



## A new knowledge-guided multi-objective optimisation for the multi-AGV dispatching problem in dynamic production environments

Lei Liu, Ting Qu, Matthias Thürer, Lin Ma, Zhongfei Zhang & Mingze Yuan

To cite this article: Lei Liu, Ting Qu, Matthias Thürer, Lin Ma, Zhongfei Zhang & Mingze Yuan (2022): A new knowledge-guided multi-objective optimisation for the multi-AGV dispatching problem in dynamic production environments, International Journal of Production Research, DOI: [10.1080/00207543.2022.2122619](https://doi.org/10.1080/00207543.2022.2122619)

To link to this article: <https://doi.org/10.1080/00207543.2022.2122619>



View supplementary material [↗](#)



Published online: 23 Sep 2022.



Submit your article to this journal [↗](#)



View related articles [↗](#)



View Crossmark data [↗](#)

RESEARCH ARTICLE



# A new knowledge-guided multi-objective optimisation for the multi-AGV dispatching problem in dynamic production environments

Lei Liu<sup>a</sup>, Ting Qu<sup>b</sup>, Matthias Thürer<sup>b</sup>, Lin Ma<sup>a</sup>, Zhongfei Zhang<sup>a</sup> and Mingze Yuan<sup>a</sup>

<sup>a</sup>School of Management, Jinan University, Guangzhou, People's Republic of China; <sup>b</sup>School of Intelligent Systems Science and Engineering, Jinan University (Zhuhai Campus), Zhuhai, People's Republic of China

## ABSTRACT

The efficiency of material supply for workstations using Automatic Guided Vehicles (AGVs) is largely determined by the performance of the AGV dispatching scheme. This paper proposes a new solution approach for the AGV dispatching problem (AGVDP) for material replenishment in a general manufacturing workshop where workstations are in a matrix layout, and where uncertainty in replenishment time of workstations and stochastic unloading efficiencies of AGVs are dynamic contextual factors. We first extend the literature proposing a mixed integer optimisation model with a delivery satisfaction soft constraint of material orders and two objectives: transportation costs and delivery time deviation. We then develop a new knowledge-guided estimation of distribution algorithm with delivery satisfaction evaluation for solving the model. Our algorithm fuses three knowledge-guided strategies to enhance optimisation capabilities at its respective execution stages. Comprehensive numerical experiments with instances built from a real-world scenario validate the proposed model and algorithm. Results demonstrate that the new algorithm outperforms three popular multi-objective evolutionary algorithms, a discrete version of a recent multi-objective particle swarm optimisation, and a multi-objective estimation of distribution algorithm. Findings of this work provide major implications for workshop management and algorithm design.

## ARTICLE HISTORY

Received 31 December 2021  
Accepted 28 August 2022

## KEYWORDS

AGVs dispatching; workshop logistics; multi-objective optimisation; metaheuristics; intralogistics

## 1. Introduction

Any production system that produces physical goods needs to transport material. The larger and more complex the layout, the larger the impact of material transportation on overall production system performance (Zou et al. 2020). Automatic Guided Vehicles (AGVs) consequently gain in popularity, because they provide high efficiency and quick response (Yao et al. 2020). Motivated by this context and observations in a major domestic appliance and air-conditioner producer in Guangdong (China) that sees intralogistics as a main constraint, this paper provides a new solution approach for a multiple AGVs dispatching problem for materials replenishment in a general manufacturing workshop with a matrix layout.

In the considered matrix manufacturing workshop, a fleet of homogeneous AGVs is responsible for delivering production materials to workstations according to a schedule created by a higher-level planning system. The AGVs scheduling problem (AGVSP) can then be divided into two related subproblems (Malopolski 2018;

Li et al. 2019): a dispatching problem, concerning priority sequencing and the assignment of orders (throughout this paper, the terms orders, material orders and material delivery orders are used interchangeably) to AGVs, and a routing problem, concerning the selection of specific paths to be taken by AGVs given the order delivery sequence.

Although there exists a broad literature on the AGVDP (Li et al. 2019; Zou et al. 2020; Zhang et al. 2022), this literature does not take disturbances and dynamics into account. However, in practice there may be significant variability in terms of factors that determine the AGV schedule, such as accuracy of the schedule created by the higher-level planning system and unloading efficiencies. Variability in the replenishment times and AGV unloading efficiency are the most common factors that have a direct impact on the punctuality of AGV's delivering material orders based on our literature analysis and practical observation. They make it difficult for AGVs to supply material for demand workstations just at the scheduled times. Note that stochastic machining times

and path conflicts can also be considered part of stochastic replenishment and unloading times if they are independent. In response, this study proposes a new solution approach to the AGVDP that takes variability in these two factors into account. To accommodate uncertainty in the replenishment times calculated by the higher-level planning system, it adopts a fuzzy soft time window for each workstation to describe its accepted change range of replenishment times, and to track timeliness of material order delivery. To accommodate fluctuations in AGV unloading efficiency, a heuristic method for the estimation of AGV unloading duration is incorporated, which is based on real-world operation scenarios and a probability computing method.

Main material transportation costs include configuration cost of AGVs, AGVs' travel distance cost, and penalty cost for violating a delivery time window. These three costs are all indicators related to the economic costs of material transportation using AGVs. They constitute the total transportation costs. They can however not be represented by only one indicator or objective because they derive from quite different operational factors. In addition, the material delivery punctuality is another key indicator that cannot be easily ignored. The AGVDP under consideration can consequently be classified as a multi-objective optimisation problem, which is quite different from the single objective optimisation schemes considered in the previous literature (Fazlollahabadi and Hassanli 2018; Li et al. 2019; Zou et al. 2020; Zhang et al. 2022). We additionally extend the existing literature by considering a solution satisfaction metric on the punctuality of AGV delivering orders as a soft constraint in our problem.

The AGVDP can be considered a variant of the vehicle routing problem (VRP). It is consequently a NP-hard problem. In practice, most manufacturing enterprises adopt basic dispatching rules to schedule AGVs, such as First Come First Served (FCFS). But meta-heuristics are generally able to provide better performance than dispatching rules. For example, the Estimation of Distribution Algorithm (EDA; Larrañaga and Lozano 2001), which offers a convenient means to integrate the knowledge from experts, optimisation processes data, and historical solutions data for guiding the evolutionary optimisation of algorithms. This omits the design step of complex evolutionary operators and provides better flexibility towards different problem situations compared with other meta-heuristics. Nonetheless, there is currently no study on the application of EDA for the AGVDP in the literature (Wang et al. 2013; Duan et al. 2016; Zhang et al. 2021).

In this paper, we first propose a mixed integer optimisation model with a delivery satisfaction soft constraint

of material orders and two objectives: transportation costs and delivery time deviation. We then outline a new knowledge-guided EDA for solving the model. The effectiveness of the proposed model and algorithm is confirmed through comparison experiments with instances built from a real-world scenario. Our problem model can obtain more robust solution schemes while the multi-objective optimisation solution approach adopted in this study avoids the need to determine proper weights for multiple indicators. This facilitates application in practice. It can also provide a set of options for managers to make adaptive decisions. Meanwhile, the proposed algorithm outperforms related popular algorithms.

Previous literature related to this work is reviewed next in Section 2. The definition of our problem and the model are then presented in Section 3, before the proposed EDA is described in Section 4. Section 5 presents and analyzes the experimental results. Final conclusions are provided in Section 6 together with limitations and future research directions.

## 2. Literature review

The two related subproblems of the AGV scheduling problem, the AGVDP and the AGV routing problem, can be investigated simultaneously (Nishida and Nishi 2022; Wang and Zeng 2022) or in isolation (Murakami 2020) according to different problem settings and solution methods (Malopolski 2018). Fazlollahabadi and Saidi-Mehrabadi (2015) further highlight that there exist different methodologies, which can be categorised into exact methods, such as mathematical programming and analytical models, simulation studies, heuristics and metaheuristic, and artificial intelligence (AI) based approaches. In Section 2.1, we next review the literature relevant to our study following this categorisation. A discussion of the literature is then presented in Section 2.2.

### 2.1. Related literature

The column generation algorithm as an exact method can be deployed to create a tasks-to-AGVs assignment scheme if there are very few tasks and if AGVs operate in a static routing context (Krishnamurthy, Batta, and Karwan 1993). For solving the problem instance with more AGVs or tasks, Desaulniers et al. (2003) further developed an exact method combining a column generation, a greedy search heuristic and a branch and cut algorithm. For dynamic manufacturing systems, Sinriech and Kotlarski (2002) designed a dynamic algorithm to dispatch multiple-load AGVs in a single loop that

minimises transfer times of tasks and the number of loops travelled by the AGV. For the simultaneous solving of the dispatching and routing problem of multi-AGV, some exact methods, such as the bi-level decomposition algorithm and the network simplex-based algorithm, were also developed to minimise the total weighted tardiness of tasks or the total travel distances of vehicles (Nishi, Hiranaka, and Grossmann 2011; Fazlollahtabar and Hassanli 2018). But exact methods are only suitable for small-scale multi-AGV dispatching problems with simple operation characteristics and single optimisation objective. More complex real-world problems are typically NP-hard, and heuristics or metaheuristics need to be applied (Ahmadi-Javid and Ardestani-Jaafari 2021; Deng et al. 2022; Singh et al. 2022).

Fazlollahtabar, Saidi-Mehrabad, and Balakrishnan (2015) solved the multiple AGVs dispatching problem using an integrated heuristic algorithm that minimises the penalised earliness and tardiness. For manufacturing system with capacitated AGVs and buffers, Miyamoto and Inoue (2016) proposed a heuristics that combines local and random searches to minimise the total weighted sum of the travel distances of vehicles and delays for delivery points. Meanwhile, Bae and Chung (2017) focused on the optimal number of AGVs required in a manufacturing system. They designed a primal-dual heuristic to tackle the heterogeneous AGVs' routing problem to minimise the sum of AGVs travel costs. Riazi et al. (2019) developed a new heuristic based on Benders decomposition for the multiple AGVs scheduling problem in large-scale flexible manufacturing systems. More recently, Zhang et al. (2022) proposed an improved iterated greedy algorithm to solve the AGVDP, minimising total transportation costs in a matrix manufacturing workshop.

In terms of metaheuristics, a hybrid genetic algorithm was used by Umar et al. (2015) to solve a multi-objective optimisation problem for the integrated scheduling of orders and AGVs in a flexible manufacturing systems. To address coordination issues between dispatching and routing aspects of multiple AGVs, Lee et al. (2020) transformed the problem into a winner determination problem, and then proposed a improved genetic algorithm with knowledge based operators to solve it. Qin et al. (2021) developed an improved parallel multiple-objective genetic algorithm to simultaneously address the problem of AGV scheduling and lot targeting for stochastic wafer production environments. Apart from genetic algorithms, the literature also used the sheep flock heredity algorithm (Subbiah, Nageswara Rao, and Narayana Rao 2009), particle swarm optimisation (Mousavi et al. 2017), harmony search (Li et al. 2019) and ant colony optimisation

(Xu et al. 2019). Meanwhile, Zou et al. (2020) presented a discrete artificial bee colony (DABC) algorithm for solving the AGVDP in a generalised workshop with a matrix layout.

Finally, simulation or AI based approaches tend to focus on the optimisation of the routes of AGVs, for example to avoid AGVs collision and path congestion (Li et al. 2019; Hwang and Jang 2020; Sun and Li 2020; Hu et al. 2021; Zhou and He 2021; Tsolakis et al. 2022).

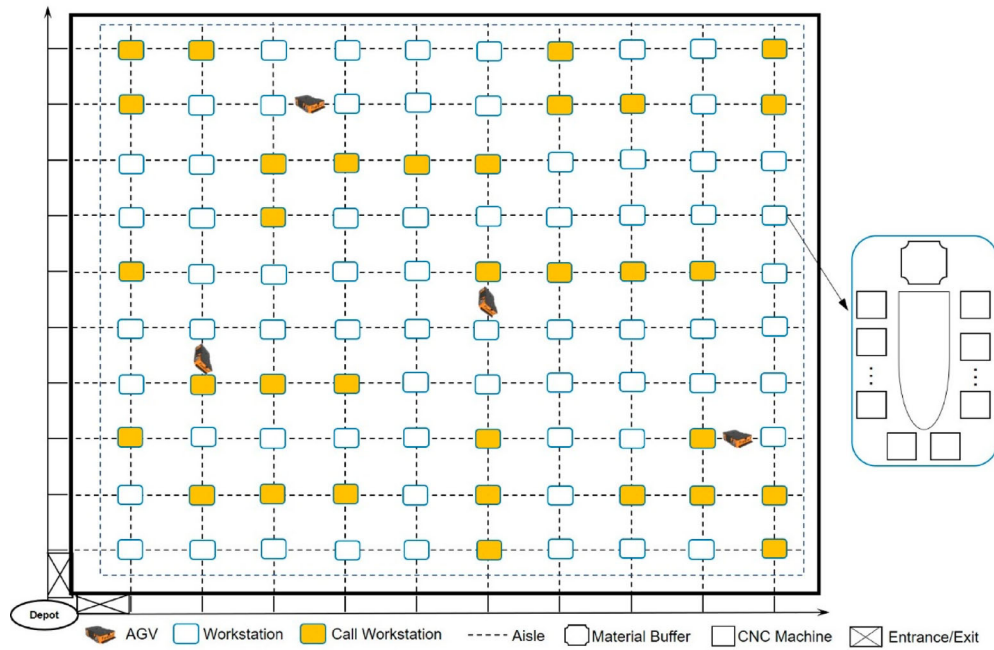
## 2.2. Discussion of the literature

Given its importance, there exists a large body of literature on the AGVDP in flexible manufacturing systems. However, most of this literature focusses on environments with only few functionally similar workstations. Much less work exists in the context of generalised workshops with many multifunctional workstations. To best of our knowledge, Li et al. (2019) was the first to address the AGVDP in a workshop with matrix layout, using an harmony search algorithm. However, the authors did not consider disturbances and dynamics common to real-life shops. The same criticism holds for Zou et al. (2020) and Zhang et al. (2022) who, although achieving excellent solution results, both adopt a static problem setting in terms of replenishment time and AGVs unloading efficiency. Moreover, existing literature integrates multiple indicators related to transportation costs and delivery punctuality for the AGVDP into a single optimisation objective by setting respective weights for these indicators. Determining the proper weight for each indicator is usually difficult for managers. There exists consequently a need to further study the AGVDP in generalised workshops with a matrix layout, proposing solutions that accommodate production system dynamics and that use multi-objective optimisation methods.

