Flexible time–space network formulation and hybrid metaheuristic for conflict-free and energy-efficient path planning of automated guided vehicles

Xin, Jianbin; Wei, Liuqian; D'Ariano, Andrea; Zhang, Fangfang; Negenborn, Rudy

DOI
10.1016/j.jclepro.2023.136472

Publication date
2023

Document Version
Final published version

Published in
Journal of Cleaner Production

Citation (APA)
Xin, J., Wei, L., D'Ariano, A., Zhang, F., & Negenborn, R. (2023). Flexible time–space network formulation and hybrid metaheuristic for conflict-free and energy-efficient path planning of automated guided vehicles. *Journal of Cleaner Production*, *398*, Article 136472. https://doi.org/10.1016/j.jclepro.2023.136472

Journal of Cleaner Production 398 (2023) 136472

Contents lists available at ScienceDirect

Journal of Cleaner Production

journal homepage: www.elsevier.com/locate/jclepro

Flexible time–space network formulation and hybrid metaheuristic for conflict-free and energy-efficient path planning of automated guided vehicles

Jianbin Xin [a], Liuqian Wei [a], Andrea D'Ariano [b], Fangfang Zhang [a,*], Rudy Negenborn [c]

[a] School of Electrical and Information Engineering, Zhengzhou University, No. 100, Science Road, Zhengzhou 450001, PR China
[b] Dipartimento di Ingegneria, Università Degli Studi Roma Tre, via della Vasca Navale, 79 00146 Roma, Italy
[c] Department of Maritime and Transport Technology, Delft University of Technology, Mekelweg 2, 2628CD, Delft, The Netherlands

ARTICLE INFO

Handling Editor: Kathleen Aviso

Keywords:
Automated guided vehicles
Energy efficiency
Flexible time–space network model
Hybrid metaheuristic

ABSTRACT

Operations of Automated Guided Vehicles (AGVs) are desired to be more energy-efficient while maintaining high transport productivity, motivated by the green production requirements. This paper investigates a new energy-efficient planning problem for determining conflict-free paths of the AGVs in its transport roadmap. In this problem, the vehicle path and transport time in the roadmap are jointly optimized, based on a flexible time–space network (FTSN). We provide the mathematical problem formulation of the energy-efficient path planning problem. The resulting optimization problem is proved to be a non-convex mixed-integer nonlinear programming which is computationally intractable. We further propose a hybrid metaheuristic that integrates the genetic algorithm and estimation of the distribution algorithm to improve its computational efficiency. Numerical results show the effectiveness of the developed algorithm based on the FTSN framework, compared to the existing metaheuristics, the conventional path planning method, and a commercial solver. The proposed method has a wide application in improving energy use of material handling, providing a guiding significance on promoting cleaner production of flexible manufacturing systems.

1. Introduction

Automated Guided Vehicles (AGVs) are autonomous vehicles designed for horizontal movement of materials. Due to the advantages of productivity improvements, labor cost savings, and environmental benefits, AGVs have been widely used in manufacturing (He et al., 2022; Cai et al., 2022), warehouse (Hu et al., 2017), container terminals (Xin et al., 2015b; Yue et al., 2021) and other applications. A market report forecasts fast growth in the AGV market from 2018 to 2025 (Ryck et al., 2020).

The efficiency of the AGVs systems is closely related to the performance of the production and logistics systems. For the company managers, the minimization of the completion time is the primary objective. Producers aim to maximize the production rates to meet the increasing requirements made by customers. High production rates result from optimizing the completion time of the AGVs when finishing the assigned tasks.

The energy-aware path planning for the operations of multiple AGVs is motivated by green production requirements (Bechtsis et al., 2017; van Duin et al., 2018; Rinaldi et al., 2020). However, there are several challenges to be addressed for this energy-aware planning problem. The first challenge is that two objectives (time-related objective and energy-related objective) must be considered at the same time. The transport process involves both event-driven dynamics and time-driven dynamics (so-called hybrid dynamical system) for minimizing the completion time and the energy consumption (Xin et al., 2014). To search for the energy-efficient paths of the AGVs, a mathematical model that is capable to represent these two types of dynamics needs to be formulated carefully, and then the AGV can be better coordinated by accelerating and decelerating properly for energy saving. Furthermore, the energy consumption representation in practice is nonlinear and the resulting optimization problem is mixed-integer nonlinear programming (MINLP), with consideration of the path sequencing in the discretized AGV roadmap. The MINLP in general is more challenging to be solved than mixed integer programming (MIP) as this is commonly seen in path planning problems.

In this paper, we focus on improving the energy efficiency of the AGV operations, *i.e., using less energy to achieve the same level of service*. The energy saving objective is less important than the time saving objective. The latter is the primary objective because the business performance is essentially determined by productivity (Xin et al., 2014). To maximize the profit, the producers tend to reduce AGV's energy

* Corresponding author.
E-mail addresses: j.xin@zzu.edu.cn (J. Xin), zhangfangfang@zzu.edu.cn (F. Zhang).

https://doi.org/10.1016/j.jclepro.2023.136472
Received 26 July 2021; Received in revised form 13 February 2023; Accepted 15 February 2023
Available online 22 February 2023

consumption without deteriorating the service of high productivity. Therefore, energy consumption is expected to be reduced without deteriorating the time-related performance to further decrease the operating costs. To improve the energy efficiency of the AGVs when planning their paths for transporting materials, we propose a flexible time–space network model that is formulated as a dedicated MINLP.

1.1. Related work

This section reviews the routing of multiple AGVs and the related energy saving strategies. Then, the approaches based on the flexible time–space network model are also reviewed.

1.1.1. Routing of multiple AGVs

For the AGVs, operational planning problems are further classified into two directions: task allocation and path planning. The task allocation problem seeks optimal assignments and sequences of tasks that are carried out by AGVs while satisfying resource constraints (Zou et al., 2020). When the task involves a time-related objective, the task allocation problem is regarded as a scheduling problem. Energy-efficient scheduling problems have been extensively investigated in the manufacturing systems (Barak et al., 2020). For a path planning problem, feasible paths are determined to avoid collisions and deadlocks due to safety and space constraints.

The scheduling problem and the path planning problem are closely coupled, and some researchers investigate how to integrate these two problems to improve the overall performance of multiple AGVs (Saidi-Mehrabad et al., 2015; Xin et al., 2015a; Miyamoto and Inoue, 2016). For instance, Miyamoto and Inoue (2016) consider dispatching and routing decisions simultaneously by considering that each vehicle has a finite capacity. However, driven by future requirements like flexibility, robustness, and scalability, the current trend for planning the AGVs is decentralization (Ryck et al., 2020). Decentralized methods decompose the overall problem into smaller subproblems, in which scheduling and routing decisions are formulated hierarchically (Demesure et al., 2018; Fanti et al., 2018), where the scheduling problem is solved at a higher level, while the path planning is formulated at a lower level. Here, we focus on the path planning problem.

When using the AGV, the mesh routing environment is a common layout to configure the AGV path in production and logistics (Yi et al., 2019; Nishi et al., 2020; Polten and Emde, 2021). The mesh routing environment uses fixed path guided vehicles to simplify the planning of multiple vehicles, and it is suitable for limited working areas (Miyamoto and Inoue, 2016). When the mesh routing is considered, time–space network (TSN) models have been proposed for determining feasible conflict-free vehicle paths by modeling time constraints and space constraints (Miyamoto and Inoue, 2016; Nishi et al., 2020). In these TSN-based path planning problems, a single objective problem is typically considered to minimize some productivity indices (e.g., the completion time). These TSN models take into account a fixed transport time for two adjacent nodes, which may lose possible opportunities for reducing energy consumption by changing these transport times. Each AGV must wait at a certain node to avoid collisions with other AGVs that may occupy its driving path, this collision avoidance strategy can lead to an increase in the completion time of some tasks.

1.1.2. Energy saving when routing the AGVs

Compared to the extensive research that targets the completion time minimization, little attention has been paid to the energy-aware path planning for the operations of multiple AGVs (Adamo et al., 2018; Riazi et al., 2020). Here, we review these works. In Adamo et al. (2018), a conflict-free pick-up and delivery problem with time windows is studied to optimize the path and speed of AGVs. A branch-and-bound algorithm is developed to determine the path and the speed of AGVs on each arc, by satisfying the time window and minimizing the total energy consumption. However, a mathematical formulation for energy-aware path planning of AGVs is still missing, and the relationship between these two objectives is not investigated.

Riazi et al. studied the energy minimization problem for large-scale AGV systems (Riazi et al., 2020). They identified that travel speed and distance are the most important factors that affect the energy consumption of the AGV system. In Riazi et al. (2020), energy minimization is considered separately after determining the path of AGVs, and only the effect of the maximum speed on the objectives is analyzed. Therefore, it is still unclear how the AGV speed and the path should be optimized jointly to improve the energy efficiency of its path planning.

1.1.3. Flexible time–space network model

TSN models have been widely used for modeling routing problems in the transportation and logistics systems (Nishi et al., 2005; Miyamoto and Inoue, 2016; Nishi et al., 2020). In such TSN models, a transportation network (or roadmap) is considered to optimize the routes of vehicles, and the transport time between every two adjacent nodes is *fixed*.

In railway applications, a variant of the TSN model referred to as the flexible time–space network (FTSN) model, has been studied (Meng and Zhou, 2014; Zhang et al., 2019). In such an FTSN model, the transport time between the nodes of the railway network becomes a decision variable, allowing for jointly optimizing the routes and passing times at each station of the selected route, to improve their punctuality. In Meng and Zhou (2014), the FTSN model is built based on network cumulative flow variables. The objective is to minimize the total deviation time of all involved trains (without minimizing the energy consumption), and an integer programming problem is solved by Lagrangian relaxation. A systemic summary regarding these FTSN models is reported in Table 1 of Zhang et al. (2019).

It is observed that the existing FTSN models are specialized in railway applications and the optimization objective is limited to productivity metrics (such as total deviation time and the total journal time). Because the operations of logistics systems, such as AGVs, are considerably different from the railway applications, such a flexible modeling framework is not implemented yet for the AGVs to save energy. The energy-efficient path planning problem of AGVs using the FTSN modeling framework remains to be investigated further.

1.1.4. Overall assessment of the literature

AGVs are a crucial part of the production systems, and AGVs provide a range of benefits across environmental and social sustainability dimensions for reducing energy consumption and emissions (Bechtsis et al., 2017; Barak et al., 2020). The operations of AGVs must be energy-efficient to meet not only economic but also environmental criteria.

At the operational level, energy-efficient path planning of AGVs has not been sufficiently investigated in the literature. First of all, the current works cannot provide a mathematical formulation to reduce energy consumption while maintaining a high transport productivity. This type of mathematical formulation is needed to manage the relationship between the energy consumption and the total transport time. Then, the current FTSN models typically solve an MIP, while our FTSN model requires solving an MINLP to jointly minimize energy consumption and the sum of the completion times. Solving this challenging MINLP problem (resulting from the FTSN model) requires efficient and effective algorithms to be developed when introducing energy-related objectives and constraints.

1.2. Contributions of this paper

The contributions of this paper are given as follows:

- We propose a flexible time–space network model (FTSN) for energy-efficient and collision-free path planning of AGVs. The proposed FTSN model differs from the conventional TSN model that has fixed transport times, as proposed in Miyamoto and Inoue (2016) and Nishi et al. (2020). We extend the work in Yin and Xin (2019) that provides the basis to resolve collision conflicts and further study energy-efficient path planning of AGVs in the mesh roadmap environment.
- A mathematical representation is formulated to integrate the optimization of the operational speed when planning the conflict-free paths of AGVs. This optimization problem that uses the lexicographic strategy is formulated as an MINLP and the resulting MINLP is proved to be a non-convex formulation.
- A customized hybrid metaheuristic is developed to solve the studied MINLP problem. A new encoding scheme and customized algorithm procedures are proposed, taking into account the feature of the considered path planning problem. The developed metaheuristic provides higher quality solutions for solving the MINLP compared to the existing metaheuristics, such as Genetic Algorithm (GA) and Population-Based Incremental Learning (PBIL) (Tang et al., 2021; Wu and Wang, 2018).

The remainder of this paper is organized as follows: Section 2 introduces the FTSN model for describing the transport process of the AGVs. In Section 3, a customized hybrid metaheuristic is further proposed to determine energy-efficient paths of vehicles. Section 4 conducts case studies and analyzes the numerical results. Section 5 concludes this paper and provides future research directions.

2. Modeling framework

This section defines the research problem for planning paths of AGVs in the mesh routing environment. The FTSN model and framework are then introduced to determine energy-efficient and collision-free paths of AGVs.

2.1. Problem description

In manufacturing or warehouse systems, AGVs are used to transport materials between their origins and destinations in a mesh routing environment.

In the mesh routing environment (Ryck et al., 2020), nodes represent loading and unloading locations and intersections of the roadmap, each edge is a connected path between two adjacent nodes. Every task is defined for moving the material by a particular vehicle from their specific origin to destination.

Regarding this process, the following important assumptions are made:

- The size of each AGV is sufficiently small compared with the unit path length, regarding an AGV as a point.
- The number of vehicles is known and every transportation task is assigned to one AGV already.
- A pickup node (start node) and a delivery node (end node) of each task are given in advance. There are no duplicated nodes among the assigned tasks.
- Each AGV can transport a single load at any time.
- Each AGV can only stop at a network node and no more than one AGV can occupy a node at the same time.
- The turning time is included in the transport time.
- Each arc of the network can be occupied by one AGV at any time.

2.2. FTSN representation

This part gives the modeling framework based on the FTSN representation, which is used for planning the collision-free paths of AGVs in an energy-efficient way.

As mentioned in Section 1.1, the FTSN representation allows to optimize the transport time and the route jointly in a transportation network. For planning the path of the AGVs, this representation is different from the existing TSN modeling frameworks, which have fixed transport times in their roadmap network (Nishi et al., 2005; Miyamoto and Inoue, 2016). Because flexible transport times are considered in the FTSN representation, the operation speed of each AGV can be optimized with the route jointly to reduce the energy consumption of the AGVs. Here, we provide the mathematical formulation of the FTSN model based on the framework proposed in Meng and Zhou (2014).

In the FTSN representation, we consider a directed graph $G = (V, E)$ ($k \in \{1, 2, \dots, N\}$) for N AGVs. V is the collection of nodes, while $E = \{(i, j) | i \in V, j \in V\ i \neq j\}$ is the collection of arcs. Arc (i, j) maps the path from node i to j for two adjacent nodes. Each vehicle can change the path at node i if multiple nodes are connected to node i.

We consider a planning horizon $T \times \Delta t$ that is discretized equally into a set of time slots, denoted by $\{\Delta t, 2\Delta t, \dots, T \times \Delta t\}$. Δt is a time slot and T is the total number of time slots. As a result, the time–space network decomposes the overall routing process of multiple robots into several time slots. For each time instant $t \in \{1, 2, \dots, T\}$, each AGV can either stays at a node or move within the arc between two adjacent nodes. During each time interval, the detailed path will be considered to avoid collisions among these AGVs.

Before detailing the proposed FTSN model, the defined parameters and variables are introduced in Table 1. The FTSN model includes space constraints, time constraints, and time–space constraints; these three parts are next given.

Part 1: space constraints

$$\sum_{j \in N_{o_k}} x(o_k, j, k) = 1, \ \forall k \in \psi \tag{1}$$

$$\sum_{i \in N_j} x(i, j, k) = \sum_{n \in N_j} x(j, n, k), \forall k \in \psi, j \in V, j \neq o_k, j \neq s_k \tag{2}$$

$$\sum_{i \in N_{s_k}} x(i, s_k, k) = 1, \ \forall k \in \psi \tag{3}$$

Part 2: time–space constraints

$$d(i, j, k, t) \leq a(i, j, k, t), \quad \forall k \in \psi, (i, j) \in E, t \in \Phi \tag{4}$$

$$\sum_{i \in N_j} d(i, j, k, t) = \sum_{n \in N_j} a(j, n, k, t), \forall k \in \psi, j \in V, j \neq o_k, j \neq s_k, t \in \Phi \tag{5}$$

$$x(i, j, k) = a(i, j, k, T), \quad \forall k \in \psi, (i, j) \in E \tag{6}$$

$$y(i, j, k, t) = a(i, j, k, t) - d(i, j, k, t), \forall k \in \psi, (i, j) \in E, t \in \Phi \tag{7}$$

$$\sum_{k} y(i, j, k, t) + \sum_{k} y(j, i, k, t) \leq 1, \quad \forall (i, j) \in E, t \in \Phi \tag{8}$$

Part 3: time constraints

$$TT(i, j, k) = \sum_{t} t \times ((d(i, j, k, t) - d(i, j, k, t-1)) - (a(i, j, k, t) - a(i, j, k, t-1))), \quad \forall k \in \psi, (i, j) \in E \tag{9}$$

$$TT(i, j, k) \geq T_{\min}(i, j) \times x(i, j, k), \quad \forall k \in \psi, (i, j) \in E \tag{10}$$

$$\sum_{k} d(i, j, k, t) - \sum_{k} d(i, j, k, t-1) \leq 1, \quad \forall (i, j) \in E, t \in \Phi \tag{11}$$

$$d(i, j, k, t) \geq d(i, j, k, t-1), \quad \forall k \in \psi, (i, j) \in E \tag{12}$$

$$a(i, j, k, t) \geq a(i, j, k, t-1), \quad \forall k \in \psi, (i, j) \in E \tag{13}$$

Eqs. (1)–(3) correspond to the space constraints on the start node, the intermediate node and the destination node of each AGV in the

Table 1
Defined parameters and decision variables in the FTSN model.

Parameter	Description
V	Set of all nodes, $\{1,2,\dots,N\}$
Φ	Set of time indices, $\{1,2,\dots,T\}$
ψ	Set of vehicle indices, $\{1,2,\dots,K\}$
i,j,n	Node index $i,j,n \in V$
k	Vehicle index, $k \in \psi$
t	Time index, $t \in \Phi$
E	Set of all arcs (i,j)
O	Set of start nodes of AGVs, $\{o_1,\dots,o_k,\dots,o_N\}$
S	Set of destination nodes of AGVs, $\{s_1,\dots,s_k,\dots,s_N\}$
N_i	Set of neighbor nodes of node i
$E^s(i)$	Set of arcs ending at node i
$T_{\text{start},k}$	Start time for vehicle k
$T_{\min}(i,j)$	Minimal time between nodes i and j
Variables	**Description**
$x(i,j,k)$	Binary, $x(i,j,k)=1$ if arc (i,j) is selected by vehicle k, otherwise $x(i,j,k)=0$
$y(i,j,k,t)$	Binary, $y(i,j,k,t)=1$ if vehicle k occupies arc (i,j) at time t, otherwise $y(i,j,k,t)=0$
$a(i,j,k,t)$	Binary, $a(i,j,k,t)=1$ if vehicle k has arrived at arc (i,j) by time t, otherwise 0
$d(i,j,k,t)$	Binary, $d(i,j,k,t)=1$ if vehicle k has left arc (i,j) by time t, otherwise 0
$TT(i,j,k)$	Integer, transport time for arc (i,j) of vehicle k

roadmap, respectively. Eq. (1) ensures that only one adjacent node of the start node o_k for AGV k is selected. Eq. (2) guarantees the route continuity for AGV k. Eq. (3) makes sure that only one adjacent node for the end node s_k of AGV k is selected.

Eq. (4) is the time constraint of AGV k for its arrival time and its departure time at arc (i,j). $a(i,j,k,t)$ and $d(i,j,k,t)$ are cumulative flow variables to denote if AGV k has reached or left arc (i,j) by time t. Since AGV k leaves after its arrival for arc (i,j), Eq. (4) holds. Eq. (5) is arc-to-arc transition constraint, aiming to guarantee $d(i,j,k,t) = a(j,n,k,t)$ if the adjacent arcs (i,j) and (j,k) are both used for AGV k. Eq. (6) maps the cumulative flow variable $a(i,j,k,t)$ in the time–space network to the graph variable $x(i,j,k)$ in the physical network, if arc (i,j) is selected by AGV k to traverse the roadmap from its origin to destination. Eq. (7) describes the relationship of the variable $y(i,j,k,t)$ and the cumulative flow variables ($a(i,j,k,t)$ and $d(i,j,k,t)$). The variable $y(i,j,k,t)$ represents whether arc (i,j) is occupied by AGV k at time t, and this occupancy is expressed by the difference between $a(i,j,k,t)$ and $d(i,j,k,t)$. Equality (8) guarantees that each arc can be occupied by at most one AGV at any time t.

Eq. (9) constraints on transport time $TT(i,j,k)$ on arc (i,j) by AGV k, which is computed by the difference between the departure and arrival times of AGV k when occupying arc (i,j). For the sake of compactness, these arrival and departure times (defined as $A(i,j,k)$ and $D(i,j,k)$) can be calculated by $a(i,j,k,t)$ and $d(i,j,k,t)$, as follows:

$$A(i,j,k) = \sum_t \left(t \times \left(a(i,j,k,t) - a(i,j,k,t-1)\right)\right), \forall k \in \psi, (i,j) \in E \quad (14)$$

$$D(i,j,k) = \sum_t \left(t \times \left(d(i,j,k,t) - d(i,j,k,t-1)\right)\right), \forall k \in \psi, (i,j) \in E \quad (15)$$

Eq. (10) gives the minimum transport time on arc (i,j) by AGV k. Constraint (11) represents that for each arc at most one AGV can leave at any time. Constraints (12) and (13) provide time connectivity by using the cumulative flow variables ($a(i,j,k,t)$ and $d(i,j,k,t)$), indicating that $a(i,j,k,t)$ or $d(i,j,k,t)$ has to be 1 for all later moments if AGV k has reached or departed from arc (i,j) at time t.

Here, we use Fig. 1 to better understand the occupancy of arc (i,j) by AGV k via the cumulative flow variable ($a(i,j,k,t)$, $d(i,j,k,t)$) in the considered flexible time-space network.

Fig. 1 shows that $a(i,j,k,t)=1$ for $t \geq 6$ and $d(i,j,k,t)=1$ for $t \geq 8$. This indicates that AGV k reaches and leaves arc (i,j) at $t=6$ and $t=8$, due to the definitions of $a(i,j,k,t)$ and $d(i,j,k,t)$ in Table 1. Meanwhile, it is observed that the arc (i,j) is occupied by AGV k at $t=6$ and $t=7$, using Eq. (7). It is noted that $a(i,j,k,t) - a(i,j,k,t-1) = 1$ only when $t=6$ for the arrival, while $d(i,j,k,t) - d(i,j,k,t-1) = 1$ only when $t=8$ for the departure. In this case, $A(i,j,k)=6$ and $D(i,j,k)=8$, as computed by Eqs. (14)–(15).

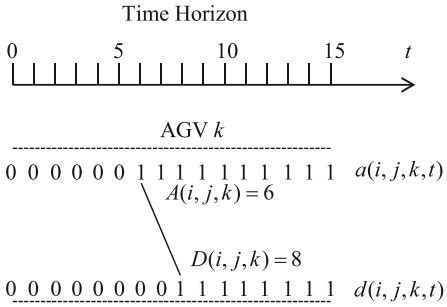


Fig. 1. Illustrative example of occupancy of arc (i,j) by AGV k via the cumulative flow variables.

2.3. The planning architecture

This part proposes a hierarchical architecture for energy-efficient path planning of AGVs based on the FTSN model, illustrated in Fig. 2. Hierarchical planning decomposes a complex decision problem into smaller subproblems and solve the overall problem effectively, the planning procedures are given as follows:

1. The time controller of AGV k computes the minimal transport time $T_{\min}(i,j,k)$ on the arcs of the roadmap;
2. The supervisory controller receives these minimal transport times from each AGV time controller;
3. The supervisory controller determines the optimal transport time $TT(i,j,k)$ and the schedule information $A(i,j,k)$ and $D(i,j,k)$. The obtained schedule are then sent to the time controller of each AGV;
4. The time controller of each AGV sends the transport times on the visited arcs to its speed controller. The speed controller of AGV k designs a control algorithm to track speed $v(k,t)$ regarding the schedule information.

In the following sections, we focus on solving the optimization problem decided by the supervisory controller.

2.4. Energy-efficient planner

This section proposes an energy-efficient planner based on the FTSN model presented in Section 2.2. We introduce the two types of objective functions and give an overall formulation of the energy-efficient optimization problem when introducing energy-related objectives and constraints.

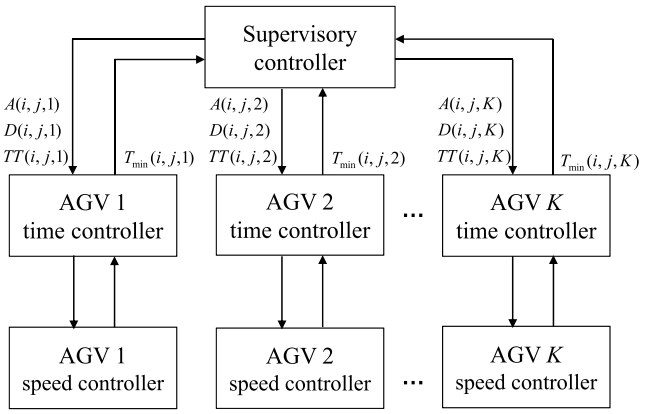


Fig. 2. The hierarchical control architecture for the multiple AGVs.

2.4.1. Time-related objective

For a logistics system (e.g. warehouse or manufacturing), the primary objective is to minimize the completion time of all the tasks. As suggested in Murakami (2020), the objective defined as J_1 is to minimize the sum of the completion time of all given tasks. Compared to the makespan (the maximum of all completion times), this objective is more strict. For AGV k, the task completion time is expressed as $\sum_{(i,s_k)\in E^s(s_k)} D(i,s_k,k)$, so J_1 is represented as follows:

$$J_1 = \sum_k \sum_{(i,s_k)\in E^s(s_k)} D(i,s_k,k). \tag{16}$$

If J_1 is considered, the corresponding optimization problem, which is defined as P_1, is given as follows:

$$(P_1) \quad \min J_1$$
$$\text{s.t. } (1)-(13).$$

This optimization is Mixed Integer Programming (MIP), which can be handled by commercial solvers, like Gurobi (Edward, 2014). The minimum value of J_1 solved by the problem P_1 is denoted as $J_{1,\min}$.

2.4.2. Energy-related objective

The energy consumption, defined as J_2, consists of the following parts (Young et al., 2013):

$$J_2 = J_a + J_r, \tag{17}$$

where J_a is the part resulting from the vehicle acceleration, while J_r is the part generated by the rolling resistance. As the vehicle speed is relatively low, the energy consumption coming from air drag is neglected.

The energy consumption resulting from vehicle acceleration J_a is given as follows:

$$J_a = 0.5M \sum_k \sum_t z(k,t) \times [v^2(k,t) - v^2(k,t-1)], \tag{18}$$

where M is the AGV weight, $v(k,t)$ is the speed from $t-1$ to t and $v(k,t) \in [0, v_{\max}]$ ($v_{\max}$ is the maximal speed). We define the variable $z(k,t)$ as follows:

$$z(k,t) \triangleq \begin{cases} 1 & v(k,t) \geq v(k,t-1) \\ 0 & v(k,t) < v(k,t-1) \end{cases} \tag{19}$$

Regarding J_r, the traction force $f(i,j,k)$ of AGV k on arc (i,j) is represented as follows:

$$f(i,j,k) = MgC_r, \tag{20}$$

where C_r is the coefficient of rolling resistance. Since energy is the product of force and distance, J_r can be described as follows:

$$J_r = \sum_k \sum_{(i,j)} MgC_r \cdot S(i,j) \cdot x(i,j,k), \tag{21}$$

where $S(i,j)$ is the distance for arc (i,j).

To obtain energy-efficient paths for AGVs, we give the direct optimization problem (defined as P_2) in a lexicographic-form (Pérez-Cañedo and Concepción-Morales, 2020), which is represented as follows:

$$(P_2) \quad \min J_2$$
$$\text{s.t. } J_1 \leq J_{1,\min},$$
$$(1)-(13).$$

However, the problem P_2 cannot be solved as a standard optimization formulation because J_2 depends on the condition $z(k,t) = 1$. We next rewrite the problem P_2 as a describable standard optimization problem to be solved.

Theorem 1. *The problem P_2 can be formulated as a describable mixed-integer programming problem to be solved.*

Proof. In J_a, there are two variables, $v(k,t)$ and $z(k,t)$. $v(k,t)$ can be represented by (22) directly. But not every arc (i,j) is visited for the AGV k, the decision variable $TT(i,j,k)$ may be zero that Eq. (22) is not feasible.

$$v(k,t) = \sum_{(i,j)} y(i,j,k,t) \cdot S(i,j)/TT(i,j,k) \tag{22}$$

It is noted that, due to the condition constraint in (19) of $z(k,t)$, we cannot optimize J_2 in P_2 directly. We introduce new variables and constraints to deal with the two kinds of circumstances.

For $v(k,t)$, we introduce a discretized integer number μ ($\mu \in \{1,2,\ldots,h\}$, h is set as the maximum number of time slots the AGV moves on the arc) for $TT(i,j,k)$ as follows:

$$TT(i,j,k) = \mu \times \sum_{\mu} L(i,j,k,\mu), \ \forall k \in \psi, (i,j) \in E, \tag{23}$$

where $L(i,j,k,\mu)$ is a binary decision variable to select an element μ. With this, $v(k,t)$ can be further described as follows:

$$\begin{aligned} v(k,t) &= \sum_{(i,j)} y(i,j,k,t) \frac{S(i,j)}{\mu \times \sum_{\mu} L(i,j,k,\mu)} \\ &= \sum_{(i,j)} y(i,j,k,t) \sum_{\mu} \frac{L(i,j,k,\mu) S(i,j)}{\mu}, \forall k \in \psi, t \in \Phi. \end{aligned} \tag{24}$$

For formula (19), $-v_{\max} \leq v(k,t) - v(k,t-1) \leq v_{\max}$. Based on the method in Bemporad and Morari (1999), formula (19) is equivalent to the following inequalities:

$$v(k,t-1) - v(k,t) \leq v_{\max}(1 - z(k,t)), \tag{25}$$

$$v(k,t-1) - v(k,t) \geq \epsilon - z(k,t)(v_{\max} + \epsilon), \tag{26}$$

where ϵ is a small tolerance (typically the machine precision).

By substituting $v(k,t)$ via Eq. (24) into (18), and we can have a nonlinear objective function subject to linear constraints.

Ultimately, we reformulate the problem P_2 as an MINLP as follows:

$$(P_3) \quad \min J_2$$
$$\text{s.t. } J_1 \leq J_{1,\min},$$
$$(1)-(13), (23)-(26). \quad \square$$

For our path planning problem, the first goal is the shortest completion time of each task, which has an important impact on the production efficiency of the system. On this basis, the energy consumption target is further considered. The energy consumption of the AGVs is composed of the energy consumption generated by rolling resistance and the energy consumption generated by vehicle acceleration. Among them, the energy consumption generated by rolling resistance is determined by variable $x(i,j,k)$, which is related to the path length of AGVs; the energy consumption generated by vehicle acceleration is determined by the variables $z(k,t)$ and $v(k,t)$, which are related to the vehicle speed change during the AGV driving.

Property 1. *The optimization problem P_3 is a non-convex MINLP.*

Proof. For the nonlinear objective function, the Hessian matrix of $z(k,t)y(i,j,k,t)^2L(i,j,k,\mu)^2$ is not positive definite, and the problem P_3 is a non-convex MINLP. □

The problem P_3 is proved as a non-convex MINLP, and non-convex MINLPs are regarded as *difficult* optimization problems and usually turn out to be NP-hard (Burer and Letchford, 2012). The resulting problem is not computationally efficient when solved by the existing commercial MINLP solver Baron, as shown in the results section. The following section proposes an efficient meta-heuristic algorithm for providing high-quality solutions in a reasonable computation time.

3. Solution approach

This section proposes a new algorithm to efficiently solve the formulated non-convex MINLP problem (P_3). The proposed algorithm uses a hybrid metaheuristic, integrating the genetic algorithm (GA) and the estimation of distribution algorithm (EDA). For solving the MINLP problems, GA has shown its capability (Young et al., 2007; Tang et al., 2021). However, our MINLP includes two aspects of decision variables: path and motion time. When the motion time of the AGV is incorporated into the decision variables, the decision dimension becomes significantly higher than the optimization problem without optimizing the motion time. GA may thus have slow convergence rate, due to insufficient local search ability and easy falling into local optimal. To address this drawback, GA is integrated with other metaheuristics to improve its search ability (Lee et al., 2019).

In the proposed hybrid metaheuristics, we integrate GA with EDA to determine the paths and motion times of the AGVs, respectively. The EDA is capable to deal with high-dimension complex decision problems (Hauschild and Pelikan, 2011). However, the paths need to satisfy the continuity constraints and the path probability model of EDA cannot be directly established. Considering the efficiency of GA to obtain a feasible path (Farooq et al., 2021), we use GA to search for the paths of AGVs. Regarding the motion times, the motion time is decided within a continuous range, and the probability model of EDA can be easily obtained.

3.1. Encoding

When designing the metaheuristics, the encoding scheme is crucial for the solution quality (Tang et al., 2021: ?). For solving the considered MINLP problem, a key step is to design a suitable encoding scheme, which fits the FTSN framework.

To match the FTSN framework, a two-dimension encoding scheme is designed for each individual X. Each chromosome contains two types of information in two dimensions individually. Dimension 1 includes the path node sequence $X^{\mathrm{r},k}$ of the AGVs by following constraints (1)–(3), while Dimension 2 gives the transport time X^{t} of the AGVs, as illustrated in Fig. 3(a). $X^{\mathrm{r},k} = \left[X_1^{\mathrm{r},k}, \ldots, X_l^{\mathrm{r},k}, \ldots, X_N^{\mathrm{r},k}\right]$, while $X^{\mathrm{t},k} = \left[X_1^{\mathrm{t},k}, \ldots, X_l^{\mathrm{t},k}, \ldots, X_N^{\mathrm{t},k}\right]$, $l \in \{1, 2, \ldots, N\}$. $X_l^{\mathrm{r},k}$ represents the node located at the lth position of $X^{\mathrm{r},k}$, and $X_l^{\mathrm{t},k}$ denotes the transport time located at the lth position of $X^{\mathrm{t},k}$. The encoding length used here is $N \times K$.

For Dimension 1, origin node O_k for AGV k corresponds to $x_1^{\mathrm{r},k}$, while its destination node S_k corresponds to one of the path nodes in $X_l^{\mathrm{r},k}$. For Dimension 2, the value of $X_1^{\mathrm{t},k}$ is zero, because $x_1^{v,k}$ is the origin node. $X_l^{\mathrm{r},k}$ $(1 < l \leq N)$ corresponds to the transport time $TT(X_{l-1}^{\mathrm{r},k}, X_l^{\mathrm{r},k}, k)$ spent on arc $(X_{l-1}^{\mathrm{r},k}, X_l^{\mathrm{r},k})$ by AGV k. Note that the location of S_k is the node between $X_l^{\mathrm{r},k}$ and $X_T^{\mathrm{r},k}$. Under this condition, the nodes after S_k become meaningless and the related transport time becomes zero.

We use the simple example shown in Fig. 3(b) to illustrate the designed encoding scheme ($K = 2$, $N = 5$). The path sequences are as follows: AGV 1: $1 \xrightarrow{2} 3$, AGV 2: $4 \xrightarrow{1} 3 \xrightarrow{1} 5$. The nodes 1 and 4 are the origin and destination nodes of AGV 1, while the nodes 4 and 5 are the origin and destination nodes of AGV 1. For the transport time, $TT(1,3,1) = 2$ for AGV 1, while $TT(4,3,2) = 1$ and $TT(3,5,2) = 1$ for AGV 2. The path sequences in Dimension 1 will be updated by GA while the transport times in Dimension 2 will be improved by EDA, using our proposed hybrid metaheuristic.

3.2. Algorithm

In the proposed hybrid metaheuristic, we integrate GA and EDA to determine the paths and the motion times of the considered AGVs.

Fig. 4 gives the flowchart of the proposed hybrid metaheuristics. Based on GA and EDA, our algorithm is a population-based metaheuristic. The entire population (defined as P) consists of three sub-populations P_{t}, P_{e} and P_{l}, to maintain the diversity of updating transport time $TT(i,j,k)$. P_{e} represents the sub-population with the feature of the shortest transport time, P_{e} represents the sub-population with the feature of the energy-saving transport time, and P_{l} represents the sub-population with the learning feature of the transport time. The ratio of these three sub-populations is set to 1:1:2, as suggested in Wu and Wang (2018).

In each sub-population, the elite strategy is considered to select a certain ratio of best solutions after evaluating the population fitness. For generating new solution candidates, Dimension 1 of solution X is updated by the mutation strategy using the GA, while Dimension 2 of the solution X is updated by the population-based incremental learning (PBIL) to construct the probability distribution model. After generating the new solutions, the infeasible solutions will be repaired. The algorithm continues until the maximum iteration is reached.

As for the GA used in the hybrid metaheuristic, the mutation strategy is used to customize the GA and efficiently construct path sequences. The crossover strategy is not suggested because it is observed that the crossover could result in longer computation times and deteriorate the solution quality when solving the routing problem (Halim and Ismail, 2019). For the EDA, PBIL is employed to update the probability matrix for generating the transport times, because the transport times of two successive arcs are independent and PBIL has an effective learning strategy (Meng et al., 2016).

3.2.1. Solution generation

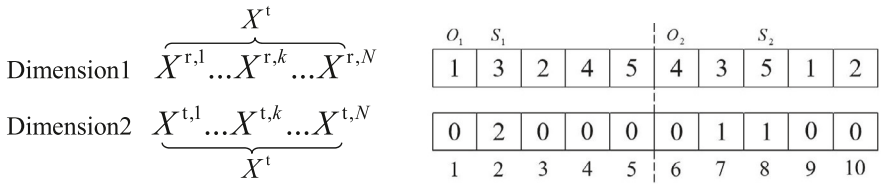
The generated solutions are based on the selected elite solutions. Regarding the selection, the top-ranking method is employed.

Regarding Dimension 1 of the new solutions, three mutation operators (flip, swap, and slide) of the GA are used to update the path of AGVs. The flip mutation works by randomly choosing two positions in the chromosome and reversing the order in which the values appear between those positions. The swap operator randomly swaps the values of two positions in the solution chromosome. For the slide operator, two positions in the chromosome are randomly selected, and the contents between these two positions move one position to the left. The details of these operations are given in Halim and Ismail (2019).

For Dimension 2, the PBIL algorithm is adapted to construct the probability distribution model. The time probability matrix, defined as $C^k(iter)$, is to compute the transport time $x_l^{t,k}$ for AGV k at iteration $iter$. The composition of $C^k(iter)$ is given as follows:

$$C^k(iter) = \begin{bmatrix} \rho_{11}^k(iter) & \cdots & \rho_{1h}^k(iter) \\ \vdots & & \vdots \\ \rho_{l1}^k(iter) & \cdots & \rho_{lh}^k(iter) \\ \vdots & \ddots & \vdots \\ \rho_{N1}^k(iter) & \cdots & \rho_{Nh}^k(iter) \end{bmatrix} \tag{27}$$

where $\rho_{l\mu}^k(iter)$ represents the probability $p_{iter}(x_l^{t,k} = \mu)$.



(a) Composition of X in the designed scheme. (b) Example of the designed scheme.

Fig. 3. Illustration of the designed encoding that fits the FTSN framework.

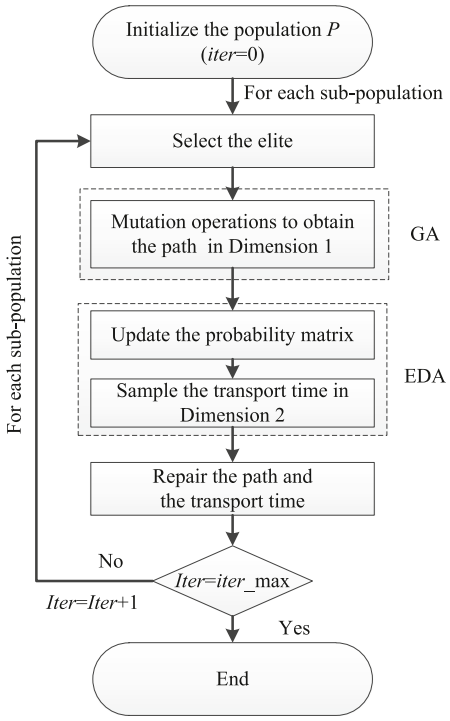


Fig. 4. Flowchart of the proposed hybrid metaheuristic.

In the PBIL algorithm, $C^k(iter)$ is adjusted to make the populations evolve towards the excellent individuals, element $\rho_{l\mu}(iter+1)$ of this matrix is updated according to the elite populations, as follows:

$$\rho^k_{l\mu}(iter+1) = (1-\alpha)\rho^k_{l\mu}(iter) + \alpha\frac{1}{E}\sum_{s=1}^{E} I^{k,s}_{l\mu}(iter) \tag{28}$$

$$I^{k,s}_{l\mu}(iter) = \begin{cases} 1, & \text{if } X^{t,k}_l = \mu \text{ in the } s\text{th elite solution} \\ 0, & \text{else,} \end{cases} \tag{29}$$

where $\alpha \in (0,1)$ is the learning rate and $I^{k,s}_{l\mu}(iter)$ is the indication function corresponding to the sth solution of the elite population by AGV k at iteration $iter$. E is the number of the elite populations. After updating the probability matrix, the second chromosome needs to be sampled by them respectively, such that the population $P(iter+1)$ are generated for the next iteration.

Note that $C^k(iter)$ is updated in three sub-populations P_t, P_e and P_l, corresponding to three notations defined as $C^k_t(iter)$, $C^k_e(iter)$ and $C^k_l(iter)$. The updates of $C^k_t(iter)$, $C^k_e(iter)$ and $C^k_l(iter)$ are the same as presented in Eqs. (27), (28), and (29). The initializations of $C^k_t(iter)$, $C^k_e(iter)$, and $C^k_l(iter)$ are provided in Appendix.

3.2.2. Fitness function

$F(X)$ is the fitness function of solution X that consists of two parts: the objective function $J(X) = J_2$ and the penalty function $p(X)$. The aim is to obtain the solution with the lowest energy consumption for the shortest total completion time.

The penalty function $p(X)$ avoids searching in infeasible parts of the solution space at iteration $iter$, as shown in Eqs. (30)–(32).

$$p(X) = p_1(X) + p_2(X) \tag{30}$$

$$p_1(X) = (T_{end}(X) - T^1_{end}) \times R_1 \tag{31}$$

$$p_2(X) = \begin{cases} R_2 & \text{Eq. (8) does not hold for } X \\ 0 & \text{else,} \end{cases} \tag{32}$$

where T^1_{end} is the shortest completion time computed by using the conventional TSN model, and R_1 and R_2 are relatively large constants. $p_1(x)$ guarantees that the solutions with the shortest completion time have a better fitness, and $p_2(x)$ ensures that the fitness values of infeasible solutions are very poor.

3.2.3. Repairing operations

To ensure the feasibility of solution X, individuals in population P may need to be repaired. In principle, *Dijkstra* algorithm is used to repair the unconnected paths in Dimension 1. The specific repairing operations are as follows:

$$X^{r,k}(l_1 : l_1+1) = D[x^{r,k}_{l_1}, x^{r,k}_{l_1+1}], \quad if\ x^{r,k}_{l_1} \notin E^s(x^{r,k}_{l_1+1}) \tag{33}$$

$$X^{r,k}(l_2 : l_3) = [X^{r,k}_{l_2}], \quad if\ X^{r,k}_{l_2} = X^{r,k}_{l_3} \tag{34}$$

where, l_1, l_2 and l_3 are the gene locations; $X^{r,k}(l_1 : l_1+1)$ represents the sequence fragment from the l_1st position to the l_1+1st position of the path sequence; $D[x^{r,k}_{l_1}, x^{r,k}_{l_1+1}]$ represents a path sequence with $x^{r,k}_{l_1}$ as the start point and $x^{r,k}_{l_1+1}$ as the end point obtained by using the *Dijkstra* algorithm.

For Eq. (33), if the path sequence fragment from the l_1st position to the l_1+1st position is unconnected, *Dijkstra* algorithm is used to generate a connected path to replace it. The repairing operation may cause the path sequence to contain repeating paths, the further operation of the repeating sequence segment as shown in Eq. (34). If the value of the path sequence at the l_2st position coincides with the l_3st position, the unnecessary intermediate path sequence will be deleted.

Then, if X^t does not satisfy the time constraints of the FTSN model, $TT(i,j,k)$ in Dimension 2 will be adjusted. The specific repairing operation is as follows:

$$x^{t,k_1}_{l_1} = x^{t,k_1}_{l_1} + 1, \quad if\ x^{t,k_1}_{l_1} = x^{t,k_2}_{l_2}, \sum_{l=1}^{l_1} x^{t,k_1}_l = \sum_{l=1}^{l_2} x^{t,k_2}_l \tag{35}$$

where, l_1 and l_2 are the gene locations. Eq. (35) means that if any two AGVs are in the same position at the same time, one AGV will be selected, and its encoding time value at that position will add 1 to avoid collisions between AGVs.

Algorithm 1 The designed HGE based on the FTSN model.

Input: Task assignments for all the AGVs
Output: The best individual X_{best}

$iter = 0$, $F_{\min} = Inf$
initialize $P(iter)$
while $iter \leq iter_{max}$ **do**
  for each sub-population P_{t}, P_{e} and P_{l} **do**
    evaluate fitness $F(X) = J_a + J_r + p(X)$
    if $F(X) \leq F_{\min}$ **then**
      record X as the best individual X_{best}
      record $F_{\min} = F(X)$
    end if
    select the elite of the sub-population
    flip, swap or slide X^{r} in the elite to construct $P^{\text{r}}(iter+1)$
    update the probability matrix $C^k(iter)$ ($C^k_{\text{t}}(iter)$ for P_{t}, $C^k_{\text{e}}(iter)$ for P_{e}, and $C^k_{\text{l}}(iter)$ for P_{l})
    for $X^{\text{r}} \in P^{\text{r}}(iter+1)$ **do**
      sample X^{t} according to $C^k_{\text{t}}(iter)$, $C^k_{\text{e}}(iter)$ and $C^k_{\text{l}}(iter)$ respectively, to construct $P^{\text{t}}(iter+1)$
    end for
  end for
  for $s = 1$ to N_{p} **do**
    if $X_s^{\text{r},k}$ does not correspond to a connected path **then**
      repair $X_s^{\text{r},k}$ by Equations (33) and (34)
    end if
    if X_s does not satisfy time constraints **then**
      adjust $X_s^{\text{t},k}$ by Equation (35)
    end if
  end for
  $iter = iter + 1$
end while

3.2.4. Pseudocode of the algorithm

This part gives the pseudocode of the developed hybrid metaheuristics for the considered MINLP. This pseudocode is described in Algorithm 1, detailing the flowchart of Fig. 4.

As discussed earlier, the entire population P is divided into three sub-populations P_{t}, P_{e}, and P_{l}. For each iteration indexed by $iter$, a certain ratio (marked by $\eta\%$) for each sub-population is selected as the elite, marked as $P'_{\text{t}}(iter)$, $P'_{\text{e}}(iter)$, and $P'_{\text{l}}(iter)$. Then, three mutation operations are performed for the path sequence (X^{r}) in the elite, to generate new solutions of Dimension 1 and construct $P^{\text{r}}(iter+1)$, which contains Dimension 1 of all solutions in the next iteration. Afterwards, for each newly generated solution, update the probability matrix $C^k(iter)$ and sample the transport time represented in X^{t} to construct $P^{\text{r}}(iter+1)$, which corresponds to Dimension 2 for every solution in population P. When the new population $P(iter+1)$ is generated by GA and EDA, perform the repair operations (detailed in Section 3.2.3) to adjust the unfeasible solutions.

4. Case studies

This section presents the numerical experiment results of the developed hybrid algorithm GA-EDA (HGE) to solve the MINLP by using the FTSN model. The first part introduces the benchmark of a material transport system and experiment settings. The second part discusses how the key parameters of HGE are selected. Then, the proposed HGE algorithm (using the FTSN representation), is compared to the existing methods for addressing the same path planning problem. We then use an example to show how the energy efficiency is improved by analyzing the determined paths.

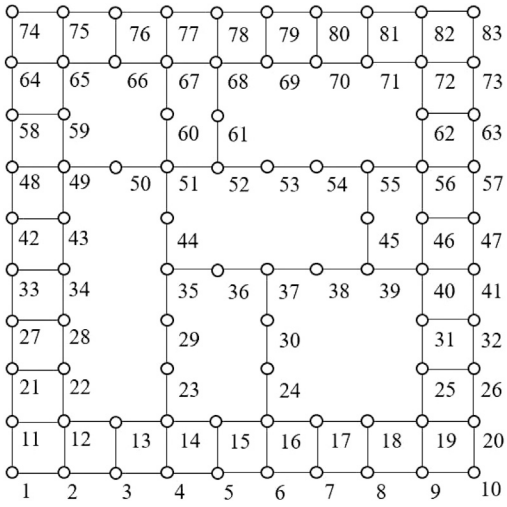


Fig. 5. Example of a roadmap in the industrial application (Scenario I).

4.1. Setting of the AGV transport system

4.1.1. Benchmark

To evaluate the performance of the proposed FTSN model, we consider a material handling benchmark system used in the manufacturing or warehouse environment. The roadmap is $n \times n$ squared graph with different connection conditions. An example of the roadmap is shown in Fig. 5, adapted from the layout in Nishi et al. (2020).

This benchmark takes into account the physical features of AGVs, including the dynamical behavior parameters (such as vehicle speed and weight). The key parameters of the AGV system are listed as follows:

- The total weight of AGV: $M = 320$ kg.
- Gravitational constant: $g = 9.81$ m/s^2.
- Maximal speed: $v_{max} = 1$ m/s.
- Rolling resistance coefficient: $C_r = 0.01$.

4.1.2. Experiment setting

This part sets up case studies to evaluate the proposed methodology in different layout topologies similar to Fig. 5 with different numbers of AGVs. For the conducted experiments, two types of scenarios (small-scale and large-scale) are considered. The small-scale scenarios use a small $n \times n$ ($n = 3, 4, 5$) roadmap to show that the proposed HGE can find the optimal solution. The large-scale scenarios consider a large $n \times n$ ($n = 10$) roadmap with 5 different connection conditions to show the advantage of the proposed HGE.

For each case, 20 experiments are tested for comparison in general. In each experiment, the pickup point and the delivery point of a particular task are given randomly in advance. The maximum computation time is set to be 1 h. The settings of these case studies are given as follows:

- The pickup point is considered as the initial position of each AGV.
- The material weight to be transported for each task is assumed to be the same.
- All the materials are assumed to be ready at each pickup point.
- The service times of each AGV at the pickup node and the delivery node are ignored.
- The distance $S(i, j)$ is set to be 10 m for an indoor inventory system, as suggested by Adamo et al. (2018). The unit time ΔT is considered to be 10 s.

Table 2
Factor levels of four key parameters of the proposed GEDA.

Parameter	1	2	3	4
P_size	50	100	150	200
$\eta(\%)$	10	20	30	40
α	0.1	0.2	0.3	0.4
β	0.5	0.6	0.7	0.8

Three key performance indicators (KPIs) are considered to evaluate the benchmark system: (1) *Completion time*: the sum of completion times of the transport tasks; (2) *Energy consumption*: the sum of the kinetic energy of the AGVs for completing these tasks; (3) *Computation time*: the time to compute the solution for each planning problem.

The results of the developed FTSN-HGE method are compared with the TSN model, the MINLP solver Baron, and three standard commonly-used metaheuristics (GA, Particle Swarm Optimization (PSO), and PBIL). The comparison methods are briefly described as follows:

- The TSN model is a state-of-the-art representation for the conflict-free path planning of multiple AGVs (Nishi et al., 2005; Murakami, 2020). In the TSN model, the transport time between every two successive nodes in the roadmap is fixed. Therefore, using the TSN model, a single-objective optimization, which minimizes the completion time only, is considered. The resulting optimization problem is an MIP solved by Gurobi.
- Baron is regarded as an efficient commercial solver for solving MINLPs. Baron implements a branch and bound algorithm that utilizes linear programming for the bounding step to solve MINLP problems.
- GA is an efficient metaheuristic for solving MINLPs (Tang et al., 2021). In our MINLP, the standardized GA is used to generate the path sequence and transport time jointly (Farooq et al., 2021).
- PSO is a typical swarm optimization algorithm for solving the routing problem (Wang et al., 2003). The PSO is used to determine both the path sequence and transport time.
- PBIL is an efficient EDA for solving the MINLPs (Meng et al., 2016; Wu and Wang, 2018). The standard PBIL is used to update the path sequence and transport time jointly. Both GA, PSO, and EDA are designed based on the FTSN framework.

The hardware for all these experiments is an Intel i7-9700 processor (3.0 GHz) with 8 GB of memory. The optimization problems are modeled and solved in Python. The fitness evaluation times of the four metaheuristics (HGE, GA, PSO, and PBIL) are set to 10,000.

4.2. Parameters selection

There are four key parameters in the designed HGE: P_size, η, α and β. We determine these parameters by the Taguchi method for the design-of-experiment (DOE) (Farooq et al., 2021). A large-scale instance ($K = 8$) is considered, and the layout is shown in Fig. 5. For each parameter, four-factor levels are selected as shown in Table 2.

Based on these settings, the orthogonal array $L_{16}(4^4)$ is selected, and 30 experiments are carried out for each combination of the parameters in $L_{16}(4^4)$ In these experiments, the number of evaluations is set as 10,000, $R_1 = 1000, R_2 = 10000$. The average proportion of energy consumption reduction is taken as the response variable (RV) value for each combination of these parameters. Table 3 lists the orthogonal array and RV values.

The influence trend curves of parameters are shown in Fig. 6. The number of populations P_size has the most significant effect on performance, as P_size affects the search depth. For the elite population ratio, a small η may get insufficient information, while a large η may lead to wrong information in the probability matrix. The learning rate α affects the algorithm convergence, and the mutation operator β can help to improve the population diversity. Here, we select $P\_size = 150$, $\eta = 20$, $\alpha = 0.2$, $\beta = 0.6$ for the following experiments.

Table 3
Orthogonal array and RV values.

Number	P_size	η	α	β	RV
1	1	1	1	1	0.104
2	1	2	2	2	0.118
3	1	3	3	3	0.112
4	1	4	4	4	0.096
5	2	1	2	3	0.126
6	2	2	1	4	0.129
7	2	3	4	1	0.129
8	2	4	3	2	0.125
9	3	1	3	4	0.129
10	3	2	4	3	0.125
11	3	3	1	2	0.129
12	3	4	2	1	0.129
13	4	1	4	2	0.125
14	4	2	3	1	0.125
15	4	3	2	4	0.125
16	4	4	1	3	0.121

4.3. Results and discussion

4.3.1. Results on small-scale cases

In this subsection, the experimental results on the small-scale scenarios with 2 AGVs are presented by using the six different methods. The compared results with respect to the given KPIs (completion time, energy consumption, and computation time) are reported in Table 4. The proposed HGE algorithm is compared to the TSN method, the MINLP solver Baron, GA, PSO, and PBIL. The proposed HGE, Baron, GA, PSO, and PBIL are based on the FTSN model. Three small-scale scenarios are tested by using the $n \times n$ roadmap layout ($n = 3, 4, 5$). We note that for these small-scale scenarios, the MINLP solver Baron returns the optimal solution for all these scenarios, and the optimality of each solution is thus guaranteed by the Gurobi solver.

Table 4 shows that the FTSN model achieves the same completion time as the one computed by the TSN method, which minimizes the completion time only. This means that the proposed HGE algorithm obtains the minimum completion time. Regarding the energy consumption, the FTSN model uses less energy consumption than the TSN method for the scenarios ($n = 4, 5$); the proposed HGE algorithm computes the same value when compared to the other four methods (Baron, GA, PSO, and PBIL). For the small-scale scenarios, we conclude that the proposed HGE algorithm uses the minimum energy consumption for the minimum completion time.

It is shown in Table 4 that the computation time of these methods is quite different. Since the computational complexity of the MINLP is higher than the one of MIP provided by the TSN model, the computation time of the Baron solver is greatly larger than the one of the TSN method. The four metaheuristics (HGE, GA, PSO, and PBIL) compute their solutions within a considerable shorter computation time than the MINLP solver Baron.

4.3.2. Results on large-scale cases

This subsection compares the performance on large-scale scenarios regarding the six methods based on the TSN model and the proposed FTSN model. The computational results are given in Tables 5, 6, and 7. The five roman letters represent the five 10×10 squared roadmaps with different connection conditions.

Table 5 reports the average completion time for the large-scale scenarios. When compared to the other methods, the proposed HGE and PBIL have the shortest completion time for each scenario. The completion times determined by HGE and PBIL are slightly shorter than those determined by TSN. In certain scenario settings, FTSN may achieve a shorter completion time than TSN because FTSN uses flexible transport time rather than fixed transport time. Due to the lack of search capability when solving the non-convex MINLP, GA and PSO have slower completion times than TSN. Due to the computational

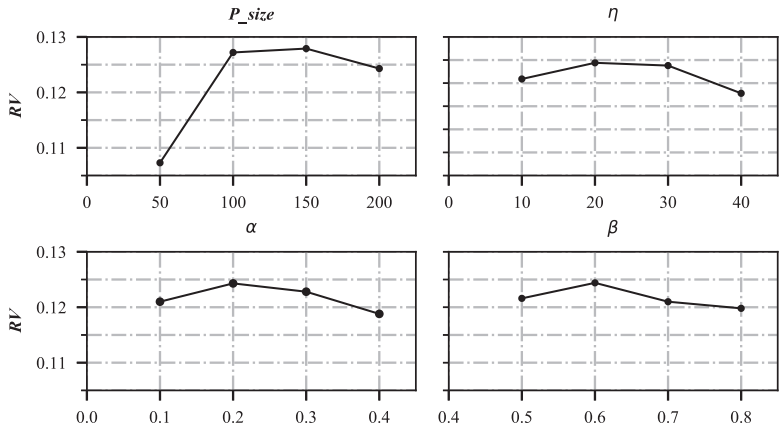


Fig. 6. Evaluation of key parameters of the proposed HGE on the RV value.

Table 4
Averaged results the small-scale scenarios ($K = 2$).

Completion time (Unit: seconds)						
Scale	TSN	FTSN				
		Baron	GA	PSO	PBIL	HGE
$n = 3$	35.00	35.00	35.00	35.00	35.00	35.00
$n = 4$	55.00	55.00	55.00	55.00	55.00	55.00
$n = 5$	58.00	58.00	58.00	58.00	58.00	58.00
Energy consumption (Unit: Joule)						
Scale	TSN	FTSN				
		Baron	GA	PSO	PBIL	HGE
$n = 3$	321.10	321.10	321.10	321.10	321.10	321.10
$n = 4$	361.86	341.73	341.73	341.73	341.73	341.73
$n = 5$	375.37	345.89	345.89	345.89	345.89	345.89
Computation time (Unit: seconds)						
Scale	TSN	FTSN				
		Baron	GA	PSO	PBIL	HGE
$n = 3$	1.18	34.05	6.83	5.13	1.65	2.61
$n = 4$	1.92	245.52	9.35	4.98	3.36	3.52
$n = 5$	2.73	484.93	16.21	15.23	8.75	4.96

Table 5
Average completion times for large-scale scenarios by using the five methods. (unit: seconds).

Settings		TSN	FTSN				
			Baron	GA	PSO	PBIL	HGE
I	$K = 8$	662.50	–	662.50	662.50	662.50	662.50
	$K = 9$	660.50	–	660.50	660.50	660.50	660.50
	$K = 10$	735.00	–	740.00	741.00	735.00	735.00
II	$K = 8$	629.50	–	629.50	629.50	629.00	629.00
	$K = 9$	660.00	–	661.50	661.50	660.00	660.00
	$K = 10$	746.00	–	747.00	746.50	746.00	746.00
III	$K = 8$	657.50	–	657.50	657.50	657.50	657.50
	$K = 9$	751.00	–	751.00	751.00	750.50	750.50
	$K = 10$	754.00	–	754.00	754.00	754.00	754.00
IV	$K = 8$	637.00	–	636.50	636.50	636.50	636.50
	$K = 9$	707.00	–	710.00	708.00	707.00	707.00
	$K = 10$	820.00	–	820.00	820.00	820.00	820.00
V	$K = 8$	583.00	–	583.00	583.00	583.00	583.00
	$K = 9$	682.50	–	682.50	682.50	682.50	682.50
	$K = 10$	774.00	–	774.00	774.00	774.00	774.00
	Average	697.30	–	697.97	697.87	**697.20**	**697.20**

complexity of the studied MINLP, the MINLP solver Baron cannot obtain a solution within the maximum computation time, as shown in Table 5 .

The energy consumption, which is our research focus, is shown in Table 6. The four metaheuristics (HGE, GA, PSO, and PBIL) reduce the energy consumption compared to the TSN method, which minimizes the completion time only. The proposed HGE obtains the lowest energy consumption for each scenario, when compared to GA, PSO, and PBIL. On average, the proposed HGE reduces the energy consumption by 11%, which is more than GA (6.23%), PSO (7.07%), and PBIL (8.00%). Analyzing the results in Table 6, the energy-saving effect for $K = 10$ may be worse than for $K = 8$ and $K = 9$. The reason could be that the running paths of AGVs are more complex when more AGVs are considered for the same roadmap.

Table 7 gives the computation times for the six methods in the large-scale scenarios. The TSN method solves the MIP, while the FTSN methods (Baron, GA, PSO, PBIL, and the proposed HGE) solve a more complicated MINLP, including nonlinear constrains and more decision variables than the MIP. This explains why the TSN uses much less computation time than the FTSN methods. As for the FTNS methods, the proposed HGE is the most computational-efficient metaheuristic, since HGE makes full use of GA to search the best path sequences, while PBIL is adopted to search the best transport time. GA, PSO, and PBIL take longer computational times than the proposed HGE, and this may be due to the required repair operations in case of a large number of infeasible solutions.

Fig. 7 gives the fitness curves for the three metaheuristics (GA, PSO, PBIL, and our HGE) based on the FTSN modeling framework for Scenario I ($K = 8$), to show the computation efficiency of the proposed HGE. Fig. 7 shows that the proposed HGE converges faster than GA, PSO, and PBIL, i.e., HGE explores the solution space more efficiently for the considered MINLP. Fig. 7 also shows that HGE achieves the best convergence accuracy, because the converged fitness of HGE is lower than GA, PSO, and PBIL.

From the above results of the small-scale and large-scale scenarios, the proposed HGE shows the positive potential to minimize the energy consumption for the minimal completion time. The studied MINLP problem, which is based on the FTSN model, is difficult to be solved for the large-scale scenarios in practice.

The proposed HGE shows its advantage over the existing metaheuristics (GA, PSO, and PBIL) to solve the MINLP. HGE achieves the same minimal completion time as the one obtained by the TSN method, both for the small-scale and large-scale scenarios. For the small-scale scenarios, HGE computes the same minimal energy consumption as the one computed by the MINLP solver Baron. For the large-scale scenarios, HGE is the most energy efficient (minimum completion time and minimal energy consumption), compared to GA, PSO, and PBIL.

4.4. Example of planned paths

This part demonstrates the planned energy-efficient paths determined when using the proposed HGE using the FTSN model. A simple

Table 6
Average energy consumption for large-scale scenarios by using the five methods. (unit: Joule).

Settings		TSN	FTSN							
			Baron	GA	Gap(%)	PSO	Gap(%)	PBIL	Gap(%)	HGE	Gap(%)
I	$K = 8$	1500.17	–	1385.96	7.61	1381.51	7.91	1367.28	8.86	**1311.29**	**9.98**
	$K = 9$	1638.04	–	1502.65	8.27	1486.72	9.24	1509.40	7.85	**1480.23**	**9.63**
	$K = 10$	1782.67	–	1692.16	5.08	1713.55	3.88	1676.46	5.98	**1610.67**	**9.99**
II	$K = 8$	1475.34	–	1370.68	7.09	1342.09	9.03	1330.45	9.82	**1311.31**	**11.12**
	$K = 9$	1652.20	–	1497.24	9.38	1481.30	10.34	1508.20	8.72	**1477.09**	**10.60**
	$K = 10$	1866.02	–	1775.12	4.82	1756.98	5.84	1732.96	6.27	**1674.02**	**10.29**
III	$K = 8$	1500.15	–	1395.10	7.00	1399.10	6.74	1366.15	8.93	**1330.15**	**11.33**
	$K = 9$	1663.12	–	1592.85	4.23	1575.11	5.29	1530.20	7.99	**1487.12**	**10.58**
	$K = 10$	1815.23	–	1705.23	6.06	1688.93	6.96	1722.05	7.61	**1635.45**	**9.92**
IV	$K = 8$	1539.35	–	1418.62	7.84	1396.25	9.30	1371.35	10.91	**1347.35**	**12.47**
	$K = 9$	1701.58	–	1587.12	6.73	1549.26	8.95	1504.69	11.57	**1479.58**	**13.05**
	$K = 10$	1833.18	–	1741.60	5.00	1763.27	3.81	1742.15	5.83	**1661.10**	**9.38**
V	$K = 8$	1481.80	–	1374.31	7.25	1348.48	9.00	1356.71	9.98	**1303.80**	**12.01**
	$K = 9$	1692.84	–	1624.28	4.05	1617.55	4.64	1557.96	7.97	**1466.81**	**13.35**
	$K = 10$	1831.76	–	1753.44	4.28	1738.69	5.08	1719.26	6.93	**1647.70**	**10.05**
	Average	1664.90	–	1561.09	6.23	1561.09	7.07	1553.02	8.00	**1481.58**	**11.01**

Table 7
Average computation time for large-scale scenarios by using the five methods. (unit: seconds).

Settings		TSN	FTSN				
			Baron	GA	PSO	PBIL	HGE
I	$K = 8$	3.07	3600	101.76	89.45	70.50	**36.96**
	$K = 9$	3.96	3600	117.33	120.42	75.83	**28.91**
	$K = 10$	4.36	3600	129.49	127.56	83.32	**28.91**
II	$K = 8$	4.10	3600	101.92	69.68	67.33	**40.52**
	$K = 9$	4.68	3600	117.39	112.14	78.89	**38.48**
	$K = 10$	5.58	3600	132.27	145.36	94.41	**46.17**
III	$K = 8$	3.80	3600	96.45	103.55	68.63	**35.73**
	$K = 9$	4.98	3600	118.44	121.66	87.76	**38.29**
	$K = 10$	4.14	3600	116.00	126.50	88.75	**36.75**
IV	$K = 8$	4.79	3600	92.37	73.28	65.95	**34.98**
	$K = 9$	4.98	3600	116.20	105.67	83.56	**37.77**
	$K = 10$	5.33	3600	125.46	119.13	93.94	**40.56**
V	$K = 8$	3.94	3600	95.79	89.32	63.55	**35.52**
	$K = 9$	4.38	3600	123.47	130.65	83.21	**38.78**
	$K = 10$	4.41	3600	128.22	132.71	90.06	**42.36**
	Average	4.43	3600	114.17	111.13	79.71	**39.38**

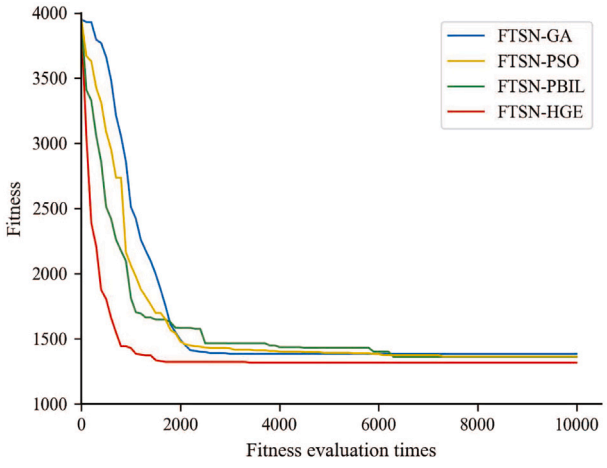


Fig. 7. Fitness curves by using different metaheuristics for Scenario I ($K = 8$).

warehouse layout with five AGVs ($K = 5$) is provided. The mesh topological roadmap for this case is shown in Fig. 8.

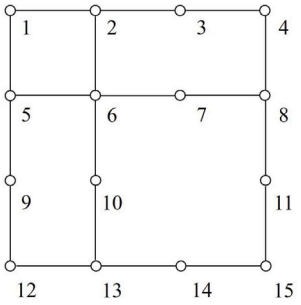
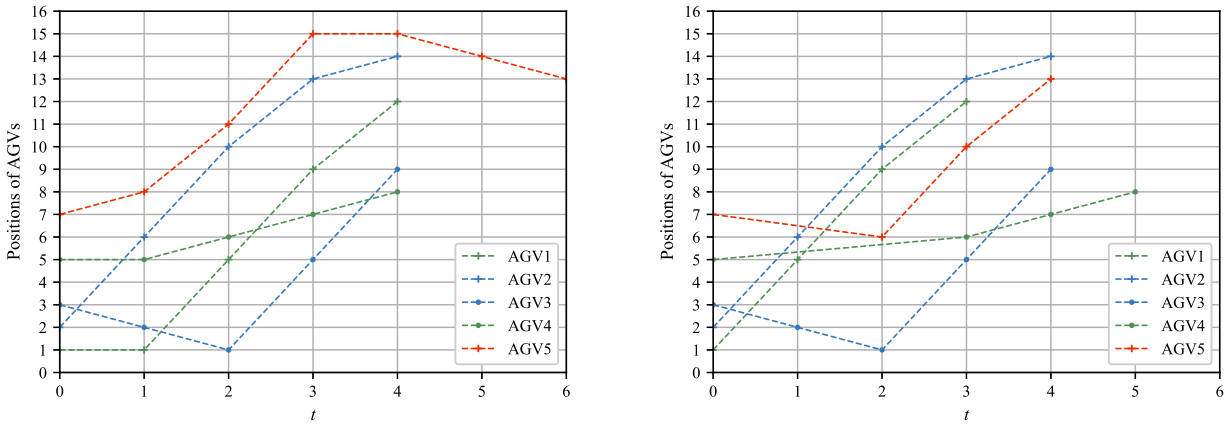


Fig. 8. Roadmap for one example with $K = 5$.

For the roadmap in Fig. 8, the setting of this roadmap is same as Section 4.1.2. In Fig. 8, five AGVs start their tasks simultaneously at time 0. The AGV tasks have been assigned in advance: $O = \{1, 2, 3, 5, 7\}$, $S = \{2, 14, 9, 8, 13\}$. The detailed results are obtained by Gurobi by using the TSN model and the proposed HGE obtained by the FTSN model, as shown in Fig. 9.

Fig. 9(a) presents the planned paths of the AGVs by using the TSN model. Fig. 9(a) shows that the five AGVs successfully move from their pickup points to delivery points. The sum of the completion times is 22. The conflicts between these AGVs are avoided because every node and every arc are not occupied by more than one AGV at any time. The TSN model considers a fixed transport time between every two connected nodes and each AGV has to wait at a certain node to avoid the collision. It can be seen that AGV 1 and AGV 4 stop at Node 1 and Node 5, respectively, within the first time slot, while AGV 5 stops at Node 15 within the fourth time slot, to avoid collision with other AGVs. AGV 2 and AGV 4 move through Node 6 at $t = 1$ and $t = 2$, respectively, since Node 6 is crowded. Furthermore, to avoid conflict with AGV 1 and AGV 3, AGV 5 moves via the path 7-8-11-15-14-13 to shorten its completion time, instead of the shortest path 7-6-10-13 without considering conflicts.

Fig. 9(b) gives the planned paths of the proposed FTSN-HGE. The sum of the completion time of the FTSN-HGE method is 20, which is shorter than the one using the TSN model. As the proposed FTSN model allows AGVs to move at a flexible speed (which can be lower than the maximum speed), AGVs no longer wait at the nodes unnecessarily to avoid conflict. AGV 5 can choose its shortest path, and the rolling resistance energy J_r can be saved for the shortest path that lowers energy consumption. The energy reduction strategy also applies to the other AGVs, as AGV 4 moves at one-third of the maximum speed from Node 5 to Node 6.



(a) Planned Paths by the TSN (b) Planned Paths by the FTSN

Fig. 9. Path results of one example of the case $K = 5$.

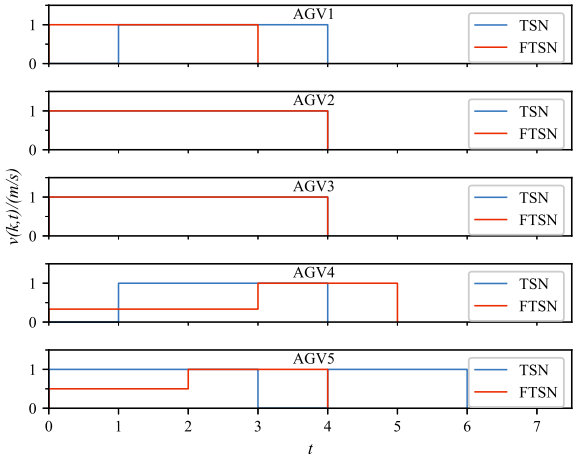


Fig. 10. Planned speeds of one example of the case $K = 5$.

Fig. 10 shows the detailed planned speeds of the AGVs when using the TSN and FTSN models. For the TSN model, AGV5 stops between $t = 3$ to $t = 4$. Therefore, AGV 5 needs to fully accelerate to its maximum speed twice. When the FTSN model is considered, AGV 5 only accelerates once by regulating the speed, and the energy consumption is thus considerably saved (by reducing the number of accelerations and decelerations).

5. Conclusions and future research

This paper investigates a new energy-efficient path planning problem of multiple AGVs, which plays a crucial role for material handling in the modern production and logistics systems. In this research, the conflict-free path and the vehicle speed are optimized jointly, using a so-called FTSN modeling framework, in order to save kinetic energy consumption without decreasing the quality of the transport productivity. Due to the nonlinearity resulting from modeling the energy-related objective and constraints, the corresponding problem is a non-convex mixed-integer nonlinear programming. To address its computational issue, we develop a dedicated hybrid metaheuristic HGE, in which new encoding schemes and algorithm procedures considers the characteristics of the planning problem. Numerical experiments have been conducted on industrial scenarios. Experiment results indicate that about 10% of the kinetic energy is reduced by using the proposed method against the conventional TSN model, without deteriorating the completion time. Experimental results show the advantage of the proposed hybrid metaheuristic compared to the genetic algorithm, the population-based incremental learning, and a commercial MINLP solver.

The proposed FTSN-HGE method shows its potential to reduce the energy use of material handling without decreasing productivity by the lexicographic strategy, which optimizes the path sequence and the vehicle speed jointly. The unnecessary vehicle waiting times resulting from preventing conflicts can be used for regulating vehicle speed. The number of accelerations and decelerations can be reduced for saving the energy consumption of material transport. The proposed planning method is environmentally friendly, and this method provides a guiding significance for promoting cleaner production of flexible manufacturing systems and smart logistics systems. The producers will save the operations costs resulting from the reduced energy consumption, without deteriorating the service of high productivity.

The energy consumption reduction requires more detailed coordination of the AGVs to regulate the speed and the path jointly in the production environment, taking into account the vehicle accelerations/decelerations and rolling resistances. To implement such an energy-efficient planner for AGV-based manufacturing systems, a designed speed controller is needed for each AGV to track the speed changes during the transport process.

Future research will investigate an extended path planning problem in which multiple tasks are considered for each AGV. In this case, it is interesting to study whether task assignment influences saving the energy consumption, and new optimization algorithms are needed to be developed to solve this extended problem.

CRediT authorship contribution statement

Jianbin Xin: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Funding acquisition. **Liuqian Wei:** Modeling, Algorithm design, Data analysis. **Andrea D'Ariano:** Supervision, Writing – original draft, Writing – review & editing. **Fangfang Zhang:** Model analysis, Writing – review & editing. **Rudy Negenborn:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This research is supported in part by the National Natural Science Foundation of China under Grant 62173311, 61703372 and 61603345, in part by the College Youth Backbone Teacher Project of Henan Province, China under Grant 2021GGJS001, in part by Henan Scientific and Technological Research Project, China under Grant 222102220123, and in part by the Training Project of Zhengzhou University, China under Grant JC21640030.

Appendix. Solution initialization

The probability matrices C_t, C_e and C_l are initialized as follows:

$$C_t^k(1) = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 0 & \cdots & 0 \end{bmatrix} \tag{A.1}$$

$$C_e^k(1) = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 1 & \cdots & 0 \end{bmatrix} \tag{A.2}$$

$$C_l^k(1) = \begin{bmatrix} 1/h & 1/h & \cdots & 1/h \\ \vdots & \vdots & & \vdots \\ 1/h & 1/h & \ddots & 1/h \\ \vdots & \vdots & & \vdots \\ 1/h & 1/h & \cdots & 1/h \end{bmatrix} \tag{A.3}$$

For the sub-population P_t, AGVs travel at the highest speed to shorten the total completion time firstly, so the probability matrix C_t^k is initialized as Eq. (A.1). While for the sub-population P_e, AGVs are operated at a lower speed to reduce energy consumption, then the probability matrix C_e^k is initialized as Eq. (A.2). The probability matrix C_l^k of the sub-population P_l is initialized as Eq. (A.3), and the probability values of each time slot are the same $(1/h)$.

References

Adamo, T., Bektaş, T., Ghiani, G., Guerriero, E., Manni, E., 2018. Path and speed optimization for conflict-free pickup and delivery under time windows. Transp. Sci. 52, 739–755.

Barak, S., Moghdani, R., Maghsoudlou, H., 2020. Energy-efficient multi-objective flexible manufacturing scheduling. J. Clean. Prod..

Bechtsis, D., Tsolakis, N., Vlachos, D., Iakovou, E., 2017. Sustainable supply chain management in the digitalisation era: The impact of automated guided vehicles. J. Clean. Prod. 142, 3970–3984.

Bemporad, A., Morari, M., 1999. Control of systems integrating logic, dynamics, and constraints. Automatica 35, 407–427.

Burer, S., Letchford, A.N., 2012. Non-convex mixed-integer nonlinear programming: A survey. Surv. Oper. Res. Manag. Sci. 17, 97–106.

Cai, L., Li, W., Luo, Y., He, L., 2022. Real-time scheduling simulation optimisation of job shop in a production-logistics collaborative environment. Int. J. Prod. Res. 1–21.

Demesure, G., Defoort, M., Bekrar, A., Trentesaux, D., Djemaï, M., 2018. Decentralized motion planning and scheduling of AGVs in an FMS. IEEE Trans. Ind. Inform. 14, 1744–1752.

Edward, Rothberg, 2014. Gurobi optimization. OR/MS Today 41, 22–24.

Fanti, M.P., Mangini, A.M., Pedroncelli, G., Ukovich, W., 2018. A decentralized control strategy for the coordination of AGV systems. Control Eng. Pract. 70, 86–97.

Farooq, B., Bao, J., Raza, H., Ma, Q., Sun, Y., 2021. Flow-shop path planning for multi-automated guided vehicles in intelligent textile spinning cyber–physical production systems dynamic environment. J. Manuf. Syst. 59, 98–116.

Halim, A.H., Ismail, I., 2019. Combinatorial optimization: comparison of heuristic algorithms in travelling salesman problem. Arch. Comput. Methods Eng. 26, 367–380.

Hauschild, M., Pelikan, M., 2011. An introduction and survey of estimation of distribution algorithms. Swarm Evol. Comput. 1, 111–128.

He, L., Chiong, R., Li, W., Budhi, G.S., Zhang, Y., 2022. A multiobjective evolutionary algorithm for achieving energy efficiency in production environments integrated with multiple automated guided vehicles. Knowl.-Based Syst. 243, 108315.

Hu, W., Mao, J., Wei, K., 2017. Energy-efficient rail guided vehicle routing for two-sided loading/unloading automated freight handling system. European J. Oper. Res. 258, 943–957.

Lee, S., Kahng, H.G., Cheong, T., Kim, S.B., 2019. Iterative two-stage hybrid algorithm for the vehicle lifter location problem in semiconductor manufacturing. J. Manuf. Syst. 51, 106–119.

Meng, X., Li, J., Zhou, M., Dai, X., Dou, J., 2016. Population-based incremental learning algorithm for a serial colored traveling salesman problem. IEEE Trans. Syst. Man Cybern. 48, 277–288.

Meng, L., Zhou, X., 2014. Simultaneous train rerouting and rescheduling on an n-track network: A model reformulation with network-based cumulative flow variables. Transp. Res. B 67, 208–234.

Miyamoto, T., Inoue, K., 2016. Local and random searches for dispatch and conflict-free routing problem of capacitated agv systems. Comput. Ind. Eng. 91, 1–9.

Murakami, K., 2020. Time-space network model and milp formulation of the conflict-free routing problem of a capacitated agv system. Comput. Ind. Eng. 141, 106270.

Nishi, T., Akiyama, S., Higashi, T., Kumagai, K., 2020. Cell-based local search heuristics for guide path design of automated guided vehicle systems with dynamic multicommodity flow. IEEE Trans. Autom. Sci. Eng. 17, 966–980.

Nishi, T., Ando, M., Konishi, M., 2005. Distributed route planning for multiple mobile robots using an augmented lagrangian decomposition and coordination technique. IEEE Trans. Robot. 21, 1191–1200.

Pérez-Cañedo, B., Concepción-Morales, E., 2020. A lexicographic approach to fuzzy linear assignment problems with different types of fuzzy numbers. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 28, 421–441.

Polten, L., Emde, S., 2021. Scheduling automated guided vehicles in very narrow aisle warehouses. Omega 99, 102204.

Riazi, S., Bengtsson, K., Lennartson, B., 2020. Energy optimization of large-scale agv systems. IEEE Trans. Autom. Sci. Eng. 18, 638–649.

Rinaldi, M., Picarelli, E., D'Ariano, A., Viti, F., 2020. Mixed-fleet single-terminal bus scheduling problem: Modelling, solution scheme and potential applications. Omega 96, 102070.

Ryck, M., Versteyhe, M., Debrouwere, F., 2020. Automated guided vehicle systems, state-of-the-art control algorithms and techniques. J. Manuf. Syst. 54, 152–173.

Saidi-Mehrabad, M., Dehnavi-Arani, S., Evazabadian, F., Mahmoodian, V., 2015. An ant colony algorithm (aca) for solving the new integrated model of job shop scheduling and conflict-free routing of agvs. Comput. Ind. Eng. 86, 2–13.

Tang, L., D'Ariano, A., Xu, X., Li, Y., Ding, X., Samà, M., 2021. Scheduling local and express trains in suburban rail transit lines: Mixed–integer nonlinear programming and adaptive genetic algorithm. Comput. Oper. Res. 135, 105436.

van Duin, J.R., Geerlings, H.H., Verbraeck, A.A., Nafde, T.T., 2018. Cooling down: A simulation approach to reduce energy peaks of reefers at terminals. J. Clean. Prod. 193, 72–86.

Wang, K.P., Huang, L., Zhou, C.G., Pang, W., 2003. Particle swarm optimization for traveling salesman problem. In: Proceedings of the 2003 International Conference on Machine Learning and Cybernetics, Xi'an, China, pp. 1583–1585.

Wu, C.g., Wang, L., 2018. A multi-model estimation of distribution algorithm for energy efficient scheduling under cloud computing system. J. Parallel Distrib. Comput. 117, 63–72.

Xin, J., Negenborn, R., Corman, F., Lodewijks, G., 2015a. Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance. Transp. Res. C 60, 377–396.

Xin, J., