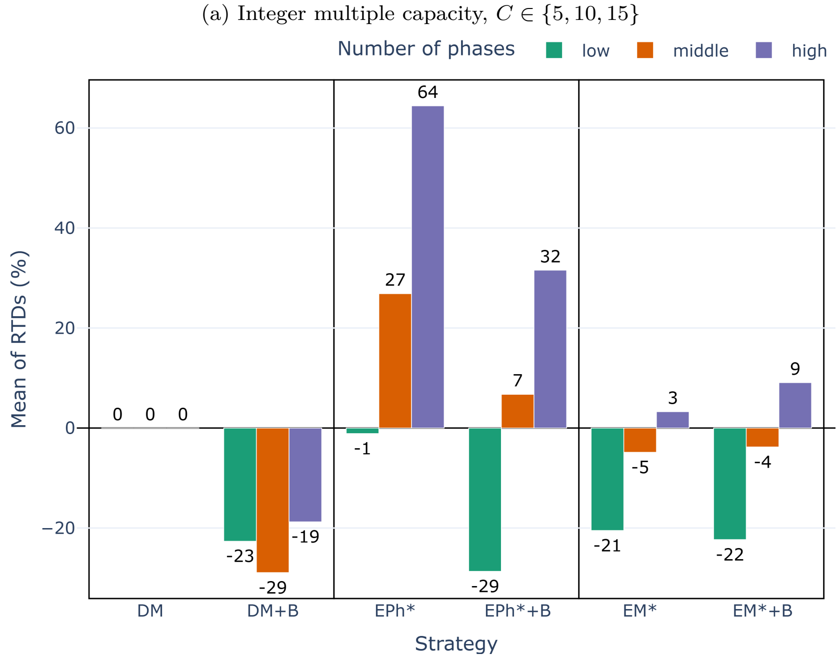


(b) Non-multiple capacity, $C \in \{7, 11, 13\}$

Figure 8: RTD of all experiment runs.



(a) Integer multiple capacity, $C \in \{5, 10, 15\}$



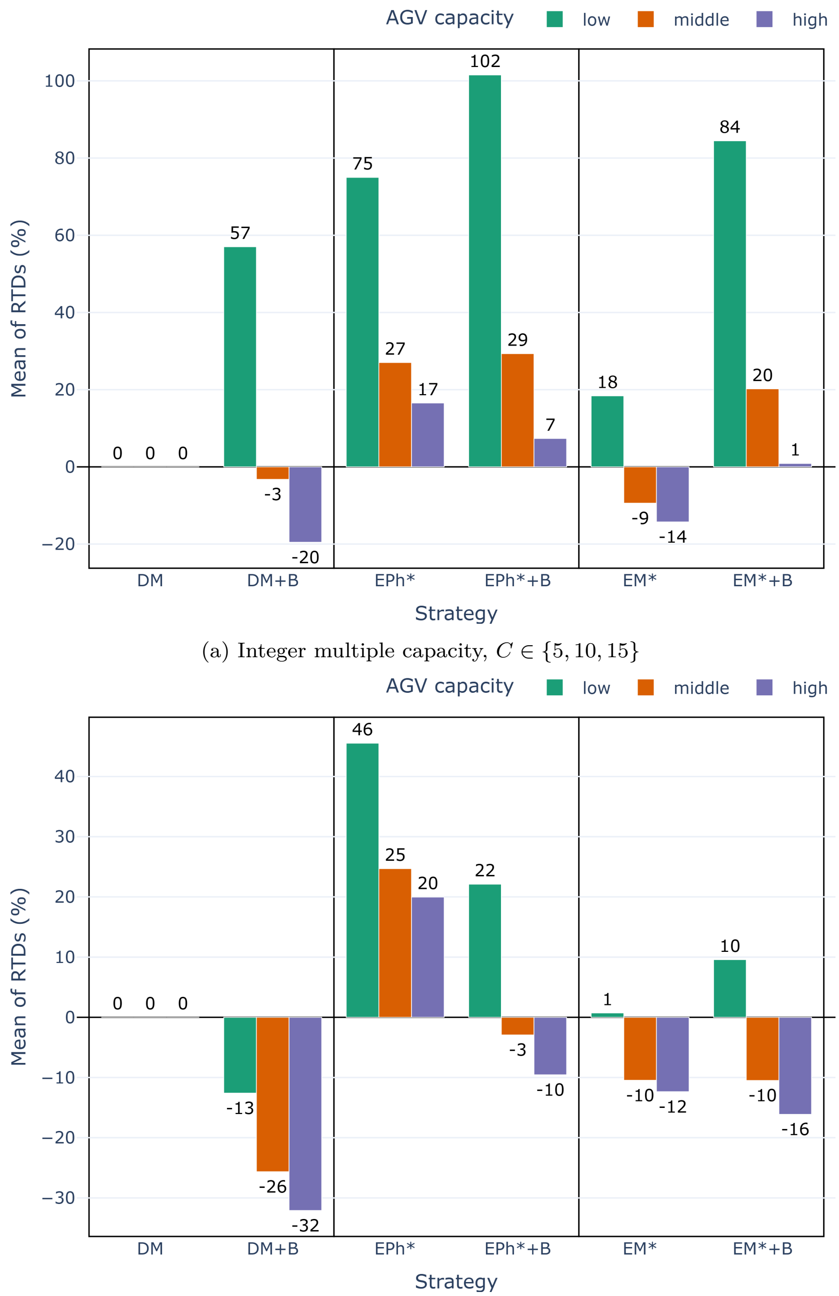
(b) Non-multiple capacity, $C \in \{7, 11, 13\}$

Figure 9: RTD colored by number of process phases.

strategies struggle to maintain performance, as the mismatch between capacity and aisle structure introduces inefficiencies. The data suggest that as the number of production phases increases, the potential for bucketing to mitigate imbalances grows, but only under well-aligned physical layouts.

