Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

SIMULATION-BASED AGV MANAGEMENT WITH A LINEAR DISPATCHING RULE

Nitish Singh
Jeroen B.H.C. Didden
Alp Akcay
Tugce Martagan
Ivo J.B.F. Adan

Department of Industrial Engineering and Innovation Sciences
Eindhoven University of Technology
PO Box 513, Eindhoven, 5600MB, North Brabant, THE NETHERLANDS

ABSTRACT

This paper considers the problem of real-time dispatching of a fleet of heterogeneous automated guided vehicles (AGVs) with battery constraints. The AGV fleet is heterogeneous in terms of material handling capabilities; some can tow loads, some can lift loads while others manipulate loads with the assistance of a robotic arm. Transport requests arrive in real-time and include a soft time window, with late delivery incurring tardiness costs. Transport requests need to be assigned to a capable AGV based on required material handling capabilities with the objective to minimize a weighted sum of tardiness costs of transport requests and travel costs of AGVs. In this paper, an AGV-specific linear dispatching rule (LDR) learning approach is proposed to assign AGVs to randomly arriving transport requests in real time over a finite horizon. The proposed approach is compared with a heuristic policy from practice by using real-world data provided by our industry partner.

1 INTRODUCTION

Industry 4.0 aims to bring a fourth wave of industrial revolution in manufacturing. A recent trend is a shift from the traditional top-down and central architectures towards a more decentralized control wherein systems and devices possess their own intelligence. A trend is to enable semi-centralization, where the central entity makes a decision after gathering information from devices that retain their intelligence independently (Didden et al. 2021). This intelligence may allow devices to have a certain autonomy in decision-making in the presence of information within a production or logistic setting. Recently, smarter mobility within production areas has been made possible due to the wide adoption of AGVs. AGVs are autonomous guided vehicles that move supplies and materials between multiple locations, including workstations, storage facilities, and shipping and receiving areas (Confessore et al. 2013; Vivaldini et al. 2016). Because of improved process flexibility, space utilisation, product safety, computer-integration, and control, the use of AGVs has increased significantly over the last decade. AGV technology has enhanced with new advances in their flexibility, where each vehicle could have specific capabilities (for example, one can lift light or heavy goods, while another can tow loads) to do diverse jobs automatically (Riazi et al. 2019; De Ryck et al. 2020).

This paper deals with the dynamic management of transport requests with a fleet of AGVs, wherein AGVs may have differences in their material handling capabilities, operating costs, and speeds to cater to a more diverse set of requirements. It is an outcome of our collaboration with Brainport Industries Campus (BIC), which is a new high-tech campus constructed in Eindhoven, The Netherlands. The campus is a joint initiative of high-tech suppliers that co-exist and collaborate on multiple fronts such as shared logistics

979-8-3503-6966-3/23/$31.00 ©2023 IEEE

and warehousing (Brainport Industries Campus 2020) in this campus. AGVs are an important part of these shared resources. Since multiple tenants collectively make use of a pool of heterogeneous AGVs, widely used heuristics to manage homogeneous AGV fleets become inefficient at handling heterogeneous AGV fleets. Therefore, practitioners are looking for a better understanding on how to manage heterogeneous fleets. A widely used stream of heuristics that is well-suited to cope with increasing problem sizes includes dispatching rules. Dispatching can be considered as a crucial and also one of the most challenging design aspects of an AGV management and control system (De Ryck et al. 2020). Le-Anh and De Koster (2006) define dispatching as selecting and assigning tasks to vehicles, including the routes that vehicles travel to accomplish these tasks. If all tasks are known prior to the planning period, the problem can be solved simultaneously for all tasks. Literature refers to this type of problem as static or offline dispatching problems. However, in practice, tasks are often unknown in advance and revealed over time, i.e. dispatching relies on real-time information. Hence, the dispatching problem needs to be solved sequentially. Literature refers to this type of problem as a real-time, dynamic, or online dispatching problem (Vis 2006). In this paper, we consider a dynamic AGV dispatching problem. Most dispatching rules used in literature are single-attribute ones, which take a dispatching action based on one parameter only.

Even though single-attribute dispatching rules have proved to work well in the past, the downside is that they are highly dependent on their environment (Tay and Ho 2008). Holthaus and Rajendran (1997) show that when the utilization of the shop floor changes, other rules become more efficient. However, in most cases, a combination of various dispatching rules outperforms single-dispatching rules. Multiple studies that use a Linear Dispatching Rule (LDR) for AGV scheduling (Jeong and Randhawa 2001; Le-Anh and De Koster 2005; Bilge et al. 2006; Guan and Dai 2009) show that combining multiple attributes can outperform single-attribute dispatching rules significantly. As stated in Kim et al. (2021), LDRs are underrepresented in the literature, while it has the potential to outperform state-of-the-art competitors. Research on learning the weights of different attributes for an LDR in AGV dispatching is minimal, while the implementation of LDRs has also proved to work well in various other domains, most notably when the rules are tailored to each entity on the shop floor specifically (e.g., machines or AGVs). Kim et al. (2021) and Didden et al. (2023) study LDRs in machine shop settings. Here, the authors use a gradient-based approach to adaptively adjust the weights of each attribute depending on the environment. Similarly, Wang et al. (2019) look at using a genetic algorithm to create combined dispatching rules, showing that these tailored rules can outperform single dispatching rules.

The specific problem we address in this paper is the dispatching of AGVs with battery constraints and heterogeneous capabilities. Our problem holds some similarities to the Electric Vehicle Routing Problem with Time Windows (EVRPTW) addressed by Keskin and Çatay (2016) and Zhao and Lu (2019). The EVRPTW considers a fleet of Electric Vehicles (EVs) having limited driving range due to their battery capacities, which may need to visit charging stations while servicing customers in their routes. We address a dynamic EVRPTW problem that has similarities with the works of Singh et al. (2022). Our problem, however, distinguishes from the EVRPTW literature by the following features to comply with industrial needs. First, the fleet of AGVs considered in this study is heterogeneous in terms of travel speeds, costs, and, more importantly, capabilities to service different types of requests from various tenants. Second, partial charging for the AGVs is allowed under consideration of a critical battery threshold. Third, we consider both the travel costs of AGVs and the tardiness costs of transport requests, where different types of AGVs and requests have different unit travel costs and penalty charges, respectively. Finally, we consider dynamic request arrival scenarios where the set of transport requests is unknown at the beginning of the planning horizon and revealed over time to the dispatcher. Motivated by our industry collaboration at BIC, the combination of these features constitutes the main novelty of the problem. We aim to make decisions on which AGVs the transport requests are assigned so that a weighted average of the total travel and total tardiness costs is minimized. For this particular problem, to the best of our knowledge, we are the first to propose an LDR and a mechanism to learn its parameters. The proposed approach is compared with a heuristic policy commonly used in practice.

The remainder of this paper is organized as follows. The problem is formally described in Section 2. The proposed method, its training procedure, and details on determining the relevant attributes (also referred to as features) are presented in Section 3. We describe the simulation model, discuss the training behaviour, and present computational results in Section 4. Finally, conclusions and future research directions are discussed in Section 5.

2 PROBLEM DESCRIPTION

The problem concerns a set of transport requests R serviced by a heterogeneous AGV fleet V with battery constraints. The AGVs travel over a two-dimensional space and serve the transport requests that randomly arrive during a time horizon of length H. The layout consists of a set of nodes where each node can represent either a pickup-or-delivery location or a charging station. Since the transport requests arrive randomly, the set R is unknown to the decision maker at the beginning of the time horizon. The source (pickup) and destination (delivery) nodes of request $r \in R$ are denoted with $s_r^{\mathscr{R}}$ and $d_r^{\mathscr{R}}$, respectively. Each request $r \in R$ requires a set of capabilities $A_r^{\mathscr{R}}$ from an AGV for transport, and it has a time window $[e_r^{\mathscr{R}}, l_r^{\mathscr{R}}]$, where $e_r^{\mathscr{R}}$ is the earliest pickup time and $l_r^{\mathscr{R}}$ is the latest delivery time of the request. If an AGV arrives at the destination of request r after its latest delivery time, the tardiness cost $c_r^{\mathscr{R}}$ is incurred per unit time.

Each request needs to be performed by exactly one capable AGV in the heterogeneous fleet. The AGV $k \in V$ has a set of capabilities $A_k^{\mathscr{V}}$ and is capable of servicing request $r \in R$ if $A_r^{\mathscr{R}} \subseteq A_k^{\mathscr{V}}$ holds, i.e., the capability requirements of a request are included in the capabilities of an AGV. For example, with a request classified as a 'heavy load' that needs to be lifted on a pallet, only those AGVs that have the capability to lift heavy loads can carry out that request.

Unnecessary travel of an AGV is undesired. Therefore, each AGV has a travel cost $c_k^{\mathscr{V}}$ per unit time. While an AGV is traveling, the battery charge level decreases proportionally with the traversed distance at a discharging rate of $d_k^{\mathscr{V}}$, and the AGV may need to visit a charging station before continuing operations. Here, the battery is recharged at a rate of $r_k^{\mathscr{V}}$. The charge level of an AGV (measured as a percentage) must take values between the minimum and maximum battery charge level denoted by b^l percent and b^u percent, respectively. The AGV k must recharge its battery if the charge level drops below a critical threshold $\underline{b}_k^{\mathscr{V}}$, and the recharging duration should be long enough to allow the charge level to reach at least this threshold. In other words, before starting a new request, the charge level of AGV k must be at or above $\underline{b}_k^{\mathscr{V}}$ percent. The critical threshold is a prespecified parameter for a given network of nodes to assure that an AGV can reach any request destination and a charging station right afterward if needed. AGVs are not permitted to visit a charging station while performing a request (i.e., carrying a load). An AGV also cannot serve more than one request at the same time. In addition, AGVs can detect objects on the shop floor and move around them while traveling. Consequently, collisions between AGVs can be avoided by hardware, thus are not considered in this paper.

The objective of a central operator is to minimize the total travel costs of AGVs and the tardiness costs of transport requests during time horizon H. The central operator may prioritize travel or tardiness costs by adjusting the weight coefficients η_1 and η_2 that correspond to travel cost and tardiness cost, respectively. Thus, the objective function is the total weighted cost over the time horizon H, and it is given by

$$\eta_1 \sum_{k \in V} c_k^{\mathscr{V}} \pi_k + \eta_2 \sum_{r \in R} c_r^{\mathscr{R}} \mathscr{T}_r, \tag{1}$$

where π_k represents the total travel time of AGV k and $\mathscr{T}_r$ represents the tardiness of request r.

When an AGV finishes the delivery of a request, it is routed to the nearest charging station. When a new transport request enters the system, it must be decided which AGV is assigned to this transport request. The assignment of a transport request to an AGV is considered feasible when the AGV is idle or charging with a charge level equal to or larger than the critical charge level and contains the capabilities required by the request. Once a transport request is assigned, the AGV must fulfill it and new requests may not be assigned to it. Also, an AGV k that has been routed to a charging station may still be assigned a transport request if its charge level is greater than $\underline{b}_k^{\mathscr{V}}$.

3 SOLUTION METHOD

In Section 3.1, we propose an AGV assignment policy that makes the assignment decisions based on a linear dispatching rule. We use discrete-event simulation to calculate the objective function in (1) and aim to learn the parameters of the linear dispatching rule that minimizes this objective. Section 3.2 presents the details of the learning algorithm. In Section 3.3, we discuss how the features that will be included in the linear dispatching rule are determined.

3.1 Linear Dispatching Rule

We develop a Linear Dispatching Rule (LDR) to make the AGV assignment decisions. The LDR assumes for each AGV in the fleet that the cost of an assignment decision can be represented as a linear function of a set of problem features (see Table 2 for the list of features considered in our study). At the arrival of a new transport request, the so-called bid value is calculated for each *capable* AGV. The bid value for AGV k is given by

$$p_k = \sum_{u=1}^{U} w_{uk} \cdot X_{uk},$$

where U is the number of selected features, X_{uk} is the feature u of AGV k, and $w_{uk} \in \mathbb{R}$ is the weight associated with the feature X_{uk}. The bid values of eligible AGVs are compared, and the AGV with the lowest bid value (i.e., minimum cost) is dispatched to the incoming transport request, as outlined in Algorithm 1:

Algorithm 1 Assignment decision

Require: Weight w_{uk} and value X_{uk} of feature $u \in \{1,\ldots,U\}$ and AGV $k \in \{1,\ldots,V\}$; request $r \in R$

1: $V_C \leftarrow \{k \in V \mid A_r^{\mathscr{R}} \subseteq A_k^{\mathscr{V}}\}$
2: **for** k = 1: $|V_C|$ **do**
3: $\quad p_k = \sum_{u=1}^{U} w_{uk} \cdot X_{uk}$
4: **end for**
5: $k^* \leftarrow \text{argmin}\{p_k \mid \forall k \in V_C\}$
6: Assign request r to AGV k^*

Intuitively, assuming the cost can be approximated as a linear function of the problem features, the LDR can be a promising policy if its parameters w_{uk}, $u \in \{1,\ldots,U\}$, $k \in \{1,\ldots,V\}$ are chosen such that the resulting cost function accurately estimates the cost values coming from the simulation model. The objective of the learning algorithm in Section 3.2 is to obtain the set of weights that leads to the LDR with minimum expected cost.

3.2 Learning Algorithm

The training procedure relies on the Estimation of Distribution Algorithm (EDA) approach (Kim et al. 2021) to train the weight parameters of all AGVs. It is based on the idea of assuming each unknown weight parameter follows a univariate normal distribution and learning the parameters of this distribution by using the cost realizations of the corresponding LDR obtained by simulation.

To be specific, we initialize the process by defining a univariate normal distribution for each weight parameter w_{uk} with mean μ_{uk} and standard deviation σ_{uk}. The values μ_{uk} and σ_{uk} can be initialized arbitrarily, our goal is to iteratively learn the values of these parameters that minimize the cost under the corresponding LDR. The EDA approach entails creating a population consisting of N individuals. Let $\mathscr{N}(a,b)$ denote a univariate normal distribution with mean a and variance b. For each AGV k, feature u, and individual

n, we sample a perturbation value η^n_{uk} from the distribution $\mathscr{N}(0, e^{\sigma_{uk}})$ and create two weight values w^{n+}_{uk} and w^{n-}_{uk}, where $w^{n-}_{uk} = \mu_{uk} - \eta^n_{uk}$ and $w^{n+}_{uk} = \mu_{uk} + \eta^n_{uk}$. With the set of computed weights, two simulations are performed, one with w^{n+}_{uk} and one with w^{n-}_{uk}. The costs are collected for each individual as $f(w^{n+}_{uk})$ and $f(w^{n-}_{uk})$ where $f(.)$ represents the costs (Equation 1) that are obtained when the simulation is run based on the policy defined in Algorithm 1, i.e., when the AGVs are dispatched with sampled weights from that particular iterations. The costs are then normalized across the population to obtain $\hat{f}(w^{+}_{uk})$ and $\hat{f}(w^{-}_{uk})$, as shown in Equation (2):

$$\hat{f}(w^{n\pm}_{uk}) = \frac{f(w^{n\pm}_{uk}) - \bar{x}[f(w^{n\pm}_{uk})]}{\bar{\sigma}[f(w^{n\pm}_{uk})]}, \tag{2}$$

where $\bar{x}[.]$ and $\bar{\sigma}[.]$ are the mean and standard deviation of the simulation output data $f(w^{n\pm}_{uk})$, $n = 1, \ldots, N$. With the normalized costs, the gradients of the cost function with respect to the parameters μ_{uk} and σ_{uk}, denoted by $\Delta\mu_{uk}$ and $\Delta\sigma_{uk}$, respectively, are calculated as follows (Kim et al. 2021):

$$\Delta\mu_{uk} = \frac{1}{N} \sum_{n \in N} \frac{\eta_{uk}}{e^{\sigma_{uk}}} \left(\frac{\hat{f}(w^{n+}_{uk}) - \hat{f}(w^{n-}_{uk})}{2} \right) \tag{3}$$

$$\Delta\sigma_{uk} = \frac{1}{N} \sum_{n \in N} \frac{\eta^2_{uk} - e^{\sigma_{uk}}}{\sqrt{e^{\sigma_{uk}}}} \left(\frac{\hat{f}(w^{n+}_{uk}) - \hat{f}(w^{n-}_{uk})}{2} \right) \tag{4}$$

We use the gradients calculated by Equations (3) and (4) to update the parameters μ_{uk} and σ_{uk} as part of Adam optimizer, which is an extended version of the stochastic gradient descent optimizer (Kingma and Ba 2015). The training procedure is repeated for a predefined number of training iterations or until the learned standard deviation parameter converges to 0. Learning parameter α_0 in the Adam optimizer is decayed exponentially, i.e., $\alpha = \alpha_0 \cdot \exp\left(-\frac{g-1}{\delta}\right)$, where δ is the learning rate decay, α_0 is the initial learning rate, and g is the current iteration number. Learning parameters, including β_1 and β_2 of the Adam optimizer, were determined experimentally and are reported in Table 1. Initial μ_us are set to zero and initial σ_us are set to 0.3. Note that the developed simulation model is able to utilize multiple cores of a workstation, thus greatly increasing the speed of learning by evaluating each individual of the population on separate processing cores of the workstation.

Table 1: Parameter values of the learning algorithm.

Parameter	Description
Number of individuals, N	$U+3$
Initial learning rate, α_0	0.1
Learning rate decay, δ	1000
Adam parameter 1, β_1	0.9
Adam parameter 2, β_2	0.999

3.3 Feature Selection

We conduct a feature selection analysis to identify contributing features since using all available features can slow down the learning process while including too little may result in inefficient learning. Table 2 highlights the available features that are provided by the simulation model for assignment decisions.

We carry out a statistical analysis on available features reported in Table 2. We exclude each feature one at a time and report the percentage improvement in the costs of the corresponding trained model.

Table 2: Features considered by the LDR.

Feature	Description
DTR	Distance between AGV's current location and transport request's pickup location
CH	Charge level of the AGV when it is available to carry out the transport request
TC	Travel cost of the AGV
DTC	Distance between the AGV's current location to the center of the layout
LOC	Location of the AGV
ETA	Expected time of arrival at the transport request's pickup location
CTR	Penalty cost of the transport request
TS	Time of the day
LDT	Time until the latest delivery time of the transport request

The results are presented in Table 3. The outcome of each configuration with respect to the first row is reported as Gap and a negative value indicates an improvement in the objective value (lesser cost). First, we remove each feature one at a time and report the gap. Then, if the exclusion of one feature leads to an improvement in our objective, we analyze removing all possible combinations of those features. Finally, the best-performing set of features is selected. Note that since the learning process can be sensitive to the scale of input features, we normalize feature variables such that they take continuous values between zero and one. Following this procedure, we selected feature set 20 from the set of available features.

Table 3: Statistical analysis of features.

	DTR	CH	TC	DTC	LOC	ETA	CTR	LDT	TS	Gap [%]
1	•	•	•	•	•	•	•	•	•	-
2		•	•	•	•	•	•	•	•	3.79
3	•		•	•	•	•	•	•	•	-1.16
4	•	•		•	•	•	•	•	•	0.43
5	•	•	•		•	•	•	•	•	-0.96
6	•	•	•	•		•	•	•	•	-3.31
7	•	•	•	•	•		•	•	•	-0.77
8	•	•	•	•	•	•		•	•	-4.53
9	•	•	•	•	•	•	•		•	-1.62
10	•	•	•	•	•	•	•	•		-0.55
11	•		•	•		•	•	•	•	-2.81
12	•		•	•	•	•		•	•	-1.76
13	•		•	•	•	•	•	•		-5.58
14	•	•	•	•		•		•	•	-1.08
15	•	•	•	•		•	•	•		-4.73
16	•	•	•	•	•	•		•		-5.79
17	•		•	•		•		•	•	-3.33
18	•		•	•		•	•	•		-3.17
19	•		•	•	•	•		•		-1.56
20	•		•	•		•		•		**-6.69**
21	•	•	•	•		•		•		-4.09

(•) Feature included in analysis

4 COMPUTATIONAL EXPERIMENTS

In Section 4.1, we discuss the simulation model which has been designed for this study. We show the training behavior of the proposed solution method in Section 4.2. Finally, we conduct a sensitivity analysis in Section 4.3 and highlight the performance of the proposed method on industry-sized instances.

4.1 System Description

The layout used in our study is shown in Figure 1. This layout is of the production facility of our industry partner KMWE. Each blue square, marked as P/D, represents a pickup and delivery node, i.e., a transport request always originates and concludes at one of these nodes. Each green square, marked as C, represents a charging node. There are 61 P/D nodes and 18 charging nodes in this layout.

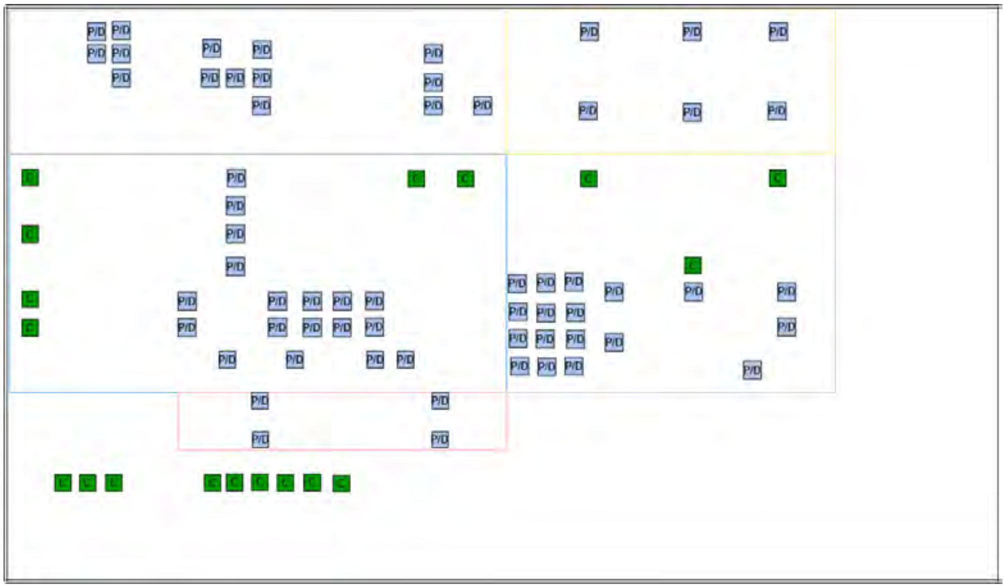


Figure 1: Layout of our industry partner.

Transport requests randomly arrive over time and between randomly generated pick-up and delivery nodes in the layout, and are categorized as low, medium, or high priority. High-priority requests are production critical and incur higher tardiness costs than medium and low-priority requests on being delivered late. Low-priority, medium-priority, and high-priority requests incur a penalty of \$0.01, \$0.02, and \$0.03 per second, respectively if they are delivered after the latest delivery time. A new transport request r has a capability requirement set $A_r^{\mathscr{R}} \in \{\{\text{A, B}\}, \{\text{B, C}\}, \{\text{C}\}, \{\text{C,D}\}, \{\text{E}\}\}$ and a tardiness cost, $c_k^{\mathscr{V}} \in [\$0.01, \$0.1]$.