## 3. Problem description and formulation

### 3.1. Problem description

The company that triggered our study produces five different categories of air conditioners, involving over 100 different products. The production process can slightly differ for the different categories and products and might be adapted to specific customer orders. There are four departments, the mould injection, the condenser, the sheet metal, and the control production, which feed into a central assembly line. Part of the departments operate as general manufacturing workshops that produce a high variety of small batches. These workshops have a certain number of computer numerical control (CNC) machines grouped into different workstations by their



**Figure 1.** Layout of a matrix manufacturing workshop.

process characteristics, which brings more challenges for intralogistics operation and management than a dedicated workshop. The layout of workstations in the general manufacturing workshops often adopts a matrix layout (as shown in Figure 1).

Workstations continuously consume the materials in their material buffers. When the stock level drops to a replenishment threshold, a replenishing material order is sent to the manufacturing execution system of the workshop. Within a given production cycle, the manufacturing execution system collects the material requirement of each workstation as one material delivery order, schedules all material delivery orders into a material supply task, and then sends the task to the material supply centre for execution. Each production cycle is consequently composed of an initial dispatching stage and a delivery stage. At the dispatching stage the manufacturing execution system uses a dispatching rule to output a delivery schedule for existing orders and the material supply centre assigns the orders among available AGVs according to the schedule. Every AGV then executes the assigned routes at the delivery stage. After finishing assigned delivery orders, AGVs go back to the depot of the material supply centre.

Two important factors that significantly impact the outcome of the AGVDP are considered in this study: uncertainty in replenishment times and fluctuations in AGV unloading efficiency. The former can be due to CNC machine breakdown, rush orders or delivery date changes. The latter may occur when AGVs wait for

service at a workstation due to early arrival, the temporary occupation of material buffers or unavailable workers.

Our research problem can then be summarised as follows: At the dispatching stage of the production cycle, how to create a satisfactory material replenishment scheme that considers variability in replenishment and unloading time, and that minimises transportation cost and delivery time deviation. This problem can be regarded as a particular case of the Vehicle Routing Problem with Time Windows (VRPTW) since it has similar characteristics as the general VRPTW. However, solving our multi-objective optimisation problem considering dynamic factors is arguably more challenging than the general VRPTW with single optimisation objective. Without loss of generality, the following assumptions are made regarding our problem:

- AGVs refers to free-range AGVs with autonomous navigation function. This means the AGV can avoid obstacles and there is no collision or road conflict among AGVs.
- AGVs have limited load capacity and the demand of each material order is smaller than the AGV load capacity.
- The travelling velocity of AGVs is constant.
- There is no shutdown or malfunction for AGVs in the manufacturing workshop.
- The effect of material geometry characteristics on AGV loading quantity is ignored.



- The maximum number of material delivery orders each AGV can serve due to its battery capacity limitation is considered.
- The situation where there are not enough AGVs, and some delivery orders must wait for one available AGV is not considered.
- A workstation with a material delivery order is visited by exactly one AGV.
- AGVs should provide the specified materials to the workstation within an acceptable time window. It bears a penalty cost for early arrival, while late arrivals are forbidden.
- Workstation priority for unloading materials follows urgency.
- It is assumed that AGV unloading durations at workstations can be estimated and follow a known normal probability distribution.
- The AGV loading duration at the depot of the material supply centre is considered to have a negligible effect.

### 3.2. Problem formulation

Based on the above description, the considered problem can be formulated as a mathematical model using graph theory. The specific parameters and constants can be defined as follows.

- $i, j$ : The identity of material delivery orders.
- $k$ : The identity of current AGV.
- $L_i$ : The location of the workstation with order  $i$ .
- $x_i$ : The coordinate in x-direction of location  $L_i$ .
- $y_i$ : The coordinate in y-direction of location  $L_i$ .
- $n$ : The total number of orders.
- $w_i$ : The load weight of order  $i$ .
- $d_{ij}$ : The AGV travel distance between order  $i$  and  $j$ .
- $t_{ij}$ : The AGV travel time between order  $i$  and  $j$ .
- CT: A production cycle.
- $T_i^0$ : The generation time of order  $i$ .
- $T_i^s$ : The scheduled delivery time of order  $i$  that is equal to the sum of  $T_i^0$  and CT.
- $T_i^e$ : The earliest delivery time of order  $i$  in a specified delivery time window.
- $T_i^l$ : The latest delivery time of order  $i$  in a specified delivery time window.
- $T_0$ : The time when AGV departs from the depot.
- $\Delta T$ : The time duration of the dispatching stage of a production cycle.
- $N_s$ : The maximum number of orders an AGV can serve.

$v$ : The velocity of an AGV.

$C$ : The load capacity of an AGV.

$t_i^s$ : The AGV unloading duration at the workstation with order  $i$ , which is an estimable variable with known normal probability distribution.

$\mu_s$ : The average AGV unloading duration.

$\sigma_s$ : The estimated standard deviation of the AGV unloading duration.

$Z_\epsilon$ : The confidence factor  $Z_\epsilon = \Phi^{-1}(\epsilon)$ , where  $\epsilon$  denotes a confidence level and  $\Phi$  represents the standard normal distribution function.

$t_m$ : The consumption time per unit production material.

$g_m$ : The weight per unit production material.

$S$ : The total stock in the material buffer at each workstation.

$S_i^0$ : The remaining stock in the material buffer of workstation  $L_i$  at  $T_i^0$ .

$c_d$ : The unit cost of AGV travel distance.

$c_a$ : The configuration cost of each AGV.

$c_p$ : The unit penalty cost for early delivery.

$P_i$ : The penalty cost function for early delivery of order  $i$ .

$SA_i$ : The satisfaction level for the delivery on-time performance of order  $i$  measured by the fuzzy membership function value of the predicted order delivery time.

$\overline{SA}$ : The mean satisfaction level for the delivery on-time performance of all orders.

$\alpha$ : The time sensitivity coefficient for early delivery within the acceptable time window.

$\beta$ : The time sensitivity coefficient for delayed delivery within the acceptable time window.

The decision variables are:

$x_{ijk}$ : if path  $(i, j)$  is visited by AGV  $k$  then  $x_{ijk} = 1$ , and 0 otherwise.

$m$ : The number of AGVs used.

$T_i^r$ : The predicted delivery time of order  $i$ , referring to the AGV arrival time for a demand workstation predicted by the AGVs dispatching scheme.

Finally, in terms of equations, let  $G = V, E$  be an undirected graph, where  $V = 0, 1, 2, \dots, n$  is the set of vertices, and  $E = (i, j) | i, j \in V, i \neq j$  denotes the set of edges between each pair of vertices. Vertices  $i = 1, \dots, n$  represent order identities, whereas vertex 0 is the depot identity of the material supply centre. Note that each order is bound to a workstation. Material depot and assigned orders consequently represent the routing of the AGV.  $K = 1, \dots, m$  represents an identity set of AGVs to be dispatched. Each edge  $(i, j)$  denotes the travel path of an AGV from orders

(or depot)  $i$  to  $j$ , where the depot has the plane coordinates of (0,0). The travel distance and travel time of the AGV travelling this path ( $i, j$ ) can be calculated by equations (1) and (2), respectively.

$$d_{ij} = |x_i - x_j| + |y_i - y_j| \quad (1)$$

$$t_{ij} = \frac{d_{ij}}{v} \quad (2)$$

When the AGV travels from orders (or depot)  $i$  to  $j$ , the arrival time of order  $j$ , that is the predicted delivery time of order  $j$ , can be obtained by the following equation:

$$T_j^r = T_i^r + t_i^s + t_{ij}, \forall i \in V, j \in V \setminus \{0\}, i \neq j \quad (3)$$

In equation (3), the AGV unloading duration  $t_i^s$  can be estimated through equations (4) and (5). Both equations are based on the classical normal variable generation method proposed by Kinderman and Monahan (1977). This estimation method reflects that urgent material replenishments are executed directly, a common situation in practice, whereas if an AGV reaches the delivery point early, it must wait for unloading until workers are available.

$$\rho = \frac{4e^{-\frac{1}{2}}}{\sqrt{2}} \times \frac{(T_i^s - T_i^r)}{(T_i^l - T_i^e)} \times (1 - SA_i) \quad (4)$$

$$t_i^s = \mu_s + \lfloor \rho \cdot \sigma_s \cdot Z_\varepsilon \rfloor \quad (5)$$

Suppose that an AGV arrives at the workstation with order  $j$ , and then unloads the required materials. The quantity unloaded, namely the material load weight of order  $j$ , is obtained by equation (6).

$$w_j = \left[ (S - S_j^o) + \left\lceil \frac{(T_j^s - T_j^o)}{t_m} \right\rceil \right] \cdot g_m, \forall j \in V \setminus \{0\} \quad (6)$$

Given the uncertainty in terms of replenishment times and the stochastic AGV unloading duration, a fuzzy soft time window is proposed as  $[T_i^e, T_i^s, T_i^l]$  to track the timeliness of material order delivery. In addition, the fuzzy membership function of predicted order delivery time, denoted as  $SA_i(T_i^r)$  and given by Equation (7), is adopted to measure the satisfaction level for the delivery time performance of order  $i$  based on the proposed time window.

$$SA_i(T_i^r) = \begin{cases} \left( \frac{T_i^r - T_i^e}{T_i^s - T_i^e} \right)^\alpha + 0.01, & T_i^e \leq T_i^r < T_i^s \\ 1, & T_i^r \leq T_i^s \\ \left( \frac{T_i^l - T_i^r}{T_i^l - T_i^s} \right)^\beta + 0.01, & T_i^s < T_i^r \leq T_i^l \\ 0, & T_i^r < T_i^e \end{cases} \quad (7)$$

The parameter  $\alpha$  and  $\beta$  are time sensitivity coefficients for the predicted delivery time within the acceptable

time window. They were introduced to construct the fuzzy membership function with a nonlinear relationship between the satisfaction and the predicted delivery time. Generally, workstations have a higher time sensitivity to delayed material delivery compared to early material delivery because the delayed material delivery typically causes capacity loss, and consequently higher profit loss.

When an AGV arrives at the workstation with order  $i$  just at the scheduled delivery time, the function  $SA_i(T_i^r)$  outputs 100% satisfaction level of the workstation for the delivery time performance of order  $i$ . Within the time window  $[T_i^e, T_i^s, T_i^l]$ , the satisfaction level decreases with the increase of the deviation between the predicted delivery time and the scheduled delivery time. The satisfaction becomes 0 when the predicted delivery time of order  $i$  is smaller than the acceptable earliest delivery time of order  $i$ , resulting in the penalty cost for early order delivery defined by Equation (8).

$$P_i(T_i^r) = \begin{cases} c_p(T_i^e - T_i^r), & T_i^r < T_i^e \\ 0, & T_i^e \leq T_i^r \leq T_i^l \end{cases} \quad (8)$$

Delivering order  $i$  after the acceptable latest delivery time of order  $i$  is forbidden, since in this case the production process of the workstation would be interrupted.

Given the above definitions, our AGVDP problem can be modelled by a mixed integer programming (MIP) model with two optimisation objectives as follows:

$$\min F_1(i, j, k) = c_d \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} x_{ijk} d_{ij} + c_a \sum_{j=1}^n \sum_{k \in K} x_{0jk} + \sum_{i=1}^n P_i(T_i^r) \quad (9)$$

$$\min F_2(i, j, k) = \sum_{i=0}^n \sum_{j=1}^n \sum_{k \in K} x_{ijk} |T_j^s - T_j^r| \quad (10)$$

Subject to:

$$\sum_{k \in K} \sum_{i=0}^n x_{ijk} = 1, \forall j \in V \setminus \{0\} \quad (11)$$

$$\sum_{i=1}^n x_{i0k} = \sum_{j=1}^n x_{0jk} = 1, \forall k \in K \quad (12)$$

$$\sum_{i=0}^n \sum_{j=1}^n x_{ijk} w_j \leq C, \forall k \in K \quad (13)$$

$$T_0 < T_i^r \leq T_i^l, \forall i \in V \setminus \{0\} \quad (14)$$

$$n/N_s \leq m \leq n/2 \quad (15)$$

$$\overline{SA} = \frac{1}{n} \sum_{i=1}^n SA_i(T_i^r) > 0 \quad (16)$$

$$x_{ijk} \in \{0, 1\}, \forall i, j \in V, \forall k \in K \quad (17)$$

$$x_{ijk} = 0, \forall i, j \in V \text{ and } i = j \quad (18)$$

$$T_i^r = T_0, i = 0 \quad (19)$$

In our model, the first objective (9) is to minimise the total transportation cost, consisting of travel cost of AGVs, configuration cost of AGVs and penalty cost for early order delivery. It is worth noting that if the value of  $c_a$  is a large constant, a reduction to one AGV is always preferred to the decrease of other costs. The second optimisation objective (10) is to minimise the total deviation of orders' predicted delivery times from the scheduled delivery times.

In terms of constraints, a workstation must be visited exactly once by one AGV (constraint (11)), each AGV travel route should start and end at the depot (constraint (12)), and the load capacity of an AGV should not be exceeded (constraint (13)). The constraint on order delivery time is represented by equation (14). Constraint (15) ensures that the number of AGVs used is in a reasonable range. The delivery satisfaction constraint is given by equation (16). It is considered a soft constraint, which allows for improving the punctuality of order delivery and decreasing the penalty cost of early delivery. Finally, constraints (17–19) give the restrictions on the values of the decision variables.

#### 4. Algorithm design

Our AGVDP is a NP-hard problem. It is consequently formulated as a MIP model with several nonlinear terms and two solution objectives, which prohibits the use of standard math solvers. There exists no specific algorithm developed for our problem and a new efficient algorithm needs to be developed. Motivated by the successful application of knowledge-based search optimisation strategies in recent related studies (Bandaru, Ng, and Deb 2017; Barba-Gonzalez et al. 2021), in this section a knowledge guided multi-objective estimation of distribution algorithm (KEDA) with delivery satisfaction evaluation (KEDAS) is proposed for solving the considered AGVDP. The proposed KEDA inherits the framework of the basic estimation of distribution algorithm (EDA). It integrates three types of knowledge derived from dispatching rules, pareto solutions quality, and order delivery satisfaction for candidate solutions, to guide its evolutionary optimisation towards superior solutions.

##### 4.1. Estimation of distribution algorithm (EDA)

EDA provides a unique paradigm in the field of evolutionary optimisation based on statistical and probability theory (Wang et al. 2013). Compared to more classical evolutionary algorithms, such as genetic algorithms (GA), EDA produces an offspring population of candidate solutions by sampling from the probability distribution of the parent population. This omits the manual design step of evolutionary operators, such as crossover and mutation operations in GA. Different from GA's micro-level evolution mode based on genes, EDA adopts a macro-level evolution mode based on the population probability distribution, which provides stronger global search ability and faster convergence speed for optimal solution. The procedure of the basic EDA is as follows:

Step 1: generate  $N$  individuals randomly as the initial population  $P(0)$  of candidate solutions.

Step 2: calculate the fitness of each individual in the current population  $P(t)$  to confirm whether the termination condition is met, and if so, terminate the procedure and output the current best individual as the global optimal solution, otherwise, go on to the next step.

Step 3: according to the individual fitness ranking, the first  $M$  ( $M \leq N$ ) promising individuals are selected from the current population  $P(t)$  to form the promising subpopulation  $D(t)$  of the  $t + 1$ th evolutionary generation.

Step 4: Update the probability model by estimating the probability distribution of the promising subpopulation  $D(t)$ .

Step 5: sample randomly  $N$  new individuals via the probability model to generate a new population  $P(t + 1)$  in the  $t + 1$ th evolutionary generation, and return to step 2.

##### 4.2. Knowledge guided multi-objective EDA (KEDA)

The above procedure of the basic EDA reflects the overall evolutionary trend of the population for candidate solutions and pays more attention to global exploration in the solution space. It ignores the more sophisticated exploitation of the local solution space. Moreover, EDA cannot be directly applied to multi-objective optimisation due to its initial design for single objective function optimisation. We therefore introduce a novel variant of EDA, called KEDA, which includes three types of knowledge guided improvement strategies for the population initialisation, the update of the probability model and the local solution search.

###### 4.2.1. Solution representation

The solution representation is obtained through coding and decoding of delivery sequences of material orders into AGV travel routes. If there are  $n$  material orders to



**Table 1.** Dispatching rules used for population initialisation.

Name	Definition	Knowledge involved
<b>First Coming First Service (FCFS)</b>	$PRI_i = T_i^o$	The generation time of the replenishment order.
<b>Shortest Workstation Distance to Depot (SWD)</b>	$PRI_i =  x_i - x_0  +  y_i - y_0 $	The location of workstation with a replenishment order.
<b>Earliest Scheduled Delivery Time (ESDT)</b>	$PRI_i = T_i^s$	The scheduled delivery time of replenishment order.
<b>Scheduled Delivery Time with Workstation Distance Weighted (SDTDW)</b>	$PRI_i = \frac{ x_i - x_0  +  y_i - y_0 }{L + W} + \frac{T_i^s}{2CT}$	The scheduled delivery time of material delivery order and corresponding workstation's location.
<b>Smallest Delivery Time Window (SDTW)</b>	$PRI_i = T_i^l - T_i^e$	The accepted delivery time window of material delivery order.
<b>Least Remaining Material Quantity (LMQ)</b>	$PRI_i = S_i^o$	The remaining stock of material buffer of workstation with a material delivery order.
<b>Delivery Urgency Ratio (DUR)</b>	$PRI_i = \frac{S_i^o}{S} \times \frac{T_i^s}{2CT}$	The scheduled delivery time of material delivery order and the remaining stock of corresponding workstation's material buffer.
<b>Delivery Slackness (DS)</b>	$PRI_i = \frac{S_i^o}{S} \times (T_i^l - T_i^s)$	The scheduled delivery time, accepted latest delivery time of material delivery order and the remaining stock of corresponding workstation's material buffer.
<b>Workstation Distance Weight by Remaining Material Quantity Weight (DMQW)</b>	$PRI_i = \frac{S_i^o}{S} \times \frac{ x_i - x_0  +  y_i - y_0 }{L + W}$	The call workstation's location and the remaining stock of corresponding material buffer.
<b>Delivery Time Window by Workstation Distance Weight (DTDW)</b>	$PRI_i = (T_i^l - T_i^e) \times \frac{ x_i - x_0  +  y_i - y_0 }{L + W}$	The call workstation's location and the delivery time window of corresponding material delivery order.
<b>Scheduled Lead Time (SLT)</b>	$PRI_i = T_i^s - T_i^o$	The generated time and the scheduled delivery time of material delivery order.
<b>Scheduled Earliest Delivery Span by Workstation Distance Weight (ESDW)</b>	$PRI_i = (T_i^s - T_i^e) \times \frac{ x_i - x_0  +  y_i - y_0 }{L + W}$	The call workstation's location and the delivery time window of corresponding material delivery order.

Note:  $PRI_i$  denotes the dispatched priority value of material delivery order  $i$ , and the smaller the value of  $PRI_i$ , the higher the priority of order  $i$ .  $x_0$  and  $y_0$  are the position coordinates of the depot. The two letters  $L$  and  $W$  express the length and width of the manufacturing workshop, respectively. The definition of other symbols in Table 1 are given in Section 3.2.

be delivered in a production cycle, we can use  $n$  integers between 1 and  $n$  to represent the identity of orders, and 0 for representing the start of each AGV route from the depot. A permutation of  $n$  orders' identities is first regarded as a delivery sequence of  $n$  orders, and then  $n$  orders are assigned to AGVs for confirming AGV travel routes according to the delivery sequence, while satisfying the delivery time constraint of each order and the constraints of AGV's max service order number and load capacity. For example, there are 10 orders with a delivery sequence as [3,7,1,9,4,2,5,10,8,6]. Suppose that the delivery sequence can be decoded into 3 AGV routes following the order of initial sequence and the problem constraints. As an example, the delivery sequence of orders in AGV route 1 could be [3, 7,1,9] and the other two routes [4, 2, 5] and [10, 8, 6]. Two adjacent AGV routes are separated by 0 indicating the depot, hence the solution representation is [0,3, 7, 1,9,0, 4,2,5,0,10,8,6].

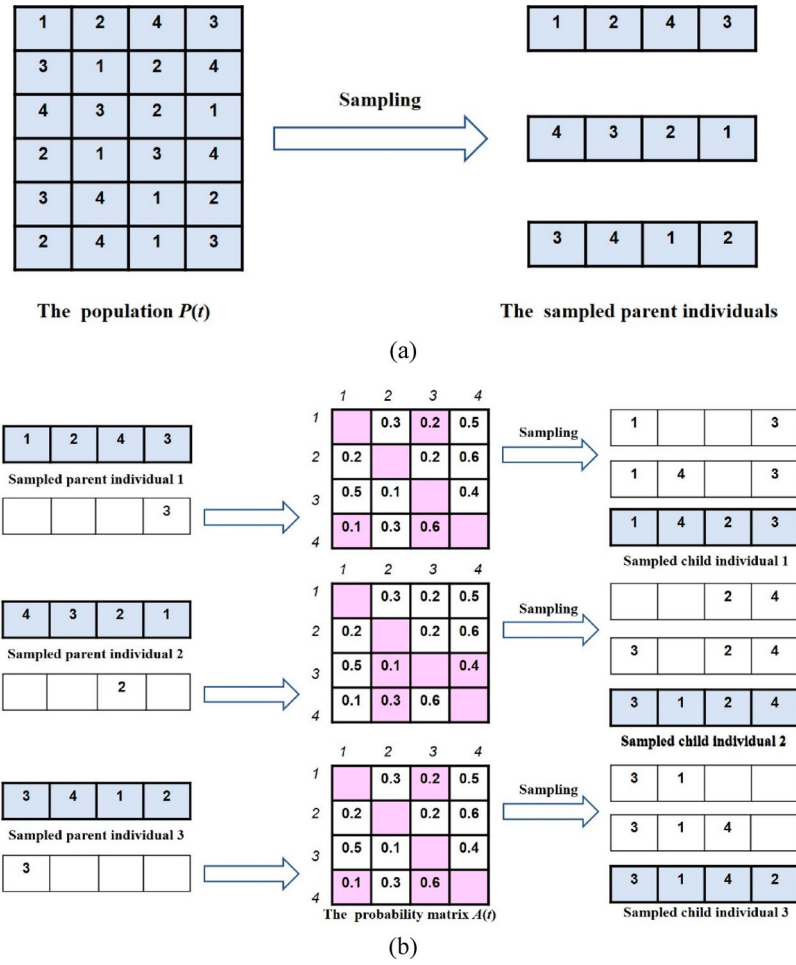
#### 4.2.2. Dispatching rule guided population initialisation

It is generally accepted that an initial population that consists of individuals with high quality and good diversity facilitates fast convergence of the algorithm and good solutions in evolutionary optimisation. We therefore adopt a dispatching rule guided strategy to obtain

a set of initial high quality order delivery sequences. In addition to FCFS, which is widely used in practice, we scanned the literature for other dispatching rules. Table 1 summarises the 12 dispatching rules considered. To ensure diversity of the initial population, alternative order delivery sequences are generated through random permutations of all material delivery orders. Suppose that the size of the population for candidate solutions is  $N$ . First, 12 initial order delivery sequences can be generated by the 12 dispatching rules presented in Table 1, where one dispatching rule is responsible for generating one order delivery sequence. Then  $N-12$  initial order delivery sequences are produced randomly. Finally, the  $N$  initial order delivery sequences can be decoded into  $N$  initial solutions according to the solution representation described above. This provides the population initialisation.

#### 4.2.3. Probability model and offspring generation

The EDA produces offspring by sampling through a probability model, which is different from the crossover and mutation operators of GA. So, the probability model has a crucial impact on the performance of the EDA. In this work, the probability model is designed as a probability matrix  $A$ . Each element  $a_{ij}$  in the matrix represents the probability that an AGV transitions from delivery



**Figure 2.** Offspring generation via sampling the probability matrix.

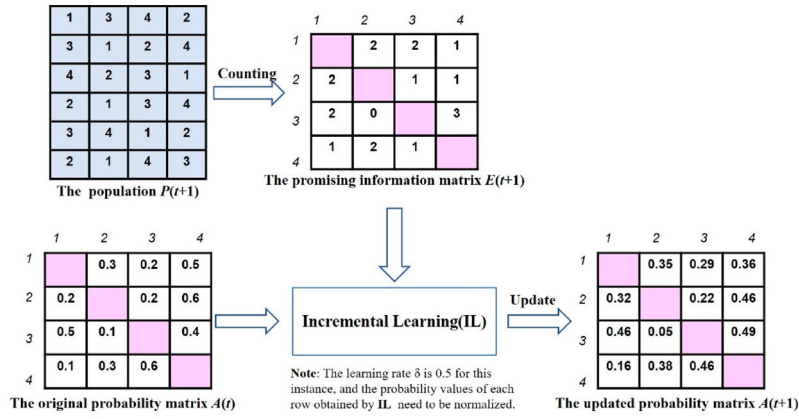
point (workstation with material demand)  $i$  to delivery point (workstation with material demand)  $j$ . For all values of  $i$  and  $j$ ,  $a_{ij}$  is initialised to  $a_{ij}^0 = 1/n$ , where  $n$  is the matrix dimension that is equal to the order quantity. This ensures that the whole solution space can be sampled uniformly.

With this matrix, an offspring of material order delivery sequences can be sampled. For example, if the KEDA has a population  $P(t)$  with the size of six individuals and searches good solutions for a material supply task with four material delivery orders at evolutionary generation  $t$ , as shown in Figure 2(a), then it samples half of the parent individuals in  $P(t)$  randomly for offspring generation. Moreover, each sampled parent individual produces a sampled child individual sampling the delivery sequence space according to the probability matrix  $A(t)$  at evolutionary generation  $t$ , as shown in Figure 2(b).

Specifically, the ratio of unchanged elements in the delivery sequence of parent individuals is first determined, and then the specified ratio selected randomly from the delivery sequence to initiate the execution of sequential sampling for other elements' positions in the

delivery sequence of child individuals. In Figure 2(b), the ratio of unchanged elements in the delivery sequence of parent individuals is 0.25 at evolutionary generation  $t$ , so each delivery sequence of parent individuals has only one unchanged element to be selected randomly. The sampling process for the delivery sequence of child individuals starts from the delivery point  $i$  represented by the unchanged element. For every position  $i$  in temporary sequences (unmarked sequences in Figure 2(b)), material delivery order (delivery point)  $j$  is selected with a probability  $a_{ij}$  (current largest value in row  $i$  of probability matrix  $A(t)$ ). If order  $j$  has already appeared in the temporary sequence, it means that order  $j$  has been scheduled. Then, the  $j$ th value in row  $i$  of  $A(t)$  will be set as zero (marked in Figure 2(b)). In such a way, a sampled child individual is constructed until all the orders appear in the delivery sequence.

The ratio of unchanged elements is gradually increased with the increase of the evolutionary iteration. The algorithm has consequently a stronger exploration ability for population diversity at an evolutionary initial stage, and a more refined exploitation ability for excellent



**Figure 3.** Probability matrix update by incremental learning.

solutions at an evolutionary mature stage. We let  $\varphi$  denote the ratio of unchanged elements in the sampled parent individual, namely the inheritance ratio, which can be dynamically adjusted according to Equation (20):

$$\varphi = \varphi_{\min} + \frac{t}{t_{\max}}(\varphi_{\max} - \varphi_{\min}) \quad (20)$$

Where  $\varphi_{\min}$  and  $\varphi_{\max}$  are the minimum value and the maximum value of the inheritance ratio for the sampled parent individual, respectively, while  $t_{\max}$  denotes the maximum evolutionary generation.

#### 4.2.4. Pareto evolutionary performance guided probability model update

This study considers a multi-objective optimisation model with possibly conflicting objectives. We therefore seek to obtain the solutions' Pareto-optimality instead of offering optimal trade-offs between objectives (Li et al. 2018; Fang et al. 2020). For this, the original method for solutions evaluation in EDA is replaced with a Pareto non-dominated evaluation in the proposed KEDA. We also apply the hypervolume metric (HV) to compare the qualities of adjacent generations of non-dominated solution sets and measure the Pareto evolutionary performance in the execution process of KEDA. The larger the HV value, the better the comprehensive performance of the corresponding algorithm. The HV metric can capture the properties of convergence and the distribution of the Pareto non-dominated solution set. It can therefore be used as a criterion for the archiving strategy, diversity mechanism or selection for the algorithm population to guide the population evolution (Zhou, Guo, and Philip Chen 2012). This significantly improves the performance of traditional Pareto non-dominated evaluation. It is also suitable for complex multi-objective optimisation problems derived from practical scenarios without known true Pareto fronts, such as our problem. The HV can be used without known theoretical reference

points of optimisation objectives but just with custom reference points (Beume, Naujoks, and Emmerich 2007; Guerreiro, Manquinho, and Figueira 2021; Yu, Jin, and Olhofer 2021).

Our probability model is the matrix  $A$  of probabilities of a AGV transition from a delivery point to another. As shown in Figure 3, when KEDA obtains the population  $P(t+1)$  at the end of the generation  $t$ , it first counts valid AGV transitions in delivery sequences of the population  $P(t+1)$  to construct the promising information matrix  $E(t+1)$ , and then updates the probability matrix as  $A(t+1)$  based on the matrix  $E(t+1)$  and  $A(t)$ .

Specifically, let  $a_{ij}$  be an element of the matrix  $A$ , then the matrix  $A$  can be updated via the following equation:

$$a_{ij}^{t+1} = \delta^{t+1} \frac{e_{ij}^{t+1}}{N} + (1 - \delta^{t+1}) a_{ij}^t \quad (21)$$

Where  $\delta^{t+1} \in (0, 1)$  is the current learning rate of matrix  $A$ ,  $e_{ij}^{t+1}$  is the element of the matrix  $E(t+1)$ , which represents the current number of AGV transitions appearing in delivery sequences of the population  $P(t+1)$  from delivery point  $i$  to  $j$ , and  $N$  denotes the population size.

The update of matrix  $A$  is essentially a process of incremental learning, where the first term on the right-hand side of Equation (21) represents learning information from the promising AGV transitions of the new population. To closely control the promising AGV transitions information in the process of incremental learning, the learning rate  $\delta$  of matrix  $A$  should be changed adaptively based on the performance of current solutions. KEDA therefore adopts a Pareto evolutionary performance guided adaptive learning rate for the probability model update, which adjusts the HV metric in the Pareto solutions evaluation as given in Equation (22):

$$\delta^{t+1} = \delta^t + \delta^t \frac{HV^{t+1} - HV^t}{HV^t + 0.01} \quad (22)$$

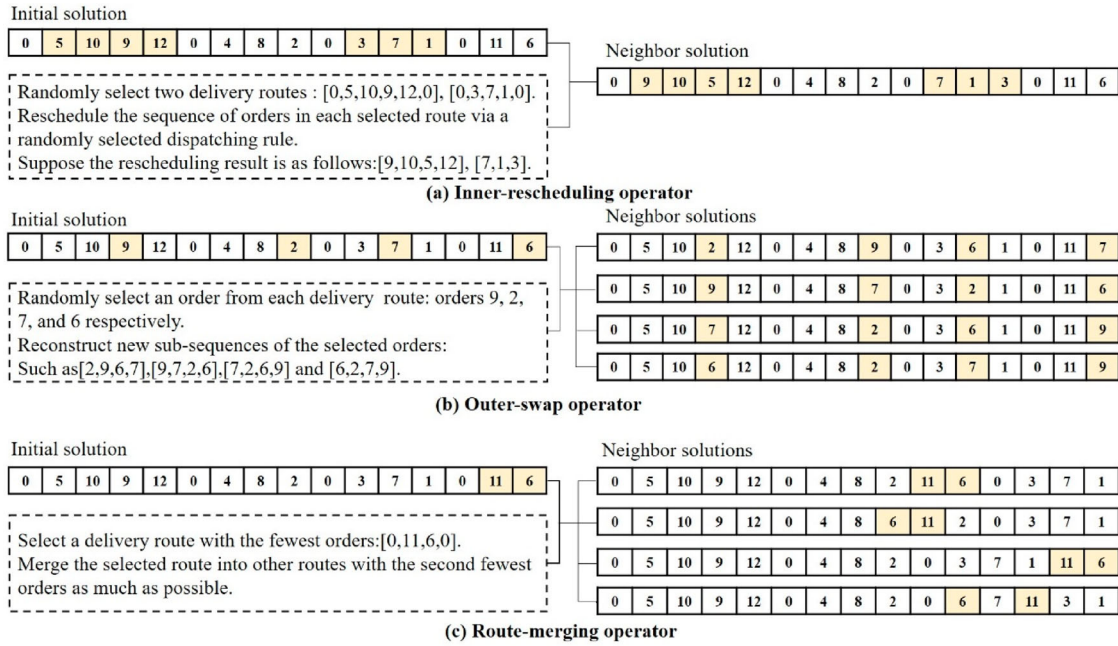


Figure 4. Illustration of neighbourhood structures.

Where  $HV^{t+1}$  is the current hypervolume value of the Pareto solutions set, while  $HV^t$  denotes the hypervolume value of the Pareto solutions set at the last evolutionary generation.

Equation (22) provides an adaptive parameter updating method for the learning rate  $\delta$  based on the Hebb rule (Heyong and Ming 2019). It allows for emphasising different learning weights for current population and previous population according to the relative performance of current HV and previous HV in matrix A. Note that the learning rate  $\delta$  needs to be set to 0.1 when Equation (22) gives a result less than or equal to 0, while it needs to be set to 0.9 if the result given by Equation (22) exceeds 1. This ensures that the value of the learning rate  $\delta$  is between 0 and 1. In this study, (1, 1) is selected as the reference point, and if PS is a set of all pareto solutions obtained by a multi-objective evolutionary algorithm, then the HV value of PS can be calculated by Equation (23) (Zitzler and Thiele 1999):

$$HV(PS) = L(\cup_{i=x}^{|PS|} v_x) \quad (23)$$

Where  $L$  is the Lebesgue measure,  $|PS|$  is the number of Pareto solutions, and  $v_x$  is the hypervolume formed by the  $x$ th pareto solution and the reference point. The two objective values of each Pareto solution are normalised according to the method Max–Min (Zhang et al. 2020).

#### 4.2.5. Delivery satisfaction guided variable neighbourhood search

In order to enhance local exploitation around good solutions, and to force the algorithm away from local optima, a variable neighbourhood search (VNS) mechanism (Bilyk and Monch 2012) is introduced in the KEDA. However, performing a VNS on all individuals of the algorithm population would be quite time-consuming due to its complexity. To ensure high execution efficiency for KEDA, a VNS method guided by the delivery satisfaction of orders is proposed, and only the subset of the population individuals that meets the specified delivery satisfaction are subject to the VNS.

A reasonable neighbourhood structure is also crucial to the performance of the VNS. In our proposed VNS method, three kinds of neighbourhood structures are considered to ensure good local search ability while reflecting the problem characteristics: inner-rescheduling operator, outer-swap operator, and route-merging operator. All three are illustrated in Figure 4.

Suppose that there is a material supply task with 12 delivery orders, and a feasible initial solution of this instance can be represented as [0,5,10,9,12,0,4,8,2,0,3,7,1,0,11,6]. The inner-rescheduling operator, illustrated in Figure 4(a), first randomly selects two delivery routes from the initial solution, and then reschedules the sequence of orders in each selected route via a randomly selected dispatching rule. Finally, it updates the selected routes with the corresponding rescheduled sequence of orders. The outer-swap operator, illustrated in Figure 4(b), first randomly selects an order from each



delivery route of the initial solution, and then reconstructs new sub-sequences of the selected orders using random permutations. Finally, it assigns these orders to the original sequence in terms of the precedence relation of orders in each sub-sequence. The route-merging operator, illustrated in Figure 4(c), first selects the delivery route with the fewest orders, and then merges the selected route into other routes with the second fewest orders as much as possible. This can reduce the number of AGVs and help optimise AGV cost.

Finally, if a neighbourhood operator produces an infeasible solution, violating some constraints of the model, the corresponding complete delivery sequence of the infeasible solution is re-decoded into a feasible solution satisfying the problem constraints by the decode method described in section 4.2.1.

In the proposed VNS, the three kinds of neighbourhood structures above are called in order, and we use  $O$  ( $O = 1, 2, 3$ ) to represent the order. Adopting a delivery satisfaction guided strategy, the procedure of our VNS can be described as follows:

Step 1: Select a set of candidate solutions with the mean delivery satisfaction of all orders greater than 0.1 from the current non-dominated solutions, and consider the selected solutions as an initial solution set  $S$ . If the number of the selected solutions exceeds 10, only the best 10 are taken as the initial solution set  $S$ . When no candidate solution has a mean delivery satisfaction greater than 0.1, then a solution is randomly selected from the current non-dominated solutions and taken as the initial solution set  $S$ .

Step 2: pop up an initial solution  $s$  from the set  $S$ , then input the solution into the variable neighborhood search and set  $O = 1$ .

Step 3: seek the local optimal solution  $s_o$  of the initial solution  $s$  in the  $o$ th neighborhood.

Step 4: Compare the performance of the local optimal solution  $s_o$  and the initial solution  $s$  via the Pareto non-dominated evaluation. If the local optimal solution  $s_o$  dominates the initial solution  $s$ , then set  $s = s_o$ , and  $O = 1$ . Otherwise, set  $O = O + 1$ .

Step 5: If  $O \leq 3$ , then repeat Steps 3–4. Otherwise, go to Step 6.

Step 6: Take the solution  $s$  into the result set  $R$ . If the set  $S$  is not null, go to Step 2. Otherwise, output the result set  $R$ .

### 4.3. KEDA with delivery satisfaction evaluation (KEDAS)

KEDA presented above presents a novel multi-objective EDA that combines three types of knowledge guided strategies for the improvement of the algorithm ability. To cope with uncertain replenishment times and stochastic

AGV unloading duration, a fuzzy soft time window for the time range of material order delivery has been defined and integrated in the formulation of our problem. Based on this time window, the delivery satisfaction constraint of orders, a soft constraint, is introduced into the model to further improve the quality of the solution. This soft constraint can be used for new solutions evaluation during the algorithm iteration. A flowchart of combining the KEDA with the delivery satisfaction constraint for evolutionary optimisation is outlined in Figure 5. This includes knowledge guided improvement strategies (boxed in dotted line) and offspring evaluation using the delivery satisfaction constraint (boxed in full line), where the offspring solutions violating this constraint are penalised by adding a large number to each of their objectives. Solutions with poor delivery satisfaction have consequently a higher probability of being eliminated for the acceleration of evolutionary optimisation. The overall procedure of the KEDA with delivery satisfaction evaluation (KEDAS) is detailed in Appendix A.

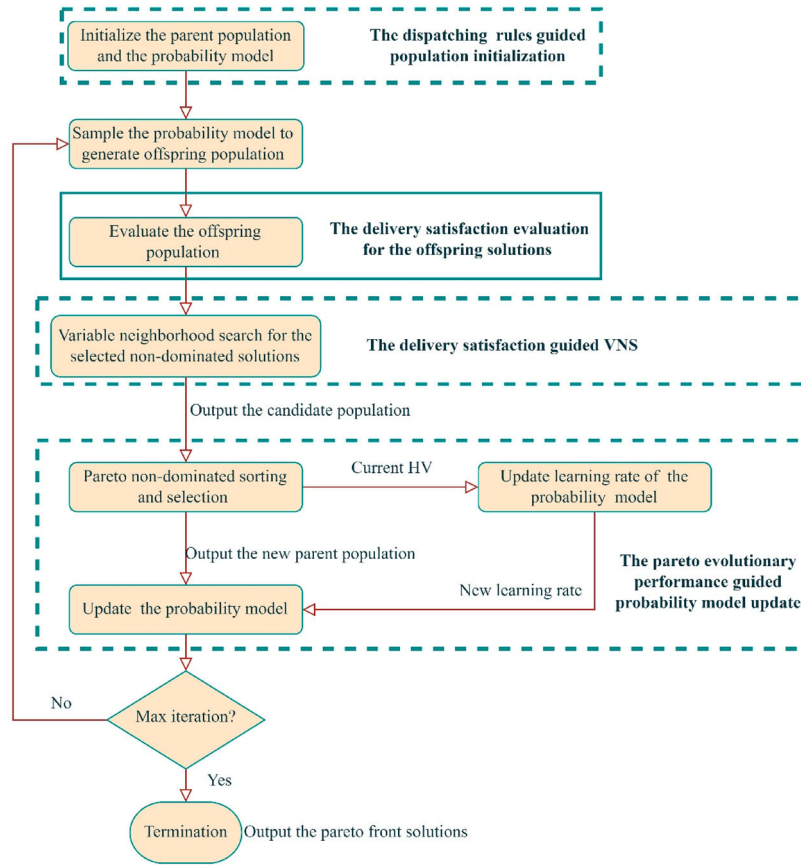
## 5. Experimental study

This section presents results from computational and simulation experiments to test our model and the proposed algorithm. We first present the experimental setting. The model is then verified using small scale instances. Finally, we use large-scale instances to provide a detailed and comprehensive numerical comparison of the proposed algorithm against state-of-the-art algorithms.

### 5.1. Experimental settings

All test instances used in this study are collected from a general intelligent manufacturing workshop that uses multi AGVs for material delivery at the major domestic appliance and air-conditioner producer that motivated our study. The test instances with the different scale of  $n = 10, 15, 20, 30, 40, 50$  material delivery orders reflect the dynamics of workstation production and AGV unloading via a specified demand time window of each workstation and varying unloading duration with known normal probability distributions, respectively. These instances are divided into small-scale instances with  $n = 10, 15, 20$  material delivery orders for model verification, and large-scale instances with  $n = 30, 40, 50$  material delivery orders for algorithms evaluation, following a splitting instances pattern similar to, for example, Zou et al. (2020). For all instances, each size of  $n$  has three replicates with different delivery requirements, so there are in total 18 test instances. Moreover, each instance includes the number of orders and the number of starving buffers,





**Figure 5.** Flowchart of the KEDAS.

**Table 2.** Parameters of the proposed model.

Items	Values	Items	Values
CT	360 s	$t_m$	30 s/piece
$T_0$	365 s	$g_m$	0.75 kg/piece
$N_s$	10	$S$	48
$v$	3 m/s	$c_d$	1
$C$	250 kg	$c_a$	200
$u_s$	15 s	$c_p$	0.1
$\sigma_s$	1	$\alpha$	0.5
$\varepsilon$	0.86	$\beta$	0.8

where a starving buffer is a material buffer for which the quantity of remaining material stocks is below one third of its total capacity. For the same number of orders, more starving buffers mean more material needs to be delivered, and more AGVs need to be dispatched. Taking an instance with 15 orders and 10 starving buffers as an example, the instance is denoted as N15S10, where N15 and S10 indicate the number of orders and the number of starving buffers existing among these orders, respectively. The related parameters of our model are presented in Table 2.

For model verification, we consider the FCFS, LMQ, and SDTDW dispatching rules as benchmark methods

to solve the model without delivery satisfaction constraint. Meanwhile, we also deploy the KEDA to compare its solutions against that of KEDAS. For the evaluation of the proposed KEDAS, we then compare its performance against five alternative multi-objective algorithms which showed much promise in the literature: (i) the non-dominated sorting genetic algorithms II & III (NSGA II and NSGA III; e.g. (Sathiya et al. 2021; Subramanian et al. 2021; Guo, Jiang, and Yang 2022)), (ii) the reference vector guided evolutionary algorithm (RVEA; e.g. Cheng et al. 2016), (iii) the multi-guide particle swarm optimisation (MGPSO; e.g. Scheepers, Engelbrecht, and Cleghorn 2019) and (iv) a multi objective version of basic EDA (MEDA), which combines Pareto non-dominated sorting with EDA.

To ensure a fair comparison, the same number of solution evaluations is applied for above evolutionary algorithms, that is, every evolutionary algorithm has the same number of maximum iterations and the same population size. The total solution evaluation times for each compared algorithm are equal to the product of the population size and the maximum iteration time. The minimum and maximum values of the inheritance ratio of the proposed KEDAS are set to 0.25 and 0.5, respectively, and

**Table 3.** Best results of the compared models across the instances solved.

Instance	MEDS												MWDS		
	FCFS			LMQ			SDTDW			KEDA			KEDAS		
	$F_1$	$F_2$	$\overline{SA}$	$F_1$	$F_2$	$\overline{SA}$	$F_1$	$F_2$	$\overline{SA}$	$F_1$	$F_2$	$\overline{SA}$	$F_1$	$F_2$	$\overline{SA}$
N10S2	1982	656	0.341	2928	775	0.091	2647	674	0.094	1766	386	0.345	<b>1672</b>	<b>356</b>	<u><b>0.486</b></u>
N10S5	2789	683	0.088	2716	1172	0	2502	894	0.088	1614	439	0.421	<b>1613</b>	<b>296</b>	<u><b>0.513</b></u>
N10S10	2206	464	0.093	2416	502	0.329	1932	755	0.113	1792	309	0.252	<b>1740</b>	<b>237</b>	<u><b>0.486</b></u>
N15S5	2886	783	0.147	3707	1235	0.093	3476	1176	0	2572	<b>599</b>	0.180	<b>1994</b>	623	<u><b>0.258</b></u>
N15S10	2794	1011	0.151	4475	1643	0.113	2787	1312	0.020	2563	627	0.223	<b>2507</b>	<b>613</b>	<u><b>0.271</b></u>
N15S15	4768	980	0.136	4189	1294	0.021	3309	1350	0.107	<b>2666</b>	661	0.205	2678	<b>560</b>	<u><b>0.314</b></u>
N20S10	3852	1811	0.201	4660	2466	0.026	3867	1910	0.045	2850	1533	0.090	<b>2835</b>	<b>1327</b>	<u><b>0.249</b></u>
N20S15	5012	1050	0.271	5618	1796	0.084	4028	2100	0	3055	1050	0.271	<b>3004</b>	<b>1006</b>	<u><b>0.289</b></u>
N20S20	4882	1402	0.064	4690	2348	0	3666	2004	0.056	3491	1239	0.156	<b>3443</b>	<b>1162</b>	<u><b>0.171</b></u>

the initial learning rate is set to 0.5 to equally represent the information of current and previous populations at the initialisation of the algorithm. The parameters of the other algorithms compared in this study were experimentally determined or based on the best recommendations of the corresponding literature. For MEDA, the inheritance ratio and learning rate are specified as 0.3 and 0.7, respectively. For NSGA II, the crossover probability and mutation probability are set to 0.8 and 0.1, respectively, and the same setting is adopted for NSGA III. The sensitivity alpha and minimum reference vector angle of RVEA are set to 2 and 0.1, respectively, and its crossover probability and mutation probability are both specified as 1 (Jazzbin 2021). For MGPSO, the inertia weight is set to 0.6, its three acceleration coefficients are taken as 1.85, 1.55 and 1.80, respectively, and the tournament size is set to 3 (Scheepers, Engelbrecht, and Cleghorn 2019).

All methods, including dispatching rules, adopt the same solution representation defined in this paper, and all evolutionary algorithms adopt Pareto non-dominated sorting for solutions evaluation. It should be noted that the original update approach of the velocities and positions of particles of MGPSO needs to be transformed into a discrete form because the MGPSO was initially developed for continuous optimisation problems. The detailed discretization approach is detailed in Appendix A. Our discretization does not change the fundamental characteristics of the original MGPSO, so we still use the term MGPSO to indicate the discrete version of MGPSO in this paper.

Finally, we use Python programming language to code all test methods and the proposed model. NSGA II, NSGA III and RVEA were developed using Geatpy (Jazzbin 2021), a high-performance algorithm framework for genetic and evolutionary algorithms. MGPSO, MEDA and KEDAS were implemented according to their respective procedure. All computational experiments are run on a PC with an Intel Core i5-1035G1 CPU at 1.19 GHz and with 16GB of RAM.

## 5.2. Model verification

In order to illustrate the effectiveness of the proposed AGVDP model that considers delivery satisfaction, we perform a set of tests for comparing the solutions obtained by different methods on small-scale instances. The proposed model excluding the delivery satisfaction constraint (MEDS) is solved by FCFS, LMQ, SDTDW, and KEDA. The proposed KEDAS is used to solve the model with the delivery satisfaction constraint (MWDS). The solutions provided by the three dispatching rules are regarded as the benchmark solutions on each test instance. For the evolutionary algorithms KEDA and KEDAS, the population size is set to 30 and the maximum iteration is taken to be 50, so the total number of solution evaluations is equal to 1500 on each instance. KEDA and KEDAS run 30 independent replications on each instance to limit the effect of randomness, whereas every dispatching rule method is only run once on each instance.

In addition to the two optimisation objectives of the proposed model, total transportation cost  $F_1$  and total delivery time deviation  $F_2$ , we also record the mean delivery satisfaction  $\overline{SA}$  for all orders. Best solution results across the 30 replications are given in Table 3. For all instances, and solving MEDS, the proposed KEDA obtains better solution results than the benchmark provided by the three dispatching rules on all metrics. FCFS gives relative better results for all metrics across the three dispatching rules for most instances. In terms of the mean delivery satisfaction, LMQ gives the worst  $\overline{SA}$  values for the instances N10S5 and N20S20, as does SDTDW for the instances N15S5 and N20S15. The KEDAS obtains the best values for all instances when solving the model MWDS. Moreover, the KEDAS finds the best values for the two objectives of the proposed model on each test instance except for the instances N15S5 and N15S15. For small-scale instances, the model MWDS solved by the KEDAS achieved better solution results than the model MEDS solved by the KEDA (or the dispatching rules).

This implies that the AGVDP model with the delivery satisfaction constraint is more effective than the AGVDP model without this soft constraint. An extended analysis of the impact of the order delivery satisfaction constraint on order delivery punctuality can be found in Appendix B.

### 5.3. Evaluation of the proposed Kedas

This section compares the proposed KEDAS algorithm with five multi-objective optimisation algorithms, NSGA II, NSGA III, RVEA, MGPSO and MEDA, using large-scale instances. These five algorithms provide a comprehensive benchmark. Meanwhile, the comparison between MEDA and KEDAS not only assesses the effect of applying knowledge-guide improvement strategies to EDA, but also evaluates the performance of the proposed KEDAS for solving the proposed AGVDP model with the delivery satisfaction constraint. For all test instances, the maximum iteration number of each algorithm is set to 100, based on Guo et al. (2022), and the Population Size is Specified as 50. The total number of solution evaluations is therefore up to 5000 for each algorithm on each test instance. Each instance is replicated 30 times for each algorithm. As a further benchmark, the FCFS dispatching rule, which is widely applied in practice, is also considered.

Table 4 reports the average value (Avg) and the minimum value (Min) for each sub-objective obtained by the six evolutionary algorithms for instances with 30 material delivery orders. From Table 4, we can observe that KEDAS realises the minimum values for both optimisation objectives on instance N30S15. Meanwhile, the average values of the two objectives realised by KEDAS are both less than those provided by alternative methods for instance N30S15. As for the instances N30S0 and N30S30, KEDAS realises much better solution results than other methods, except for NSGA II in terms of some indicators. However, KEDAS still realises near-optimal indicator values in comparison to the optimal values given by the NSGA II. As expected, the benchmark method FCFS provides the worst solution results. KEDAS realises the best indicator values for most of the test instances with different material order delivery requirements, while alternative methods cannot maintain a robust optimisation performance for all test instances. The results for instances with 40 and 50 orders confirm the results in Table 4. They can be found in Appendix C given space constraints.

To further compare the multi-objective optimisation capabilities of the algorithms, we not only compare the Pareto fronts of non-dominated solutions obtained by these algorithms, but also evaluate the convergence and

**Table 4.** Results of each sub-objective for instances with 30 orders.

Instance	Method	$F_1$		$F_2$	
		Avg	Min	Avg	Min
N30S0	FCFS	7159	7159	2636	2636
	NSGA II	4961.22	4767	2202.90	<b>1570</b>
	NSGA III	5473.71	4821	2198.3	1797
	RVEA	5359.63	4933	2050.0	1750
	MGPSO	6003.16	5661	2517.57	1986
	MEDA	5356.58	4848	2299.71	1978
	KEDAS	<b>4309.02</b>	<b>4295</b>	<b>1960.5</b>	1798
N30S15	FCFS	8243	8243	2254	2254
	NSGA II	5632.87	4565	1975.5	1304
	NSGA III	5761.02	4516	1890.28	1399
	RVEA	5100.88	4711	1768.37	1443
	MGPSO	6745.38	5465	2029.0	1765
	MEDA	5653.45	4661	1949.42	1554
	KEDAS	<b>4837.41</b>	<b>4405</b>	<b>1679.5</b>	<b>1298</b>
N30S30	FCFS	8103	8103	1803	1803
	NSGA II	5563.40	<b>4853</b>	<b>1231.8</b>	1131
	NSGA III	5922.14	5511	1591.28	1183
	RVEA	5865.64	5546	1489.27	1142
	MGPSO	6991.46	6359	1687.16	1382
	MEDA	5776.17	5499	1562.6	1316
	KEDAS	<b>5465.61</b>	4906	1388.0	<b>1098</b>

**Note:** Results of each sub-objective for instances with 40, 50 orders can be found in **Appendix C** given space constraints.

distribution of their Pareto solution sets by analyzing two performance metrics: the HV and the Spacing. The Spacing is also a common metric for the Pareto evolutionary performance. A smaller Spacing value indicates a better distribution of Pareto solution sets. The average HV values and average Spacing values of each algorithm for 30 runs on each instance are presented in Table 5. We can observe that the average HV values obtained for KEDAS are better than those obtained for alternative algorithms for most instances. Regarding the average Spacing values, KEDAS realises the minimum Spacing values for 7 out of 9 instances and realises the same best performance as NSGA II and NSGA III on the remaining two instances. The statistical significance of performance differences was assessed using an Analysis of Variance (ANOVA). Results are presented in Table 6.

Figure 6(a–b) summarise the overall distribution of the HV and Spacing values, respectively, obtained by each algorithm's 30 runs on all large-scale instances. KEDAS gives a relatively more competitive distribution for both metrics, specifically compared to MEDA and MGPSO.

Figure 7(a–c) show the Pareto fronts of non-dominated solutions of three instances with different demand point sizes to better illustrate performance differences across algorithms. Every Pareto front can be obtained by selecting optimal non-dominated solutions from the combined Pareto solution set generated by the corresponding algorithms. Figure 7(a–c) highlights that the Pareto fronts realised by KEDAS have relatively better quality than

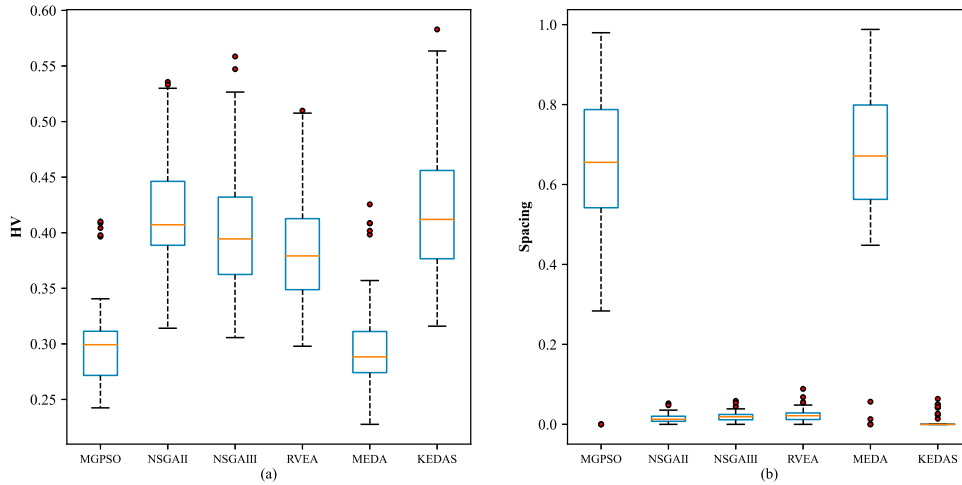
**Table 5.** Average values of each performance metric for the compared algorithms.

Instance	HV						Spacing					
	NSGA II	NSGA III	RVEA	MGPSO	MEDA	KEDAS	NSGA II	NSGA III	RVEA	MGPSO	MEDA	KEDAS
N30S0	<b>0.43</b>	0.39	0.37	0.29	0.30	0.41	0.01	0.03	0.03	0.65	0.57	<b>0</b>
N30S15	<b>0.46</b>	0.44	0.42	0.33	0.33	<b>0.46</b>	0.01	0.02	0.03	0.69	0.71	<b>0</b>
N30S30	0.52	0.53	0.49	0.40	0.41	<b>0.54</b>	<b>0.02</b>	<b>0.02</b>	0.03	0.64	0.74	<b>0.02</b>
N40S0	0.42	0.42	0.39	0.30	0.29	<b>0.44</b>	0.02	0.03	0.02	0.77	0.77	<b>0</b>
N40S20	<b>0.39</b>	0.37	0.36	0.28	0.28	<b>0.39</b>	0.02	0.02	0.04	0.75	0.74	<b>0.01</b>
N40S40	0.35	0.33	0.32	0.25	0.24	<b>0.36</b>	<b>0.01</b>	0.02	0.03	0.68	0.76	<b>0.01</b>
N50S0	<b>0.43</b>	0.42	0.40	0.33	0.31	<b>0.43</b>	0.01	0.02	0.01	0.71	0.78	<b>0</b>
N50S25	0.39	0.38	0.37	0.28	0.28	<b>0.46</b>	0.01	0.01	0.02	0.68	0.66	<b>0</b>
N50S50	0.37	0.35	0.33	0.26	0.27	<b>0.38</b>	0.03	0.02	0.01	0.67	0.67	<b>0</b>

**Table 6.** ANOVA results on performance metrics of compared algorithms.

	Source of variance	Sum of squares	Degrees of freedom	Mean squares	F-ratio	P-value
HV	Compared algorithms	0.726870	5	0.145374	48.65703	0.000000
	Residual	0.788761	264	0.002988		
Spacing	Compared algorithms	23.030072	5	4.606014	136.453716	0.000000
	Residual	8.911357	264	0.033755		

**Note:** Further statistical test results on the multiple Comparison of HVs/ Spacings' Means with Tukey Honest Significant Difference (HSD) and Family-wise Error Rate (FWER = 0.05) are given in **Appendix C**.

**Figure 6.** Boxplots of HV and Spacing values obtained by compared algorithms.

those obtained by the alternative algorithms on different instances.

We also conducted a set of extended tests, where we increased the maximum iterations to 1000 for Three Instances with Different Demand Point Number, to Explore the Solution Convergence Trends of the Considered Algorithms. Considering the NP-hard nature of the proposed problem, the compared algorithms are likely to not completely reach the convergence stage for large-scale instances and limited computing time. It is reasonable to assume a short computing time in actual manufacturing workshops, instead of pursuing the complete computing convergence at the cost of long runtimes.

Figure 8(a–c) present the mean convergence trend curves of the HV values over evolutionary iteration times on the instances N30S15, N40S20 and N50S25. Along

with the iteration increase, the KEDAS keeps the dominant convergence trends with better results compared to other algorithms, especially showing a significant superiority over MGPSO and MEDA. By comparing KEDAS with MEDA, it can be observed that the application of the knowledge-guided optimisation strategies greatly enhances EDA's search ability for excellent solutions. In general, the overall performance of MEDA is worse than NSGA II, NSGA III and RVEA for the problem instances considered. Meanwhile, Figure 8(d) provides a comparison of the average runtime (millisecond) per 100 iterations. MEDA has the shortest execution time for all selected instances. Although KEDAS has higher computational complexity, mainly due to the complexity of the VNS adopted, its average runtime is still shorter compared to the four other algorithms on all selected instances.

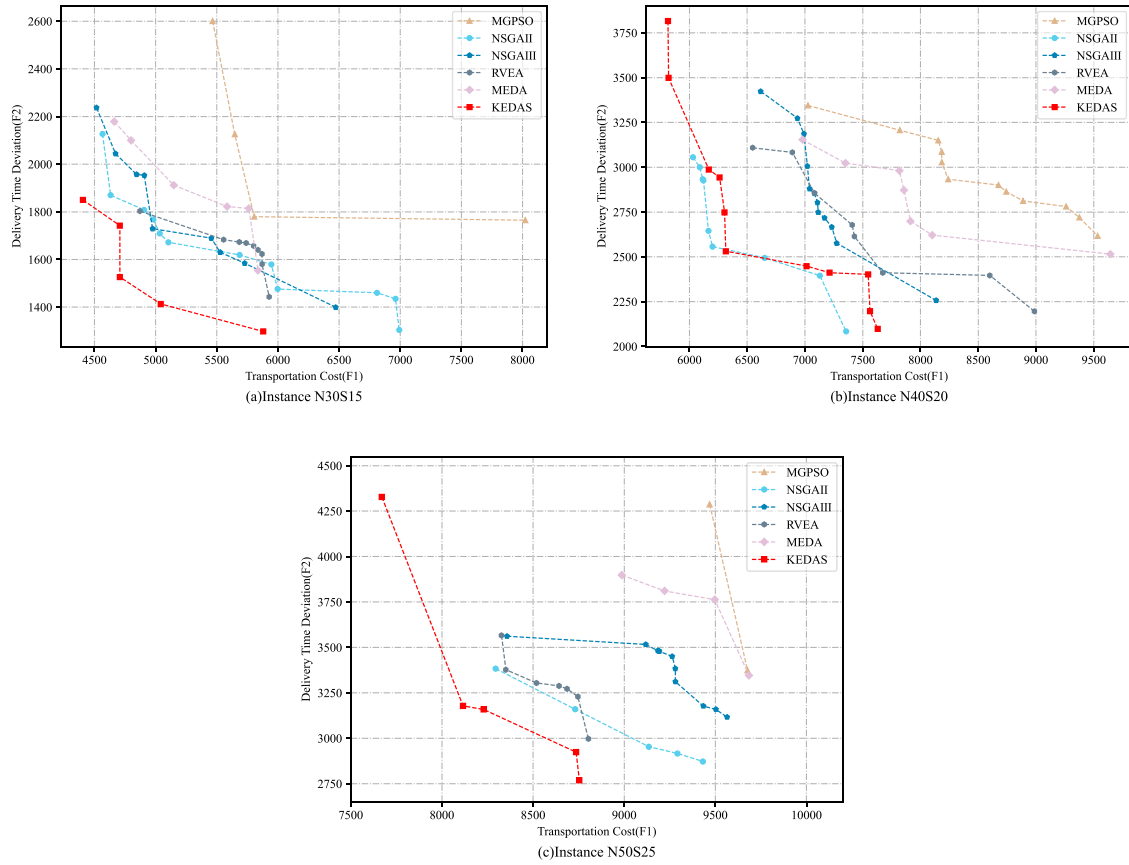


Figure 7. Pareto fronts obtained by compared algorithms on instances N30S15, N40S20 and N50S25.

#### 5.4. Discussion of results

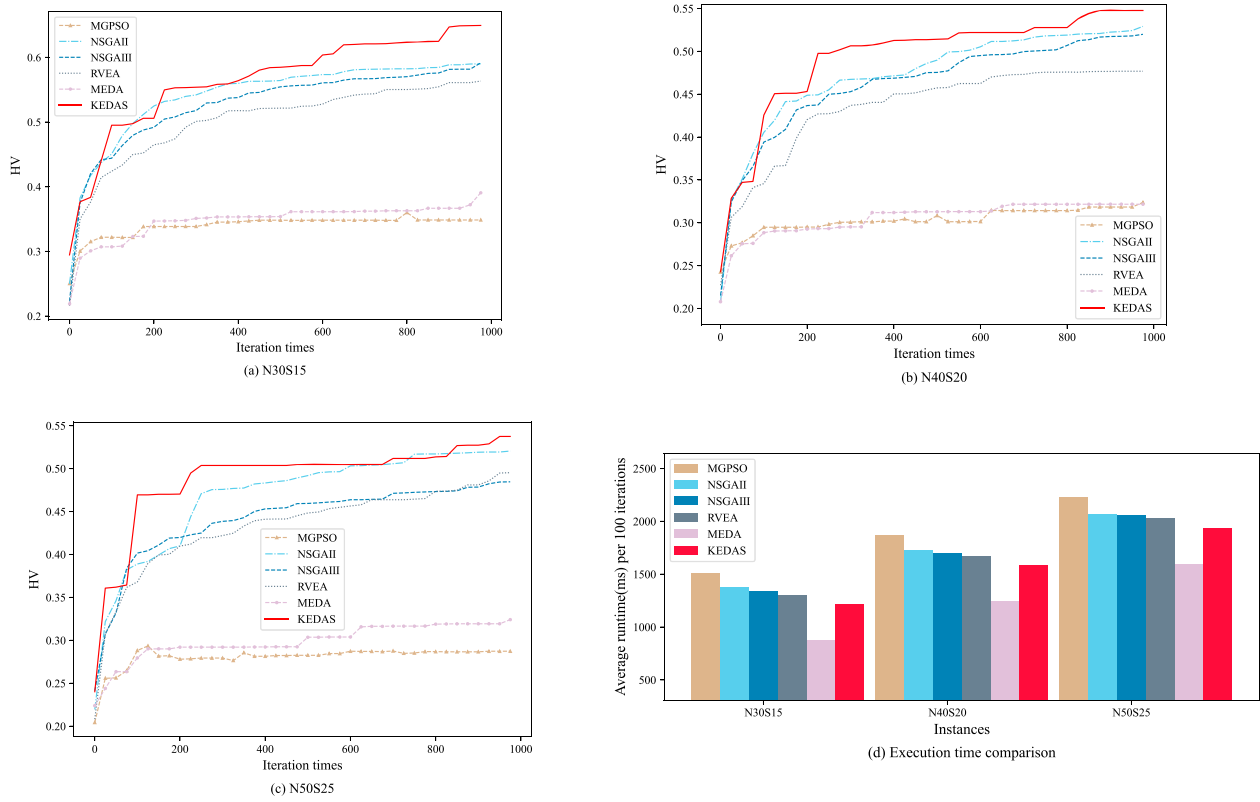
Above analysis confirmed that KEDAS has the potential to outperform alternative algorithms from the literature. Being motivated by practical needs, our problem combines the features of multi-objective discrete optimisation and AGVDP with fuzzy soft time window. This problem is hard to solve for alternative algorithms, which optimisation capabilities mainly depend on evolutionary operators that are unrelated to specific problem characteristics. Algorithms such as MGPSO and MEDA easily lose the optimisation direction and then fall into the local optimum trap, as highlighted in the test results. KEDAS benefits from knowledge-guided evolutionary strategies, i.e. problem feature-based strategies for population initialisation and local search, and algorithm evolutionary information-based strategies for offspring generation. This allows KEDAS to keep a more robust optimisation direction in the process of solving problem instances. These knowledge guided strategies of KEDAS can easily be implemented into a general knowledge guided optimisation framework for the algorithm design of other related problems.

Regarding real-life decision-making on the shop floor, KEDAS not only effectively solves the AGVDP considered in this paper, but also provides a set of AGV dispatching schemes for production managers to choose. As an example, Table 7 lists the AGVs dispatching scheme of non-dominated solutions for instance N30S30 obtained by a run of KEDAS. If managers rank transportation cost as the primary consideration, then solution 1 with the minimum objective value of F1 should be selected. If the delivery punctuality is management's main concern, then solution 7 with the minimum F2 value is preferred. If production managers are more concerned with the overall delivery satisfaction of all material orders, then solution 5 with the maximum of mean delivery satisfaction  $\bar{SA}$  can be regarded as the best scheme, since it provides a reasonable trade-off between the two optimisation objectives.

#### 6. Conclusion

This paper proposed a new solution approach for the multiple AGVs dispatching problem (AGVDP) for materials replenishment in a general manufacturing





**Figure 8.** Convergence trend curves of algorithms over iteration and execution time comparison.

**Table 7.** AGVs dispatching schemes obtained by KEDAS for instance N30S30.

Solution index	$F_1$	$F_2$	$\overline{SA}$	AGVs dispatching scheme	Number of AGVs used
1	<b>4937</b>	1523	0.165	[0, 9, 1, 28, 16, 23, 4, 0, 10, 2, 22, 13, 8, 24, 0, 5, 11, 21, 3, 0, 7, 25, 15, 17, 14, 0, 27, 29, 30, 18, 20, 26, 0, 6, 12, 19]	6
2	5028	1479	0.194	[0, 9, 1, 28, 16, 23, 4, 0, 3, 2, 22, 13, 8, 24, 10, 0, 12, 5, 11, 0, 7, 25, 15, 17, 14, 0, 27, 29, 30, 18, 20, 0, 6, 26, 21, 19]	6
3	5089	1363	0.136	[0, 1, 15, 23, 26, 19, 0, 25, 9, 10, 11, 0, 7, 28, 13, 8, 24, 4, 0, 27, 16, 17, 2, 22, 3, 0, 29, 30, 18, 21, 20, 14, 0, 12, 5, 6]	6
4	5317	1309	0.223	[0, 25, 9, 16, 17, 13, 0, 29, 23, 18, 10, 3, 24, 26, 0, 20, 6, 11, 4, 0, 19, 27, 30, 2, 22, 14, 0, 5, 12, 7, 15, 0, 1, 28, 21, 8]	6
5	5461	1286	<b>0.243</b>	[0, 19, 2, 10, 20, 17, 3, 23, 0, 15, 14, 6, 11, 0, 7, 12, 16, 22, 13, 0, 5, 25, 30, 18, 21, 4, 0, 27, 1, 9, 0, 28, 29, 8, 24, 26]	6
6	5608	1252	0.231	[0, 1, 28, 24, 17, 19, 0, 7, 30, 10, 2, 14, 0, 25, 27, 26, 11, 8, 4, 0, 22, 16, 23, 9, 13, 0, 29, 15, 18, 21, 20, 3, 0, 12, 5, 6]	6
7	5685	<b>1163</b>	0.164	[0, 28, 17, 14, 26, 24, 4, 0, 5, 25, 27, 3, 20, 13, 0, 18, 30, 15, 11, 8, 0, 7, 29, 2, 19, 10, 21, 0, 22, 6, 0, 12, 1, 9, 16, 23]	6

workshop where workstations are in a matrix layout. Driven by observations in practice, our study extended the existing literature by considering stochastic factors that impact solution quality, such as uncertainty in replenishment time and fluctuation in AGV unloading efficiency. To accommodate uncertainty in replenishment times, we introduced a fuzzy soft time-window, which is used to define the delivery satisfaction constraint of the problem. To accommodate the fluctuation of AGV unloading efficiency, we designed a heuristic method that uses the known normal probability distribution to estimate the AGV unloading duration at each workstation. To minimise both the transportation cost and the delivery time deviation of material orders, the considered AGVDP was formulated as a multi-objective mixed integer optimisation model.

A new knowledge-guided estimation of distribution algorithm with delivery satisfaction evaluation (KEDAS) was then proposed for solving this model. KEDAS utilises three knowledge-guided strategies to enhance the optimisation capabilities at its respective execution stages. It includes dispatching rule-guided population initialisation, pareto evolutionary performance-guided probability model update for offspring generation, and delivery satisfaction-guided variable neighbourhood search. To validate the proposed model and algorithm, comprehensive numerical experiments were conducted using instances based on data from a real-life intelligent manufacturing workshop. Results demonstrate that introducing a delivery satisfaction constraint can improve the quality of results. Meanwhile, KEDAS was

shown to outperform three popular multi-objective algorithms (NSGA II, NSGA III and RVEA), MGPSO, a relatively new algorithm for closely related problems, and a multi-objective version of EDA. Finally, results validate the effectiveness of the knowledge-guided strategies used for improving EDA, which provides important insights for future algorithm design.

Our problem model can obtain more robust solution schemes than existing models by considering two important operational dynamic factors that have been ignored by existing studies on AGVDP. Different from single-objective solution methods presented in related studies, the multi-objective optimisation solution approach adopted in this study avoids the need to determine proper weights for multiple indicators. It can also provide a set of options for managers to make adaptive decisions. The proposed algorithm KEDAS outperforms related popular algorithms. However, if optimal history AGVs dispatching scheme data is available, then the EDA offers a convenient way to introduce expert knowledge via its probability model. Future work can design a high-performance probability model of EDA by mining history scheme data to further enhance the proposed algorithm in this paper.

A first limitation of our study is the focus on only two factors that can impact solution performance. While this is justified by the need to keep our study reasonable focused, future research could consider other important problem characteristics, such as AGVs delivering materials and picking up products simultaneously, AGV charging demand, heterogeneous materials, availability of AGVs or AGV energy consumption. Another main limitation is that the mean delivery satisfactions for the problem instances were not very high in our experiments. This indicates that a single solution method cannot well cope with the complexity of the problem under study. Future research should consider integrated solution methods, including multi-stage programming methods and hybrid optimisation algorithms, to further improve results. Future research could also try to formulate the AGVDP as a robust or stochastic optimisation model and compare its results with the results obtained in this study.

### Disclosure statement

No potential conflict of interest was reported by the author(s).

### Data availability statement

The data instances that support the findings of this study are openly available in *science data bank* at <https://www.scidb.cn/en/s/7VNFVf>, and data 10.11922/sciencedb.01412.

### Funding

This work was supported by National Natural Science Foundation of China: [Grant Number 51875251, 71872072, 7215061 0504]; National Key Research and Development Program of China: [Grant Number 2021YFB3301701]; 2019 Guangdong Special Support Talent Program Innovation and Entrepreneurship Leading Team: [Grant Number 2019BT02S593].

### Notes on contributors



**Lei Liu** is currently a Ph.D. Student in Management Science and Engineering at the School of Management, Jinan University (Guangzhou, PR China). He received a Master degree in industrial engineering from Chongqing University of Post and Telecommunications (Chongqing, PR China) in 2019 and a Bachelor degree in Manufacturing and Automation from Northwest A&F University (Yangling, PR China) in 2014. Lei Liu worked as a big data R&D engineer from 2019 to 2020 at GREE Electric Appliances, a domestic appliance manufacturer and the world's largest air conditioner producer. His research interests include data and knowledge-driven operation decision making and optimisation for complex industrial systems, intelligent algorithms, and intelligent manufacturing for production-logistics synchronisation, production workload control, modular manufacturing systems and digital twins.



**Matthias Thüerer** is Distinguished Professor in Management Science and Engineering at Jinan University (Zhuhai, PR China). Before getting involved in academia, Matthias worked in several companies, did an apprenticeship and became a master craftsman ('Meister'). He contributed to the improvement, simplification and integration of material flow control systems, and their integration with Industry 4.0. Apart from Operations Management, Matthias is also interested in social and philosophical issues including system theory, cybernetics, causality and the philosophy of science.



**Ting Qu** is a full professor at the School of Intelligent Systems Science and Engineering, Jinan University (Zhuhai, PR China). He received his BEng and MPhil degrees from the School of Mechanical Engineering of Xi'an Jiaotong University (China) and his Ph.D. degree from the Department of Industrial and Manufacturing Systems Engineering of The University of Hong Kong. His research interests include IoT-based smart manufacturing systems, logistics and supply chain management, and industrial product/production service systems. He has undertaken over twenty research projects funded by government and industry and has published nearly 200 technical papers in these areas, half of which have appeared in reputable journals. He serves as director or board member for several academic associations in industrial engineering and smart manufacturing.



**Lin Ma** is a Ph.D. Student in Management Science and Engineering at the School of Management, Jinan University (Guangzhou, PR China). He holds a Master from Xi'an University of Science and Technology (Xi'an, PR China). Production Bottleneck Management for high variety make-to-order shops is one of his main research interests. He is also interested in intelligent manufacturing, including digital twins and production-logistics synchronisation.



**Zhongfei Zhang** is a Ph.D. candidate at the School of Management, Jinan University (Guangzhou, PR China). He received his BEng degrees and MPhil degrees in Mechanical Engineering from the School of Electromechanical Engineering of Zhengzhou University of Aeronautics (China) and Zhengzhou University of Light Industry (China) in 2014 and 2018, respectively. His research interests include cloud manufacturing systems, blockchain-enabled production logistics and intelligent synchronisation decision-making.



**Mingze Yuan** is a Master Student in Management Science and Engineering at the School of Management, Jinan University (Guangzhou, PR China). Her main research interests include Workload Control, order release mechanisms for high-variety make-to-order shops and Industry 4.0.

## References

- Ahmadi-Javid, Amir, and Amir Ardestani-Jaafari. 2021. "The Unequal Area Facility Layout Problem with Shortest Single-Loop AGV Path: How Material Handling Method Matters." *International Journal of Production Research* 59 (8): 2352–2374.
- Bae, Jungyun, and Woojin Chung. 2017. "A Heuristic for a Heterogeneous Automated Guided Vehicle Routing Problem." *International Journal of Precision Engineering and Manufacturing* 18 (6): 795–801.
- Bandaru, S., A. H. C. Ng, and K. Deb. 2017. "Data Mining Methods for Knowledge Discovery in Multi-Objective Optimization: Part B – New Developments and Applications." *Expert Systems with Applications* 70: 119–138.
- Barba-Gonzalez, Cristobal, Antonio J. Nebro, Jose Garcia-Nieto, Maria del Mar Roldan-Garcia, Ismael Navas-Delgado, and Jose F. Aldana-Montes. 2021. "Injecting Domain Knowledge in Multi-Objective Optimization Problems: A Semantic Approach." *Computer Standards & Interfaces* 78: 103546.
- Beume, Nicola, Boris Naujoks, and Michael Emmerich. 2007. "SMS-EMOA: Multiobjective Selection Based on Dominated Hypervolume." *European Journal of Operational Research* 181 (3): 1653–1669.
- Bilyk, A., and L. Monch. 2012. "A Variable Neighborhood Search Approach for Planning and Scheduling of Jobs on Unrelated Parallel Machines." *Journal of Intelligent Manufacturing* 23 (5): 1621–1635.
- Cheng, Ran, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. 2016. "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization." *Ieee Transactions on Evolutionary Computation* 20 (5): 773–791.
- Deng, Wu, Xiaoxiao Zhang, Yongquan Zhou, Yi Liu, Xiangbing Zhou, Huiling Chen, and Huimin Zhao. 2022. "An Enhanced Fast non-Dominated Solution Sorting Genetic Algorithm for Multi-Objective Problems." *Information Sciences* 585: 441–453.
- Desaulniers, Guy, André Langevin, Diane Riopel, and Bryan Villeneuve. 2003. "Dispatching and Conflict-Free Routing of Automated Guided Vehicles: An Exact Approach." *International Journal of Flexible Manufacturing Systems* 15 (4): 309–331.
- Duan, Wenzhe, Zhengyang Li, Mengchen Ji, Yixin Yang, Shouyang Tang, Bo Liu, and Ieee. 2016. "A Hybrid Estimation of Distribution Algorithm for Distributed Permutation Flowshop Scheduling with Flowline Eligibility." Paper presented at the IEEE congress on evolutionary computation (CEC) held as part of IEEE world congress on computational intelligence (IEEE WCCI), Vancouver, Canada, Jul 24–29.
- Fang, Yilin, Hao Ming, Miqing Li, Quan Liu, and Pham Duc Truong. 2020. "Multi-objective Evolutionary Simulated Annealing Optimisation for Mixed-Model Multi-Robotic Disassembly Line Balancing with Interval Processing Time." *International Journal of Production Research* 58 (3): 846–862.
- Fazlollahtabar, H., and S. Hassanli. 2018. "Hybrid Cost and Time Path Planning for Multiple Autonomous Guided Vehicles." *Applied Intelligence* 48 (2): 482–498.
- Fazlollahtabar, Hamed, and Mohammad Saidi-Mehrabad. 2015. "Methodologies to Optimize Automated Guided Vehicle Scheduling and Routing Problems: A Review Study." *Journal of Intelligent & Robotic Systems* 77 (3–4): 525–545.
- Fazlollahtabar, Hamed, Mohammad Saidi-Mehrabad, and Jaydeep Balakrishnan. 2015. "Mathematical Optimization for Earliness/Tardiness Minimization in a Multiple Automated Guided Vehicle Manufacturing System via Integrated Heuristic Algorithms." *Robotics and Autonomous Systems* 72: 131–138.
- Guerreiro, Andreia P., Vasco Manquinho, and Jose Rui Figueira. 2021. "Exact Hypervolume Subset Selection Through Incremental Computations." *Computers & Operations Research* 136: 105471.
- Guo, Wei, Pingyu Jiang, and Maolin Yang. 2022. "Unequal Area Facility Layout Problem Solving: A Real Case Study on an air-Conditioner Production Shop Floor." *International Journal of Production Research* 1–18.
- Guo, Jun, Zhipeng Pu, Baigang Du, and Yibing Li. 2022. "Multi-objective Optimisation of Stochastic Hybrid Production Line Balancing Including Assembly and Disassembly Tasks." *International Journal of Production Research* 60 (9): 2884–2900.
- Heyong, Wang, and Hong Ming. 2019. "Supervised Hebb Rule Based Feature Selection for Text Classification." *Information Processing & Management* 56 (1): 167–191.
- Hu, Hongtao, Xurui Yang, Shichang Xiao, and Feiyang Wang. 2021. "Anti-conflict AGV Path Planning in Automated Container Terminals Based on Multi-Agent Reinforcement Learning." *International Journal of Production Research* 1–16.
- Hwang, Illhoe, and Young Jae Jang. 2020. "Q(Lambda) Learning-Based Dynamic Route Guidance Algorithm for



- Overhead Hoist Transport Systems in Semiconductor Fabs.” *International Journal of Production Research* 58 (4): 1199–1221.
- Jazzbin. 2021. “Geatpy: The Genetic and Evolutionary Algorithm Toolbox with High Performance in Python.” <http://www.geatpy.com/>.
- Kinderman, Albert J., and John F. %J ACM Transactions on Mathematical Software Monahan. 1977. “Computer Generation of Random Variables Using the Ratio of Uniform Deviates.” *ACM Transactions on Mathematical Software* 3 (3): 257–260.
- Krishnamurthy, Nirup N., Rajan Batta, and Mark H. Karwan. 1993. “Developing Conflict-Free Routes for Automated Guided Vehicles.” *Operations Research* 41 (6): 1077–1090.
- Larrañaga, Pedro, and Jose A. Lozano. 2001. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Vol. 2. Springer Science & Business Media.
- Lee, Chen Wei, Wai Peng Wong, Joshua Ignatius, Amirah Rahman, and Ming-Lang Tseng. 2020. “Winner Determination Problem in Multiple Automated Guided Vehicle Considering Cost and Flexibility.” *Computers & Industrial Engineering* 142: 106337.
- Li, Guomin, Xinyu Li, Liang Gao, and Bing Zeng. 2019. “Tasks Assigning and Sequencing of Multiple AGVs Based on an Improved Harmony Search Algorithm.” *Journal of Ambient Intelligence and Humanized Computing* 10 (11): 4533–4546.
- Li, Maojia P., Prashant Sankaran, Michael E. Kuhl, Raymond Ptucha, Amlan Ganguly, Andres Kwasinski, and Ieee. 2019. “Task Selection by Autonomous Mobile Robots in a Warehouse Using Deep Reinforcement Learning.” Paper presented at the winter simulation conference (WSC), National Harbor, MD, Dec 8–11.
- Li, Xijun, Mingxuan Yuan, Di Chen, Jianguo Yao, Jia Zeng, and Acm. 2018. “A Data-Driven Three-Layer Algorithm for Split Delivery Vehicle Routing Problem with 3D Container Loading Constraint.” Paper presented at the Kdd’18: proceedings of the 24th Acm sigkdd international conference on knowledge discovery & data mining, London, United Kingdom, August 19–23.
- Malopolski, W. 2018. “A Sustainable and Conflict-Free Operation of AGVs in a Square Topology.” *Computers & Industrial Engineering* 126: 472–481.
- Miyamoto, Toshiyuki, and Kensuke Inoue. 2016. “Local and Random Searches for Dispatch and Conflict-Free Routing Problem of Capacitated AGV Systems.” *Computers & Industrial Engineering* 91: 1–9.
- Mousavi, Maryam, Hwa Jen Yap, Siti Nurmaya Musa, Farzad Tahiri, and Siti Zawiah Md Dawal. 2017. “Multi-objective AGV Scheduling in an FMS Using a Hybrid of Genetic Algorithm and Particle Swarm Optimization.” *PLOS ONE* 12 (3): e0169817.
- Murakami, Keisuke. 2020. “Time-space Network Model and MILP Formulation of the Conflict-Free Routing Problem of a Capacitated AGV System.” *Computers & Industrial Engineering* 141: 106270.
- Nishi, Tatsushi, Yuichiro Hiranaka, and Ignacio E. Grossmann. 2011. “A Bilevel Decomposition Algorithm for Simultaneous Production Scheduling and Conflict-Free Routing for Automated Guided Vehicles.” *Computers & Operations Research* 38 (5): 876–888.
- Nishida, Kosei, and Tatsushi Nishi. 2022. “Dynamic Optimization of Conflict-Free Routing of Automated Guided Vehicles for Just-in-Time Delivery.” *Ieee Transactions on Automation Science and Engineering* 1–16.
- Qin, Wei, Zilong Zhuang, Yaoming Zhou, and Yinbin Sun. 2021. “Dynamic Dispatching for Interbay Automated Material Handling with lot Targeting Using Improved Parallel Multiple-Objective Genetic Algorithm.” *Computers & Operations Research* 131: 105264.
- Riazi, Sarmad, Thomas Diding, Petter Falkman, Kristofer Bengtsson, and Bengt Lennartson. 2019. “Scheduling and Routing of AGVs for Large-Scale Flexible Manufacturing Systems.” Paper presented at the 15th IEEE international conference on automation science and engineering (IEEE CASE), Vancouver, Canada, Aug 22–26.
- Sathiya, V., M. Chinnadurai, S. Ramabalan, and Andrea Appoloni. 2021. “Mobile Robots and Evolutionary Optimization Algorithms for Green Supply Chain Management in a Used-car Resale Company.” *Environment, Development and Sustainability* 23 (6): 9110–9138.
- Scheepers, Christiaan, Andries P. Engelbrecht, and Christopher W. Cleghorn. 2019. “Multi-guide Particle Swarm Optimization for Multi-Objective Optimization: Empirical and Stability Analysis.” *Swarm Intelligence* 13 (3–4): 245–276.
- Singh, Nitish, Quang-Vinh Dang, Alp Akcay, Ivo Adan, and Tugce Martagan. 2022. “A Matheuristic for AGV Scheduling with Battery Constraints.” *European Journal of Operational Research* 298 (3): 855–873.
- Sinriech, D., and J. Kotlarski. 2002. “A Dynamic Scheduling Algorithm for a Multiple-Load Multiple-Carrier System.” *International Journal of Production Research* 40 (5): 1065–1080.
- Subbaiah, K. V., M. Nageswara Rao, and K. Narayana Rao. 2009. “Scheduling of AGVs and Machines in FMS with Makespan Criteria Using Sheep Flock Heredity Algorithm.” *International Journal of the Physical Sciences* 4 (3): 139–148.
- Subramanian, Senthilkumar, Chandramohan Sankaralingam, Rajvikram Madurai Elavarasan, Raghavendra Rajan Vijayaraghavan, Kannadasan Raju, and Lucian Mihet-Popa. 2021. “An Evaluation on Wind Energy Potential Using Multi-Objective Optimization Based Non-Dominated Sorting Genetic Algorithm III.” *Sustainability* 13 (1): 410.
- Sun, Yu, and Haisheng Li. 2020. “An End-to-End Reinforcement Learning Method for Automated Guided Vehicle Path Planning.” Paper presented at the international symposium on artificial intelligence and robotics 2020.
- Tsolakis, Naoum, Dimitris Zissis, Spiros Papaefthimiou, and Nikolaos Korfiatis. 2022. “Towards AI Driven Environmental Sustainability: An Application of Automated Logistics in Container Port Terminals.” *International Journal of Production Research* 60 (14): 4508–4528.
- Umar, U. A., M. K. A. Ariffin, N. Ismail, and S. H. Tang. 2015. “Hybrid Multiobjective Genetic Algorithms for Integrated Dynamic Scheduling and Routing of Jobs and Automated-Guided Vehicle (AGV) in Flexible Manufacturing Systems (FMS) Environment.” *The International Journal of Advanced Manufacturing Technology* 81 (9–12): 2123–2141.
- Wang, Sheng-yao, Ling Wang, Min Liu, and Ye Xu. 2013. “An Effective Estimation of Distribution Algorithm for Solving the Distributed Permutation Flow-Shop Scheduling Problem.” *International Journal of Production Economics* 145 (1): 387–396.

- Wang, Zehao, and Qingcheng Zeng. 2022. "A Branch-and-Bound Approach for AGV Dispatching and Routing Problems in Automated Container Terminals." *Computers & Industrial Engineering* 166: 107968.
- Xu, Wenxiang, Shunsheng Guo, Xixing Li, Chen Guo, Rui Wu, and Zhao Peng. 2019. "A Dynamic Scheduling Method for Logistics Tasks Oriented to Intelligent Manufacturing Workshop." *Mathematical Problems in Engineering* 2019: 1–18.
- Yao, Fengjia, Bugra Alkan, Bilal Ahmad, and Robert Harrison. 2020. "Improving Just-in-Time Delivery Performance of IoT-Enabled Flexible Manufacturing Systems with AGV Based Material Transportation." *Sensors* 20 (21): 6333.
- Yu, Guo, Yaochu Jin, and Markus Olhofer. 2021. "A Multi-Objective Evolutionary Algorithm for Finding Knee Regions Using Two Localized Dominance Relationships." *Ieee Transactions on Evolutionary Computation* 25 (1): 145–158.
- Zhang, B., Q. K. Pan, L. Gao, L. L. Meng, X. Y. Li, and K. K. Peng. 2020. "A Three-Stage Multiobjective Approach Based on Decomposition for an Energy-Efficient Hybrid Flow Shop Scheduling Problem." *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50 (12):4984–4999.
- Zhang, Zi-Qi, Bin Qian, Rong Hu, Huai-Ping Jin, and Ling Wang. 2021. "A Matrix-Cube-Based Estimation of Distribution Algorithm for the Distributed Assembly Permutation Flow-Shop Scheduling Problem." *Swarm and Evolutionary Computation* 60: 100785.
- Zhang, Xu-jin, Hong-yan Sang, Jun-qing Li, Yu-yan Han, and Peng Duan. 2022. "An Effective Multi-AGVs Dispatching Method Applied to Matrix Manufacturing Workshop." *Computers & Industrial Engineering* 163: 107791.
- Zhou, Xiuling, Ping Guo, C. L. Philip Chen, and Ieee. 2012. "An Algorithm for Calculating the Hypervolume Contribution of a Set." Paper presented at the 2012 world automation congress (Wac), Yichang, China, Nov 20–22.
- Zhou, Binghai, and Zhaoxu He. 2021. "A Novel Hybrid-Load AGV for JIT-Based Sustainable Material Handling Scheduling with Time Window in Mixed-Model Assembly Line." *International Journal of Production Research* 1–22.
- Zitzler, E., and L. Thiele. 1999. "Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach." *Ieee Transactions on Evolutionary Computation* 3 (4): 257–271.
- Zou, Wen-Qiang, Quan-Ke Pan, Tao Meng, Liang Gao, and Yu-Long Wang. 2020. "An Effective Discrete Artificial bee Colony Algorithm for Multi-AGVs Dispatching Problem in a Matrix Manufacturing Workshop." *Expert Systems with Applications* 161: 113675.