Figure 10 explores the impact of AGV capacity on RTD more directly. In multiple-aisle-size capacity scenarios with low AGV capacity, bucketing enhances performance for all strategies. This occurs because systems can operate at capacity with no residual jobs, which decreases travel time and increases efficiency. The bar heights in Figure 10a clearly demonstrate this effect, with even the more complex EM* strategy showing consistent gains. Conversely, in non-multiple-aisle-size capacity scenarios, bucketing almost always leads to degraded performance (Figure 10b). Here, the rigidity of pre-defined buckets prevents flexible job assignment, resulting in underutilized AGVs and increased travel distances. When AGV capacity does not correspond to aisle size, the inflexibility of bucketing blocks clever job assignments, generating idle AGVs and increased travel. This trend is most pronounced in the EM* strategy, where throughput is usually lower than the DM baseline, as evidenced by negative RTD values in some bars. These results demonstrate the importance of tuning AGV capacity to match physical layout constraints for the application of advanced scheduling strategies.

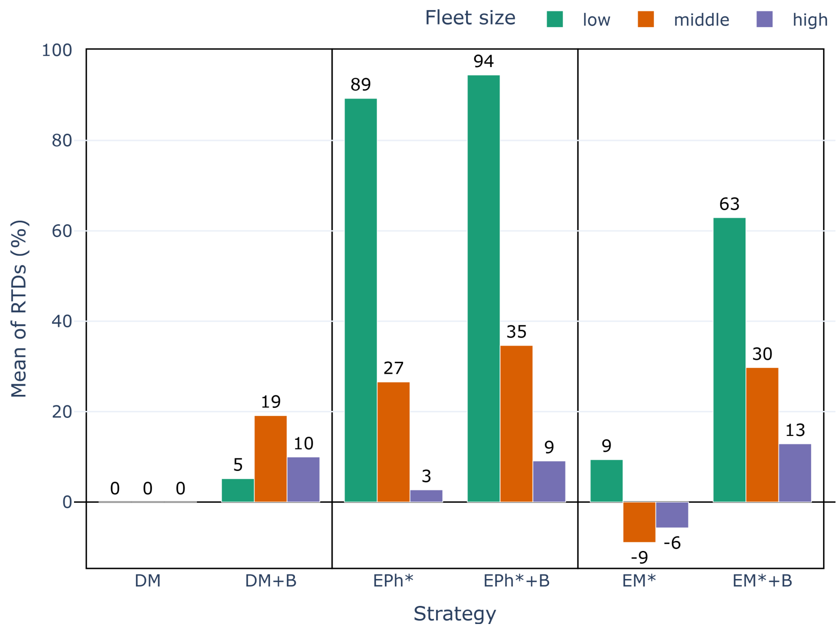
Apart from capacity, which already appears to affect results, the other AGV-related system variable is fleet size. While a small AGV fleet may seem like an easy bottleneck to address through fleet expansion, intsalling more AGVs entails substantial capital and operational costs, both initially and during production, as they require frequent charging and regular maintenance. Also, the more AGVs there are, the greater the chance of traffic blockages. Thus, optimizing dispatching strategies becomes essential, and the goal of this section is to show that choosing the right dispatching strategy is crucial with small AGV fleets. The results in Figure 11 clearly show that small AGV fleets can operate much more efficiently if the right dispatching strategy is used. This can be explained by the nature of systems with low resources: if one (or more) bottleneck is present in the system – the size of AGV fleet in this case – then the effect of different decisions are more visible and the impact is higher. The results in Figure 11a show that with integer-multiple capacities, bucketing provides substantial gains even with constrained fleet sizes, improving system efficiency despite limited transport resources. However, Figure 11b shows that in non-multiple capacity scenarios, bucketing's effectiveness diminishes with small fleets, underscoring the



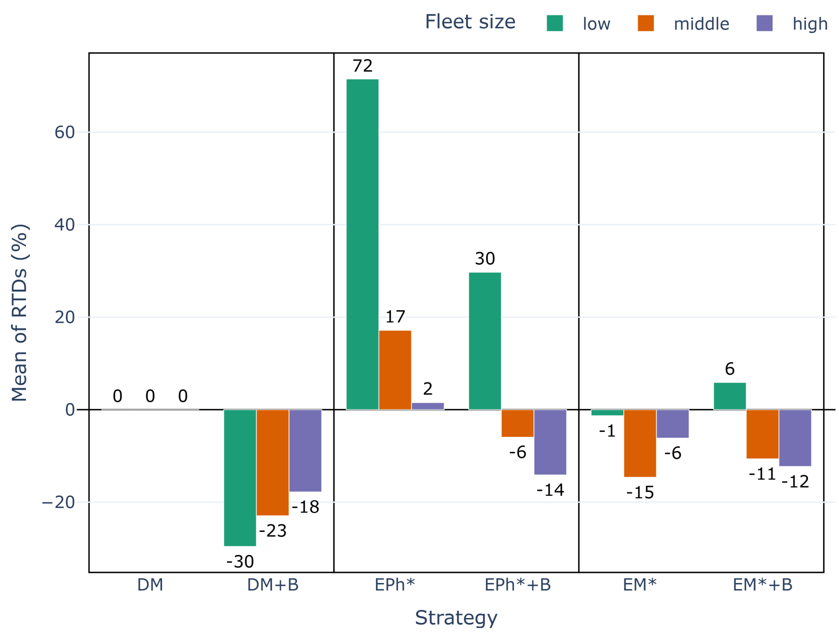
(a) Integer multiple capacity, $C \in \{5, 10, 15\}$

(b) Non-multiple capacity, $C \in \{7, 11, 13\}$

Figure 10: RTD colored by AGV capacity.



(a) Integer multiple capacity, $C \in \{5, 10, 15\}$



(b) Non-multiple capacity, $C \in \{7, 11, 13\}$

Figure 11: RTD colored by fleet size.

importance of flexible job assignment under tight resource constraints.