The AGV fleet consists of three types of AGVs with different capabilities. For example, one particular type of AGV has capabilities {A, B, C} in which A represents the capability to lift a load, B to handle heavy loads, and C to handle light loads. Capabilities D and E mean the ability to tow loads and to handle loads with a robotic arm mounted on the AGV, respectively. Therefore, when a request arrives with capability requirements {A, B}, i.e., a heavy load that needs to be lifted, the dispatcher must assign it to an AGV that contains these capabilities. The AGVs incur a travel cost of \$0.01, \$0.02, \$0.03 per second which are representative of their real-life operational and maintenance costs, but their exact values have been modified due to confidentiality. The data used in this study for the AGV fleet is shared in Table 4.

Table 4: AGV fleet charecteristics.

Type	Speed (m/s)	$r_k^{\mathscr{V}}$ (%/s)	$d_k^{\mathscr{V}}$ (%/s)	$\underline{b}_k^{\mathscr{V}}$ (%)	$A_k^{\mathscr{V}}$	$c_k^{\mathscr{V}}$ (\$/s)
1	1	0.02	0.01	20	A, B, C	0.05
2	1.5	0.02	0.01	20	B, C, D	0.01
3	2	0.02	0.01	20	C, D, E	0.1

The simulation model is a continuous-time finite-horizon model with a decision moment at each new request arrival and is based on inputs from our industry partner at BIC. The simulation model captures varying requests arrival scenarios, AGV fleet size scenarios, and time window tightness scenarios. A given replication of the simulation model starts at 7:00 am, corresponding to the start of a working day, and lasts for a total simulation duration of 24 hours. The number of AGVs, $|V|$ may vary in $\{3,6,12\}$, and are located randomly at one of the charging nodes, with an initial charge drawn uniformly from [$\underline{b}_k^{\mathscr{V}}$, 100], at the start of the simulation. The number of requests $|R| \in \{450, 900, 1800\}$ represent *not-busy, typical, and busy* arrival scenarios (the requests are first generated from a Poisson process and then the specified number of requests are sampled randomly). At each dispatching moment (i.e., the arrival of a new request), the simulation model provides a list of features (described in Section 3.3), representing the state of all AGVs in the fleet at that moment to our method and receives an assignment decision from the LDR.

4.2 Training Behavior

Figure 2 highlights a typical training curve observed during our experiments. We let μ denote the cost associated with the LDR. We report the objective of each training iteration as well as a moving average over the last 100 iterations. It can be seen that the costs go down faster during the initial phases of our training. After 600 training iterations, the learning becomes relatively stable with almost no further reductions in costs. The proposed method was trained for 1000 training iterations and further validated on 100 iterations. We note that 100 iterations are sufficient to obtain a width of around 2-5%

The training time with a population size of 8 was about 13 hours, however, the training time can be reduced by using multiprocessing for simulation evaluations. With 8 parallel cores, one for each individual in the population, the training time was recorded to be around 3 hours on a computer with AMD Ryzen 3970X CPU @ 4.50GHz CPU and 128GB RAM.

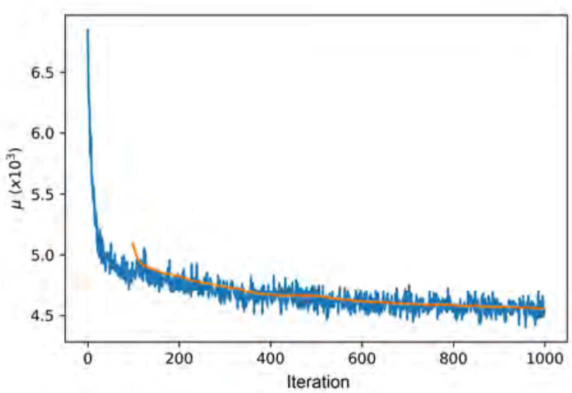


Figure 2: Training curve observed during experiments along with a rolling average over 100 training iterations.

An AGV-specific LDR is an efficient approach where each AGV in the fleet could have a distinct role, or, could prioritize certain features over other. Figure 3 highlights the progression of weights for each AGV in a given training experiment. Weight values become more stable as the training progresses, indicating that each AGV has developed a certain priority for each available feature. Note that each AGV has different weights for each feature indicating that each AGV could have a different purpose in the fleet, i.e., one could give more priority to its distance from the transport request and thus prioritizing travel costs, another could prioritize its distance to the center thus prioritizing serviceability. Some AGVs could have weights closer to zero for a certain feature, signifying that it is not a contributor to its bid, while another AGV might have a significant weight value for the same feature. We would like to note that in Figure 3, AGV 1 and AGV 2 are of type 1, AGV 3 and AGV 4 of type 2, and AGV 5 and AGV 6 of type 3 (AGV types are given in Table 4). It is interesting to see that AGVs of the same type learn to prioritize different feature values, thus playing a unique role in the system.

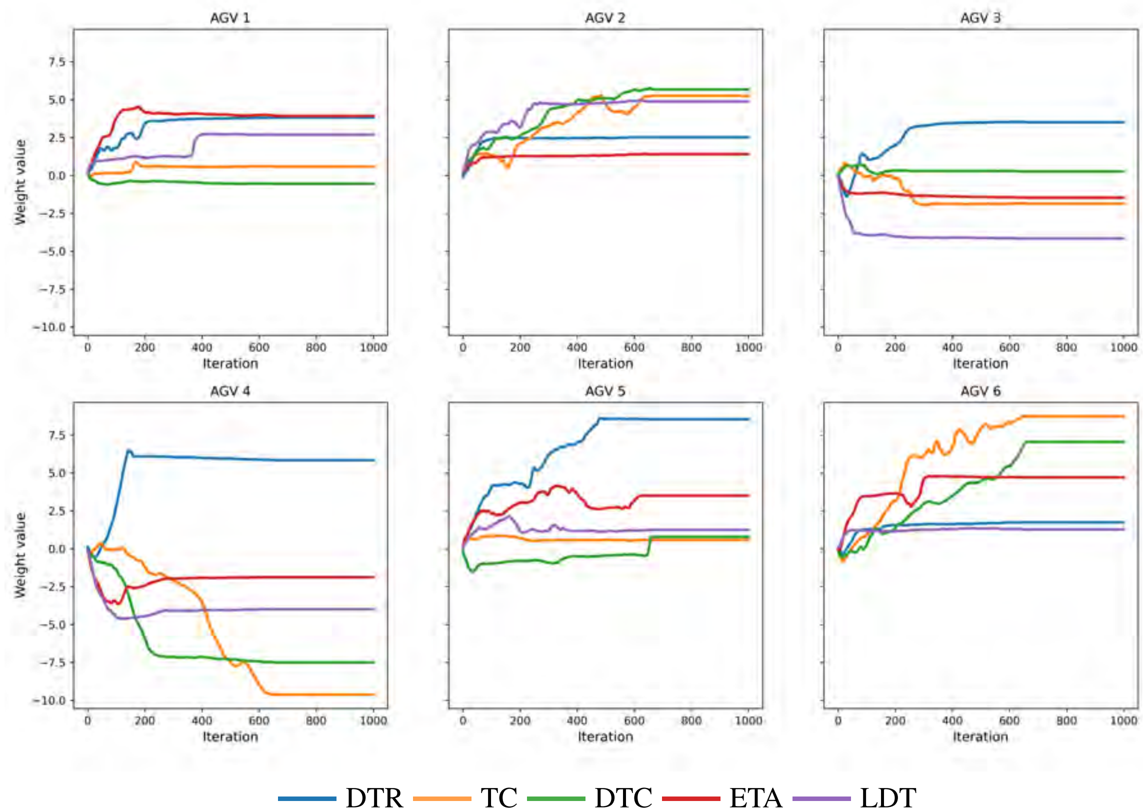


Figure 3: Development of weights for each AGV in the fleet during the training phase.

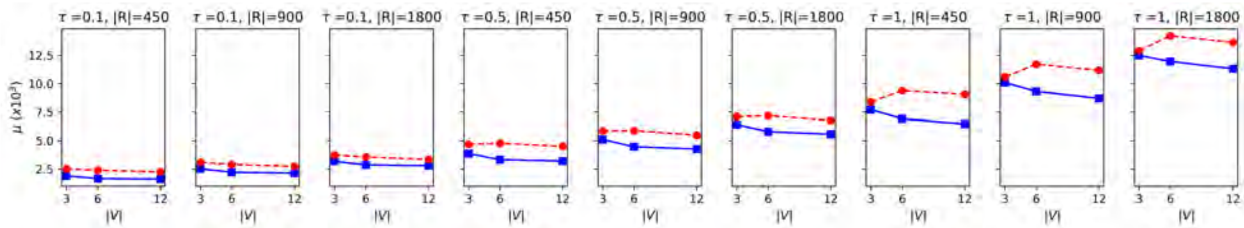
4.3 Sensitivity Analysis

We present the experiments for evaluating the proposed LDR method with a benchmark heuristic used in practice referred to as Earliest Due Date based BIDding (EDDBID). Details of this benchmark heuristic can be found in Singh et al. (2022). In short, in EDDBID a transport request is sent to the dispatcher when the material to be transported is ready at its pickup location. This rule makes use of the earliest due dates to prioritize requests for assignments on eligible idle AGVs. An AGV is considered idle when it has a sufficient charge ($b_k^{\mathscr{V}} \geq \underline{b}_k^{\mathscr{V}}$) and is not assigned to any request. The dispatcher then selects the idle AGV that is closest to the source node of that request. Note that due to the dynamic setting of this study, this rule is equivalent to assigning the nearest idle AGV when the transport request arrives.

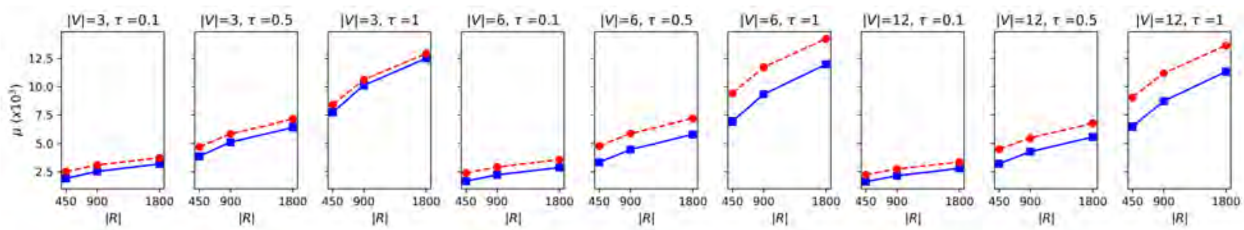
In this section, we highlight the performance of our proposed method on industry-sized instances and provide managerial insights by comparing it with EDDBID. We introduce τ as the probability of having tight time windows. The tightness of a time-window is measured by the difference between the earliest pickup time and the latest delivery time, and a value of τ, e.g., equal to 0.5, indicates that transport requests have a 50% chance of having tight time-windows, where the $\tau \in [0,1]$, i.e., with a probability τ, the latest delivery time of a request r is set as, $l_r^{\mathscr{R}} \leftarrow e_r^{\mathscr{R}} + (d_{max}/v_{min}) \times (0.25 + 0.25 \times y^{(0,1]})$, and otherwise, it is set as $l_r^{\mathscr{R}} \leftarrow e_r^{\mathscr{R}} + (d_{max}/v_{min}) \times (1 + 0.25 \times y^{(0,1]})$, where d_{max} is the maximum distance in the layout, v_{min} is the minimum speed of the AGVs, and $y^{(a,b)}$ denotes a random number between a and b.

We conduct a sensitivity analysis to investigate the effect of varying fleet size, request arrival rates, and time window tightness probabilities. We vary $|V| \in \{3,6,12\}$, $|R| \in \{450,900,1800\}$, and $\tau \in \{0.1,0.5,1.0\}$ to investigate 27 scenarios. The average costs obtained by LDR and EDDBID are summarized in Table 5. A negative gap percentage represents an improvement (savings in costs) over current practice, i.e., over EDDBID. Figure 4 shows trends observed across those scenarios.

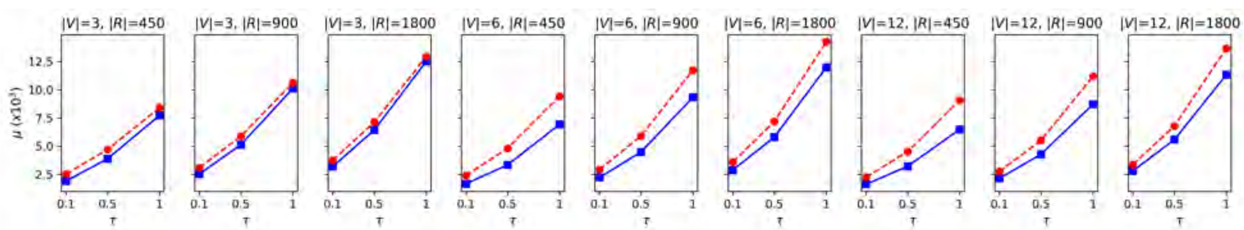
Figure 4(a) shows the effect of varying the fleet size under various request arrival scenarios on average costs. It can be observed that the performance gaps between LDR and EDDBID increase as the requests' time windows become tighter. However, the average costs tend to go down when increasing the fleet size due to an increase in the availability of capable vehicles and, as a result, increasing serviceability. Figure 4(b), under varying $|R|$, shows that the LDR outperforms EDDBID in all scenarios, with the biggest gap in scenarios where the requests arrive with relatively tighter time windows. In Figure 4(c), we see that costs go steeply upward when time window tightness increases. However, the rate of increase in EDDBID is more when compared with LDR.



(a) Average costs when varying $|V|$



(b) Average costs when varying $|R|$



(c) Average costs when varying τ

EDDBID LDR

Figure 4: Trends from numerical analysis on the heterogeneous fleet.

In summary, we observe an average savings of about 19% across all scenarios. Furthermore, the savings are pronounced when the available fleet size is large and when requests' time windows are tighter, which is also evident from the results in Table 5. Having a larger fleet size allows AGVs to assume distinct roles allowing them to utilize multiple temporal and spatial information largely ignored by other dispatching rules. Instances with tighter time windows have more potential for cost savings since, with loose windows, the benefits of using look-ahead information diminish.

Table 5: Sensitivity analysis on industry-sized instances.

#	$\|V\|$	τ	$\|R\|$	LDR	EDDBID	Gap (%)	#	$\|V\|$	τ	$\|R\|$	LDR	EDDBID	Gap (%)
1	3	0.1	450	3185.25	3740.81	-14.85	15	6	0.5	1800	9312.82	11720.34	-20.54
2	3	0.1	900	6417.95	7108.32	-9.71	16	6	1	450	1661.23	2399.89	-30.78
3	3	0.1	1800	12494.96	12917.3	-3.27	17	6	1	900	3336.96	4788.17	-30.31
4	3	0.5	450	2526.36	3094.87	-18.37	18	6	1	1800	6922.37	9400.27	-26.36
5	3	0.5	900	5119.02	5868.51	-12.77	19	12	0.1	450	2796.07	3357.33	-16.72
6	3	0.5	1800	10096.26	10599.84	-4.75	20	12	0.1	900	5578.78	6784.19	-17.77
7	3	1	450	1904.67	2510.83	-24.14	21	12	0.1	1800	11317.68	13647.4	-17.07
8	3	1	900	3870.56	4693.09	-17.53	22	12	0.5	450	2146.77	2746.04	-21.82
9	3	1	1800	7715.3	8392.9	-8.07	23	12	0.5	900	4278.08	5504.61	-22.28
10	6	0.1	450	2873.96	3573.09	-19.57	24	12	0.5	1800	8701.22	11178.75	-22.16
11	6	0.1	900	5797.87	7184.49	-19.30	25	12	1	450	1618.61	2252.24	-28.13
12	6	0.1	1800	11966.52	14233.81	-15.93	26	12	1	900	3212.72	4507.42	-28.72
13	6	0.5	450	2219.36	2913.99	-23.84	27	12	1	1800	6472.98	9059.43	-28.55
14	6	0.5	900	4470.3	5886.44	-24.05							

5 CONCLUSIONS

We study the problem of dispatching single-load heterogeneous AGV fleet using AGV-specific linear dispatching rules. These linear functions are learned by using the data collected from the discrete-event simulation of the system and can account for heterogeneity in AGVs and transport requests, thus utilizing spatial and temporal information to achieve an average cost savings of 19% on industry-sized instances. The proposed solution method relies on a learning scheme that is easily explainable and the resulting linear dispatching rule can be applied in practice with little effort. The proposed method also facilitates better utilization of the available fleet thereby reducing the costs at an operational level. An interesting future research direction is to consider non-linear dispatching rules and learning them with the help of neural networks.

ACKNOWLEDGEMENT

This research is a part of the Advanced Manufacturing Logistics program jointly funded by KMWE, IJssel Technologie, Rubix, TU/e, Erasmus, and Provincie Noord Brabant. We are grateful for their support. We would also like to thank Mr. Derek van de Wiel for their support in this study.

REFERENCES

Bilge, U., G. Esenduran, N. Varol, Z. Öztürk, B. Aydin, and A. Alp. 2006. “Multi-Attribute Responsive Dispatching Strategies for Automated Guided Vehicles”. *International Journal of Production Economics* 100(1):65–75.

Brainport Industries Campus 2020. “The Evolution of the High-Tech Manufacturing Industry”. https://www.brainportindustriescampus.com/en/innovating/factory-of-the-future. Accessed 20 March 2023.

Confessore, G., M. Fabiano, and G. Liotta. 2013. “A Network Flow Based Heuristic Approach for Optimising AGV Movements”. *Journal of Intelligent Manufacturing* 24(2):405–419.

De Ryck, M., M. Versteyhe, and F. Debrouwere. 2020. “Automated Guided Vehicle Systems, State-of-the-Art Control Algorithms and Techniques”. *Journal of Manufacturing Systems* 54:152–173.

Didden, J. B., Q.-V. Dang, and I. J. Adan. 2021. “A Semi-Decentralized Control Architecture for High-Mix-Low-Volume Factories in Industry 4.0”. *Manufacturing Letters* 30:11–14.

Didden, J. B. H. C., Q. V. Dang, and I. J. B. F. Adan. 2023. “Decentralized Learning Multi-Agent System for Complex Machine Shop Scheduling”. *Journal of Manufacturing Systems* 67:338–360.

Guan, X., and X. Dai. 2009. “Deadlock-Free Multi-Attribute Dispatching Method for AGV Systems”. *International Journal of Advanced Manufacturing Technology* 45:603–615.

Holthaus, O., and C. Rajendran. 1997. “Efficient Dispatching Rules for Scheduling in a Job Shop”. *International Journal of Production Economics* 48(1):87–105.

Jeong, B. H., and S. U. Randhawa. 2001. "A Multi-Attribute Dispatching Rule for Automated Guided Vehicle Systems". *International Journal of Production Research* 39(13):2817–2832.

Keskin, M., and B. Çatay. 2016. "Partial Recharge Strategies for the Electric Vehicle Routing Problem with Time Windows". *Transportation Research Part C: Emerging Technologies* 65:111–127.

Kim, N., S. Barde, K. Bae, and H. Shin. 2021. "Learning Per-Machine Linear Dispatching Rule for Heterogeneous Multi-Machines Control". *International Journal of Production Research* 0(0):1–21.

Kingma, D. P., and J. Ba. 2015. "Adam: A Method for Stochastic Optimization". *CoRR* abs/1412.6980.

Le-Anh, T., and M. B. De Koster. 2005. "On-line Dispatching Rules for Vehicle-Based Internal Transport Systems". *International Journal of Production Research* 43(8):1711–1728.

Le-Anh, T., and M. B. De Koster. 2006. "A Review of Design and Control of Automated Guided Vehicle Systems". *European Journal of Operational Research* 171(1):1–23.

Riazi, S., T. Diding, P. Falkman, K. Bengtsson, and B. Lennartson. 2019. "Scheduling and Routing of AGVs for Large-Scale Flexible Manufacturing Systems". *IEEE International Conference on Automation Science and Engineering* 2019-Augus:891–896.

Singh, N., Q.-V. Dang, A. Akcay, I. Adan, and T. Martagan. 2022. "A Matheuristic for AGV Scheduling with Battery Constraints". *European Journal of Operational Research* 298(3):855–873.

Tay, J. C., and N. B. Ho. 2008. "Evolving Dispatching Rules Using Genetic Programming for Solving Multi-Objective Flexible Job-Shop Problems". *Computers and Industrial Engineering* 54(3):453–473.

Vis, I. F. 2006. "Survey of Research in the Design and Control of Automated Guided Vehicle Systems". *European Journal of Operational Research* 170(3):677–709.

Vivaldini, K., L. F. Rocha, N. J. Martarelli, M. Becker, and A. P. Moreira. 2016. "Integrated Tasks Assignment and Routing for the Estimation of the Optimal Number of AGVs". *The International Journal of Advanced Manufacturing Technology* 82(1):719–736.

Wang, J., Y. Zhang, Y. Liu, and N. Wu. 2019. "Multiagent and Bargaining-Game-Based Real-Time Scheduling for Internet of Things-Enabled Flexible Job Shop". *IEEE Internet of Things Journal* 6(2):2518–2531.

Zhao, M., and Y. Lu. 2019. "A Heuristic Approach for a Real-World Electric Vehicle Routing Problem". *Algorithms* 12(2):45.

AUTHOR BIOGRAPHIES

NITISH SINGH is a Ph.D. candidate in the School of Industrial Engineering at Eindhoven University of Technology. His research interests include simulation-based optimization, and data-driven operations management, focusing on applications in supply chain planning. His email address is n.singh1@tue.nl. His website is nitishsingh.com.

JEROEN B.H.C. DIDDEN is a Ph.D. candidate in the School of Industrial Engineering at the Eindhoven University of Technology (TU/e). His current research focuses on the application of Multi-Agent System to solve complex scheduling problems in High-Mix-Low-Volume production environments. His e-mail address is j.b.h.c.didden@tue.nl.

ALP AKCAY is an Associate Professor in the School of Industrial Engineering at Eindhoven University of Technology. He received his Ph.D. in Operations Management and Manufacturing from Carnegie Mellon University. His current research interest is data-driven decision making with applications in manufacturing and maintenance. His email address is a.e.akcay@tue.nl.

TUGCE MARTAGAN is an Associate Professor in the School of Industrial Engineering at Eindhoven University of Technology. She received her Ph.D. from the University of Wisconsin-Madison. Her research interests include stochastic modeling and optimization of manufacturing systems. Her research has been recognized through various awards, including the Franz Edelman Laureate in 2022, the first prize in the POMS Applied Research Challenge in 2016, and honorable mention in the INFORMS George B. Dantzig dissertation award in 2016. Her email address is t.g.martagan@tue.nl.

IVO J.B.F. ADAN is a Professor in the School of Industrial Engineering at Eindhoven University of Technology. His current research interests are in the modeling and design of manufacturing systems, warehousing systems and transportation systems, and more specifically, in the analysis of multi-dimensional Markov processes and queueing models. His email address is i.adan@tue.nl.