Regarding machine availability, the last varying component of the experiment runs, Figure 12 shows only subtle differences across availability levels and different dispatching strategies. The pattern of these charts are somewhat similar to the previous figures, namely that in case of AGVs with integer multiple capacities bucketing improves throughput, but other statements cannot be made with high certainty.

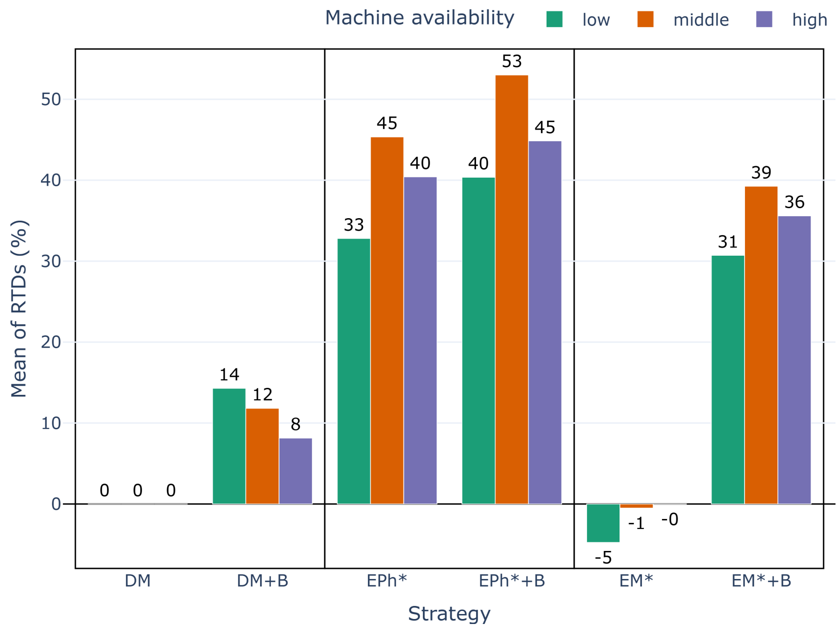
Together, these plots offer a detailed view of how main system variables – such as AGV capacity, fleet size, and machine availability – respond to dispatching strategies and bucketing. The results demonstrate the key conclusion that physical parameters (e.g., capacity and layout) and algorithmic decisions (e.g., bucketing) must be properly tuned to achieve significant performance improvements.

5.4. Superior dispatching strategies

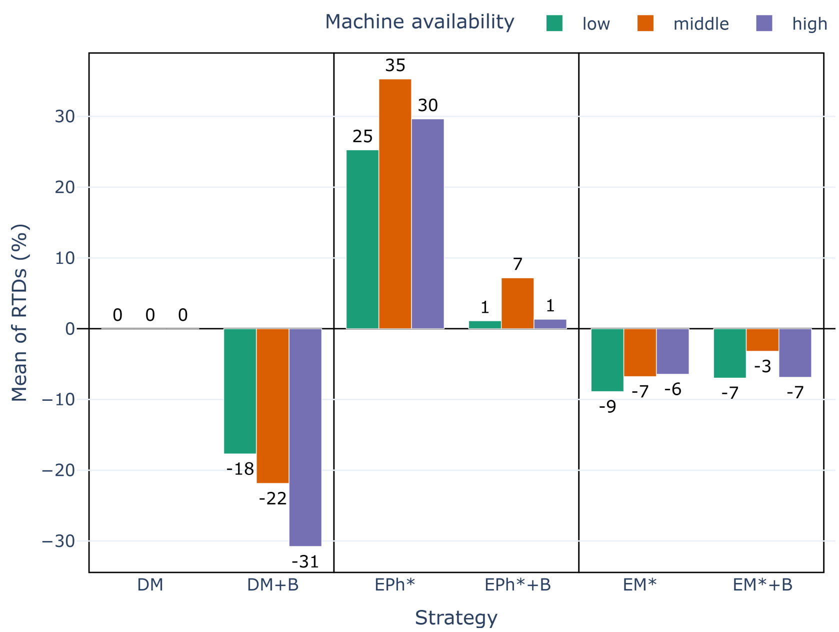
Figure 13 shows the cases in which the maximum throughput was reached by the dispatching strategy, as well as the relative performance in the different configurations. The two pie charts categorize cases in which the AGV capacity is an integer multiple of the aisle size (13a) and cases in which the AGV capacity is not an integer multiple of the aisle size (13b).

For the initial case of integer multiples, the EPh*+B combination method is the clear winner, succeeding in 43% of all cases. This demonstrates that the phase-balancing method with bucketing exploits the system's inherent regularity to maximize material flow. The pure EPh* method is also impressive (16%), lending further support to the importance of phase balancing in this structured environment. Other strategies, such as DM and EM*+B, have vastly lower frequencies, suggesting they are not competitive in these environments.

In the second case, where AGVs cannot synchronize the full potential of the warehouse capacity with aisle size for the non-integer multiple, the EPh* method stands out distinctly more, winning 68.4% of the cases. This reflects the efficacy of dynamic phase balancing in the face of irregularities in loading and transportation patterns. The DM method also performs well (winning 30.4% of the cases), indicating that, in more unstructured environments, travel distances partially compensate for inefficiencies arising from the absence of the right capacity match. The EM* method performed poorly (1.23%), reflecting its lack of effective adaptability. These findings also em-

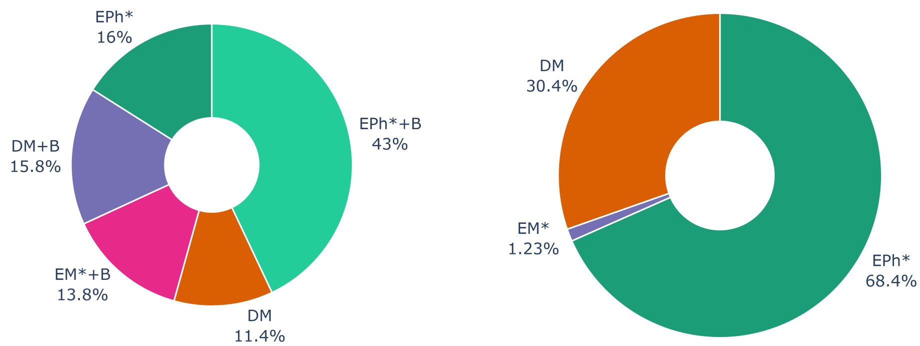


(a) Integer multiple capacity, $C \in \{5, 10, 15\}$



(b) Non-multiple capacity, $C \in \{7, 11, 13\}$

Figure 12: RTD colored by machine availability.



(a) Integer multiple capacity, $C \in \{5, 10, 15\}$

(b) Non-multiple capacity, $C \in \{7, 11, 13\}$

Figure 13: Frequency of superior dispatching strategies.

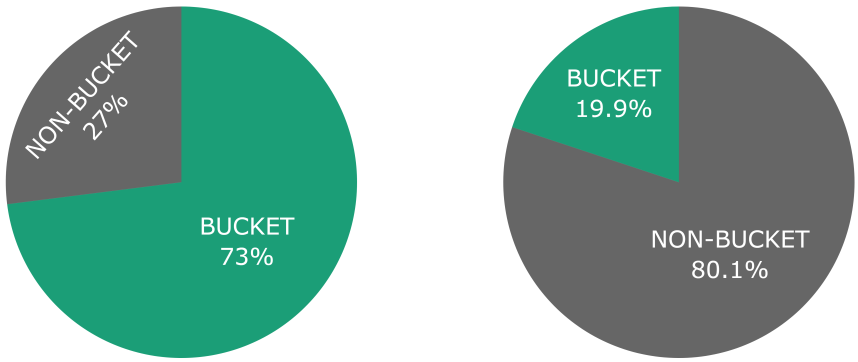
phasize that phase-aware dispatching strategies (EPh* and EPh*+B) are superior, especially when the system layout allows for synchronization.

Overall, the findings demonstrate that selecting the appropriate dispatching approach is critically dependent on the interaction between AGV capacity and aisle structure. In practice, adaptive strategies like EPh* are more effective when the capacity ratio is uneven, while EPh*+B is more effective when the capacity matches the system geometry well.

5.5. Influence of bucketing

The results discussed in the previous subsection hint that there is a strong connection between aisle size, AGV capacity, and bucketing. Several views suggest that bucketing improves the corresponding strategy's non-bucketed version in cases when capacity is an integer multiple of the aisle size. The frequency of such bucketing improvement cases is presented in Figure 14, in which every experiment run with bucketing is compared to the same run without bucketing. The intuition seems to be proven: there is a significant difference between the two type of capacity scenarios (73% vs 19.9%).

In integer-multiple capacity scenarios (Figure 14a), bucketing consistently enhances performance across different strategies and system settings. AGVs are able to operate at their full capacity without leaving residual jobs unassigned, which reduces travel time and increases system efficiency. This effect is particularly pronounced in setups where the layout and transport capacity are well aligned, enabling smoother operation and more predictable scheduling outcomes.



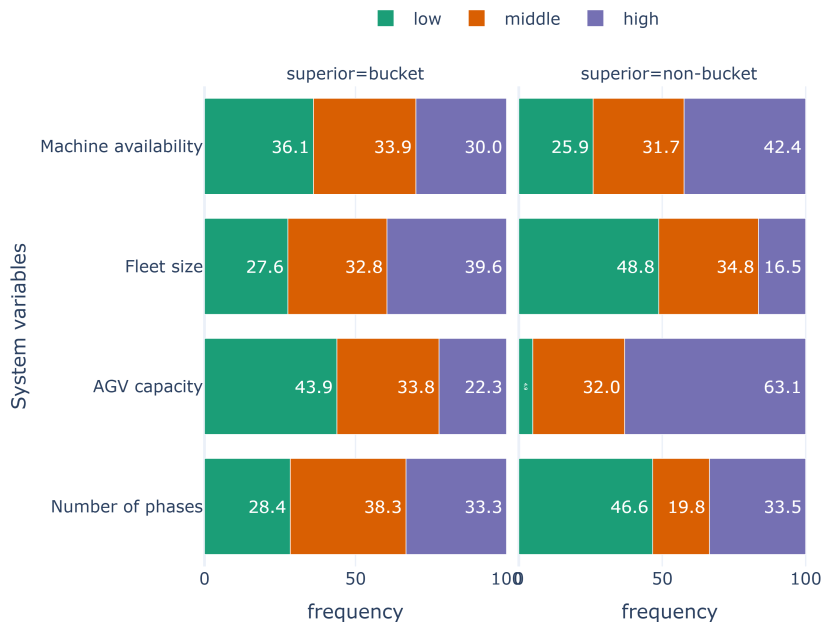
(a) Integer multiple capacity, $C \in \{5, 10, 15\}$

(b) Non-multiple capacity, $C \in \{7, 11, 13\}$

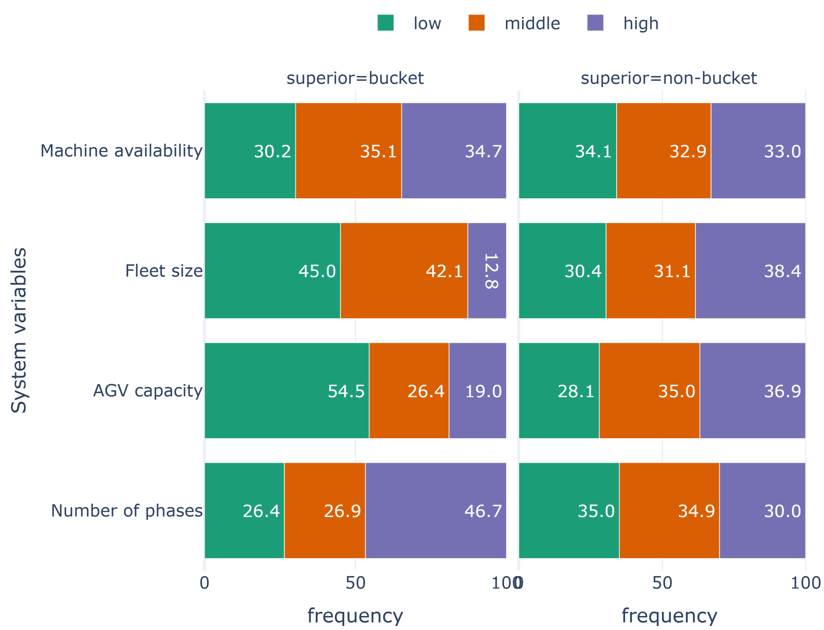
Figure 14: Frequency of throughput improvement via bucketing.

Conversely, in non-multiple capacity scenarios (Figure 14b), bucketing often degrades performance. Here, the lack of alignment between AGV capacity and aisle size creates inefficiencies: the rigid structure of buckets prevents flexible job assignments, leading to underutilized AGVs and increased travel distances. These results underscore how critical it is to synchronize transport capacity with physical layout constraints before applying advanced scheduling techniques like bucketing.

Finally, Figure 15 provides a more detailed breakdown of system variables in scenarios where bucketing outperforms the plain strategy (left) and where the plain strategy outperforms bucketing (right). Figure 15a represents integer-multiple capacity configurations and shows that bucketing tends to outperform for a wide range of configurations. High rates of success are associated with small AGV capacities (43.9%), demonstrating bucketing's proficiency under capacity-constrained transport resources that align well with aisles. Machine availability, fleet size, and phases per plan are roughly equal, suggesting that these variables have a smaller effect on bucketing performance in planned capacity scenarios. However, Figure 15b shows the results for non-multiple capacity configurations. Bucketing remains a good performer in some cases, specifically for low-capacity AGVs (54.5%), but it increasingly loses its advantage. The right-hand plots of both subfigures show cases in which plain strategies dominate. Plain strategies outperform bucketing in integer multiple capacity configurations involving high-capacity AGVs (63.1%) and configurations requiring increased flexibility in job assignment to prevent transport inefficiencies. The results show that, while bucketing



(a) Integer multiple capacity, $C \in \{5, 10, 15\}$



(b) Non-multiple capacity, $C \in \{7, 11, 13\}$

Figure 15: Frequencies (%) of system variables when bucketing wins over plain mode (left), and when plain mode wins over bucketing (right).

provides considerable advantages in well-matched systems, its rigid logic becomes a bottleneck when AGV capacity does not proportionately match the physical structure of the production system.

Overall, these results reaffirm that bucketing can be a useful tool for improving throughput, though its effectiveness depends heavily on system variables. Bucketing is most effective when capacity and aisle size are well-matched. In mismatched setups, however, bucketing can become a limitation rather than a merit. Thus, careful production system analysis is necessary before anticipating performance gains from bucketing.

5.6. Secondary metrics

In addition to throughput as the primary performance indicator, secondary metrics were evaluated to gain further insight into the operational behavior of the dispatching strategies. These metrics capture aspects of system efficiency not reflected in throughput: the total distance traveled by AGVs, non-value-added machine time, and scheduling logic runtime. Unlike earlier analyses, the results in this subsection are aggregated across all scenarios and are therefore not separated by system scenario.

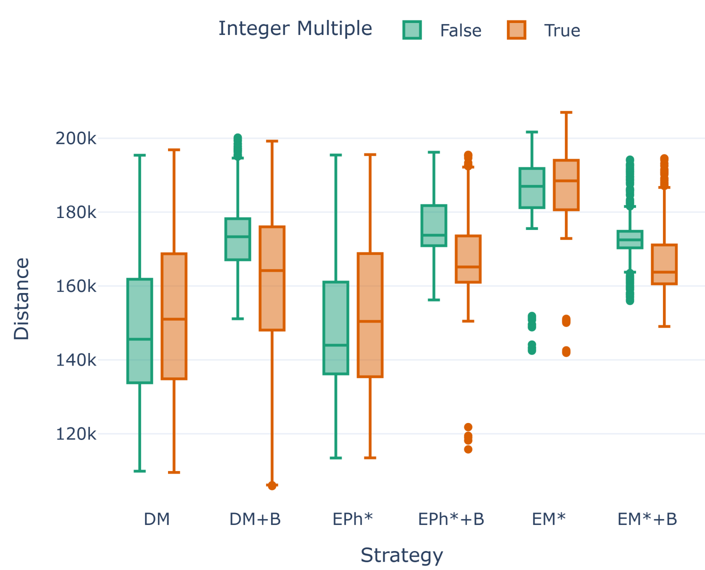


Figure 16: Average of AGV total traveled distances (m) per simulation runs grouped by AGV capacity being and integer multiple of aisle size or not.

Figure 16 shows the average total distance traveled by AGVs under the different dispatching strategies. As expected, the baseline DM strategy yields the shortest distances traveled, since minimizing route length is its explicit optimization objective. Introducing bucketing consistently increases traveled distance, a natural consequence of restricting job allocation to predefined groups and reducing routing flexibility.

Notably, the EPh* strategy achieves travel distances comparable to DM while delivering significantly higher throughput. This suggests that phase-aware balancing can enhance production performance without requiring extra transportation, a desirable feature from operational and energy efficiency standpoints. In contrast, the EM* strategy tends to generate longer routes on average. This behavior can be attributed to its prioritization of machine-level balancing over distance minimization. This prioritization may require AGVs to accept less efficient routes to serve less employed machines temporarily. Across all strategies, no extreme outliers are observed, suggesting stable routing behavior and the absence of pathological cases in the simulation runs.

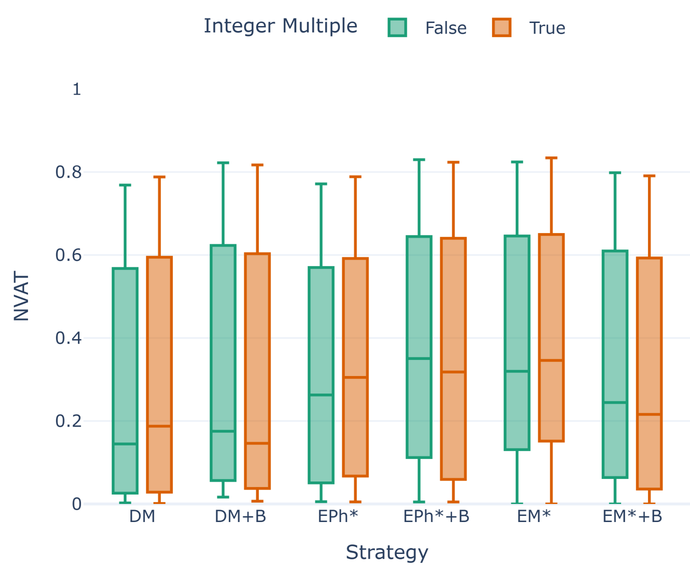


Figure 17: Average of machine NVAT portions per simulation runs grouped by AGV capacity being and integer multiple of aisle size or not.

Figure 17 summarizes the average NVAT for the investigated strategies.

Overall, the differences between the strategies are modest, suggesting that NVAT alone is not an effective indicator of dispatching quality in this setting. Although EPh* has slightly higher NVAT values than DM, EPh* achieves higher throughput. This suggests that, under DM, machines may appear to be utilized more continuously but not necessarily in a manner that advances bottleneck production phases. In other words, lower NVAT does not automatically translate into higher effective throughput.

This observation underscores a significant limitation of using NVAT as the sole optimization target: minimizing idle or blocked machine time does not ensure that the appropriate jobs are processed at the optimal time. Therefore, NVAT should be interpreted as a supportive diagnostic metric rather than an objective to be maximized when the primary goal is to improve throughput in multi-phase production systems.

Finally, the impact of bucketing on computational performance was assessed. Across all experiments, bucketing reduced scheduling runtimes by an average of 51.18% – the mean computation time for simulation over a one-week horizon was 1643.9 seconds without bucketing, and 748.2 seconds with bucketing. This substantial improvement confirms that grouping jobs into buckets effectively reduces the combinatorial complexity of assignment and sequencing decisions. From a practical perspective, this reduction in runtime strengthens the case for bucketing in scenarios where real-time responsiveness or limited computational resources are critical, provided the physical layout and AGV capacity align.

5.7. General applicability

Computational experiments have shown that superior strategies involve combinations of the EPh* technique and bucketing. EPh* is designed for manufacturing systems involving multiple production phases, in which internal logistics are critical for balancing progress across phases. Bucketing, in contrast, leverages groups of nearby machines performing similar tasks with identical processing times by synchronizing their operating cycles, thereby reducing the burden on internal logistics. These techniques can find applications in different manufacturing environments characterized by the above features. However, due to the dependence of the throughput on scenario configuration, it is not possible to provide a theoretical dominance proof for the bucketing heuristic. As the experiments illustrate, the bucketing heuristic is beneficial in some cases and not in others. Therefore, this study empirically investigates its performance over a wide range of environments and identifies those settings where bucketing dominates and should be applied.

6. Decision support

The focus of previous result analytics was to find the effect of different strategies within a wide spectrum of environmental circumstances. However, real life usability of this research comes to light when fab managers wish to improve their their throughput with minimal investment in new machinery or drastic reconfiguration of the layout. Typically, major system parameters are preset: the layout is fixed, the types of machines (therefore production phases) are already known, an AGV fleet of some size is already operating. Therefore, from the decision makers' point of view, the interesting part of such modeling and analytics solution is to see the effect of the dispatching strategies on their system, without dedicating much effort to other system parameters.

To support this goal, a web-based demonstrator was set up that allows users to intuitively see how competing strategies behave. The interactive page[2] is a simple interface through which users can modify system parameters, such as machine availability, fleet size, AGV capacity, and production phases, and immediately observe the corresponding throughput produced by different dispatching strategies. This tool bridges the gap between the simulation results presented here and their application in real-world decision-making. It allows factory operators and logistics engineers to change scenarios, identify bottlenecks, and explore the potential benefits of advanced scheduling strategies without performing costly live experiments on production shopfloors. This current version is a prototype aimed at demonstrating concepts, but it shows how the research work developed here could establish the base of a decision support system for production environments. Successive versions of the tool can incorporate real-time production facility data and provide dynamic recommendations based on current production situations. This type of system will enable decision-makers to proactively adjust their dispatching logic based on changing equipment status, transport congestion, or shifting production priorities. This interactive tool highlights the broader impact of this work, shifting the focus from theoretical performance analysis

[2] `https://juliabergmann.github.io/scheduling-shopfloor-logistics/`

to practical tools that translate advanced scheduling models into actionable outcomes for real manufacturing systems.

Industrial deployments increasingly rely on modular, software-defined control architectures to coordinate heterogeneous material handling systems (e.g., AGVs and OHTs) while reducing integration complexity [33]. Since the scheduler is by design event-driven and considers important sources of stochastic disruptions, such as machine breakdowns, processing variability, and congestion issues, it aligns with the ”relentless production” paradigm. Specifically, the algorithm uses all the necessary dynamic variables (the status of machines, jobs, and vehicles) provided by the Manufacturing Execution System (MES) and only requires some initial master data definitions. Furthermore, the solution has practically zero decision time compared to transport durations, and it does not require special integration with lower-level systems, such as traffic and charging management. In the referred industrial application scenario [32], an earlier AGV scheduler is replaced with dispatching rules – which were precursors of those discussed in current paper – without changing other system components. Any potential integration risks will be caused by noisy and delayed data.

7. Conclusion and future work

This research provides a detailed investigation of AGV scheduling methods in indoor logistics environments, particularly in applications where transportation must be coordinated with dynamic, multi-phase manufacturing processes. By designing and rigorously comparing several dispatching strategies, including phase-aware heuristics and bucketing-enhanced strategies, we proposed a unifying model that is effective, scalable, and grounded in operational insights.

The results validate the main findings outlined at the beginning of the study. First, we demonstrated that considering production phase data in dispatching logic significantly improves performance. This validates the idea that manufacturing's time structure is not a limitation, but rather an optimization opportunity.

Second, we discovered that intelligence does not have to be based on explicit predictions. Though simple and plain, the bucketing heuristic – which groups logistical jobs that are expected to appear simultaneously, therefore acts as an approximation to real-time forecasting – proved its efficiency. It swiftly imparts system background knowledge to scheduling decisions, is interpretable, and is surprisingly effective. Moreover, we found that bucketing increases throughput by 14.28% in average in cases when aisle sizes and AGV capacities are well aligned. This opens up new control points of view that are lightweight yet intelligent, offering a tradeoff between rule-based scheduling and full predictive modeling.

Third, our experiments illustrate the important fact that context matters greatly. The effectiveness of a dispatching strategy depends heavily on the environment. When AGV capacity matched aisle width, bucketing-based strategies improved both throughput and computational speed. In less uniform scenarios, flexible policies like EPh* prevailed, demonstrating resilience to uncertainty and complexity. These results support our hypothesis that there is no generic solution and that efficient logistics systems need to be tailored to their operating environment.

Aside from these specific results, the broader contribution lies in combining transportation and production considerations within a single scheduling framework. This integration, often handled separately in the literature, is central to high-throughput, low-latency logistics. By modeling transportation not as a separate problem but as an integral part of the overall production system, we showed a new approach for AGV scheduling: one that is not only distance-based but also phase-aware, and exploits the coordination between facility layout and AGV capacity.

This work establishes also a foundation for future research. One possible extension is to incorporate machine learning or hybrid predictive elements into our phase-aware heuristics to enable more nuanced decision-making under uncertainty. Another direction is to explore decentralized, congestion-aware dispatching methods, particularly in multi-AGV scenarios with shared infrastructure. Also, a fleet made of vehicles with different configurations, and dynamically changing environments require a modified methodology. From a computational perspective, future work could explore using GPU acceleration and parallelization based on Message Passing Interface to reduce scheduling latency further and enable large-scale, high-fidelity simulations and real-time dispatching. Finally, transitioning from simulation to deployment at a digital twin or live testbed will be critical to unlocking the real-world potential of the methods described here.

Overall, our paper offers more than a set of algorithms; it provides a new perspective on indoor logistics, offering a way to embrace complexity without sacrificing simplicity and motivating rigorous analysis with practical applications. We view our work as a stepping stone to the next generation of AGV systems that respond intelligently to context.

Acknowledgement

This research was supported by the Doctoral Student Scholarship Program of the Co-operative Doctoral Program of the Ministry of Innovation and Technology, financed by NRDIO, by the European Union within the framework of the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002), and by the TKP2021-NKTA-01 NRDIO grant on "Research on cooperative production and logistics systems to support a competitive and sustainable economy".

Disclosure statement

The authors declare no competing interests.

Data availability statement

The data that support the findings of this study are available from the corresponding author, J. Bergmann, upon reasonable request.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work the authors used DeepL in order to improve the language of this work. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Notes on contributors

Júlia Bergmann is a PhD candidate at Eötvös Loránd University in Budapest, Hungary, and also a research fellow at the HUN-REN Institute for Computer Science and Control (SZTAKI), Budapest, Hungary. Her research focuses on developing adaptive, transparent, and scalable machine learning methods for industrial data analytics in nonstationary manufacturing and logistics settings. She engages in characterizing data, selecting models, and applying them to decision-making in operations. This approach integrates human experience and advanced analytics to improve production performance.

András Kovács is a senior research fellow at the HUN-REN Institute for Computer Science and Control (SZTAKI), Budapest, Hungary. His research focuses on modeling and solving optimization problems in manufacturing using methods from combinatorial optimization, mathematical programming, constraint programming, and bilevel programming. He has authored 100+ scientific publications in these fields.

József Váncza is Head of the Research Laboratory for Engineering and Business Intelligence at the HUN-REN Institute for Computer Science and Control (SZTAKI), Budapest, Hungary. His research interests include engineering applications of artificial intelligence, production informatics, cyber-physical production and logistics systems, cooperative and sustainable production in networks. He has been involved in university education for three decades, now he is honorary professor at the Faculty of Mechanical Engineering of the Budapest University of Technology and Economics (BME). He is Fellow of the International Academy for Production Engineering (CIRP), and of the International Academy of Engineering and Technology (AET). For his 240+ publications he received 7000 citations.

References

[1] K. Ellithy, M. Salah, I. S. Fahim, R. Shalaby, AGV and Industry 4.0 in warehouses: a comprehensive analysis of existing literature and an innovative framework for flexible automation, The International Journal of Advanced Manufacturing Technology 134 (1) (2024) 15–38.

[2] I. Vlachos, R. M. Pascazzi, G. Zobolas, P. Repoussis, M. Giannakis, Lean manufacturing systems in the area of Industry 4.0: A lean automation plan of AGVs/IoT integration, Production Planning and Control 34 (4) (2023) 345–358.

[3] I. Vlachos, R. M. Pascazzi, M. Ntotis, K. Spanaki, S. Despoudi, P. Repoussis, Smart and flexible manufacturing systems using autonomous guided vehicles (AGVs) and the internet of things (IoT), International Journal of Production Research 62 (15) (2024) 5574–5595.

[4] J. Fang, B. Cheang, A. Lim, Problems and solution methods of ma-

chine scheduling in semiconductor manufacturing operations: A survey, Sustainability 15 (17) (2023) 13012.

[5] C. Khamyat, P. Charnsethikul, AGV scheduling problem with alternative pick up and delivery nodes by modified TSP approach, in: Proceedings of the 7th Asia Pacific Industrial Engineering and Management Systems Conference, Bangkok Thailand, 2006, pp. 17–20.

[6] K. Sidorov, A. Morozov, A review of approaches to modeling applied vehicle routing problems, arXiv preprint arXiv:2105.10950 (2021).

[7] W. Kool, H. van Hoof, J. Gromicho, M. Welling, Deep policy dynamic programming for vehicle routing problems, in: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Springer, 2022, pp. 190–213.

[8] M. Drótos, P. Györgyi, M. Horváth, T. Kis, Suboptimal and conflict-free control of a fleet of agvs to serve online requests, Computers & Industrial Engineering 152 (2021) 106999.

[9] M. Horváth, T. Tamási, A general modeling and simulation framework for dynamic vehicle routing, EURO Journal on Transportation and Logistics (2025) 100159.

[10] B. Kang, C. Park, H. Kim, S. Hong, A digital twin calibration for an automated material handling system in a semiconductor fab, Journal of Manufacturing Systems 80 (2025) 1013–1028.

[11] S. C. Sarin, A. Varadarajan, L. Wang, A survey of dispatching rules for operational control in wafer fabrication, Production Planning and Control 22 (1) (2011) 4–24.

[12] K.-B. Reith, P. Boden, M. Däumler, S. Rank, T. Schmidt, R. Hupfer, Evaluating automated guided vehicle system characteristics in semiconductor fab automated material handling systems, in: 30th Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC), IEEE, 2019, pp. 1–6.

[13] P. Boden, S. Rank, T. Schmidt, Control of heterogenous AMHS in semiconductor industry under consideration of dynamic transport carrier

transfers, in: 22nd IEEE International Conference on Industrial Technology (ICIT), Vol. 1, IEEE, 2021, pp. 1403–1408.

[14] A. Benzoni, C. Yugma, P. Bect, Vehicle look-ahead dispatching for overhead hoist transport system in semiconductor manufacturing, IEEE Transactions on Semiconductor Manufacturing 36 (1) (2023) 130–138.

[15] D. Chen, J. Zhang, L. Wu, P. Zhang, Y. Lv, J. Wang, H. Wang, Optimisation method for semiconductor wafer manufacturing system scheduling: Reinforcement learning with decision graph guiding, Journal of Manufacturing Systems 82 (2025) 1158–1170.

[16] I. F. Vis, Survey of research in the design and control of automated guided vehicle systems, European Journal of Operational Research 170 (3) (2006) 677–709.

[17] Z. Qiu, J. Long, Y. Yu, S. Chen, Integrated task assignment and path planning for multi-type robots in an intelligent warehouse system, Transportation Research Part E: Logistics and Transportation Review 194 (2025) 103883.

[18] J. Xin, Q. Yuan, A. D'Ariano, G. Guo, Y. Liu, Y. Zhou, Dynamic unbalanced task allocation of warehouse AGVs using integrated adaptive large neighborhood search and Kuhn–Munkres algorithm, Computers & Industrial Engineering 195 (2024) 110410.

[19] H. Niu, W. Wu, Z. Xing, X. Wang, T. Zhang, A novel multi-tasks chain scheduling algorithm based on capacity prediction to solve AGV dispatching problem in an intelligent manufacturing system, Journal of Manufacturing Systems 68 (2023) 130–144.

[20] K. Li, T. Liu, P. R. Kumar, X. Han, A reinforcement learning-based hyper-heuristic for AGV task assignment and route planning in parts-to-picker warehouses, Transportation Research Part E: Logistics and Transportation Review 185 (2024) 103518.

[21] M. De Ryck, D. Pissoort, T. Holvoet, E. Demeester, Decentral task allocation for industrial AGV-systems with routing constraints, Journal of Manufacturing Systems 62 (2022) 135–144.

[22] D. Gyulai, J. Bergmann, J. Váncza, Adaptive network analytics for managing complex shop-floor logistics systems, CIRP Annals 69 (1) (2020) 393–396.

[23] J. Bergmann, D. Gyulai, J. Váncza, Adaptive AGV fleet management in a dynamically changing production environment, Procedia Manufacturing 54 (2021) 148–153.

[24] G. S. May, C. J. Spanos, Fundamentals of semiconductor manufacturing and process control, John Wiley & Sons, 2006.

[25] J. Wan, H. Shin, Predictive vehicle dispatching method for overhead hoist transport systems in semiconductor fabs, International Journal of Production Research 60 (10) (2022) 3063–3077.

[26] F.-S. Zhou, R. Hu, W.-B. Zhang, B. Qian, Z.-Q. Zhang, L. Wang, Spatial decomposition-based variable neighborhood search algorithm for the integrated semiconductor backend production and distribution scheduling problem, Computers & Operations Research (2026) 107388.

[27] K. A. Hoque, O. Ait Mohamed, Y. Savaria, Towards an accurate reliability, availability and maintainability analysis approach for satellite systems based on probabilistic model checking, in: 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2015, pp. 1635–1640.

[28] S.-C. Lee, S. Demidenko, K.-H. Lee, Ic handler throughput evaluation for test process optimization, in: 2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007, IEEE, 2007, pp. 1–6.

[29] K.-M. Choi, S.-J. Lee, Physicochemical characteristics and occupational exposure of silica particles as byproducts in a semiconductor sub fab, International Journal of Environmental Research and Public Health 19 (3) (2022) 1791.

[30] S. Lee, D. Anderson, L. Nguyen, H. Takiar, Integrated assembly and strip test of chip scale packages, Chip Scale Review (2000).

[31] J. Kiefer, Sequential minimax search for a maximum, Proceedings of the American Mathematical Society 4 (3) (1953) 502–506.

[32] D. Gyulai, J. Bergmann, A. Lengyel, B. Kádár, D. Czirkó, Simulation-based digital twin of a complex shop-floor logistics system, in: 2020 Winter Simulation Conference (WSC), IEEE, 2020, pp. 1849–1860.

[33] M. U. Farooq, Y. J. Jang, Heterogeneous multi-agent fleet control system for material handling in a Software-Defined Factory, Journal of Manufacturing Systems 85 (2026) 513–530.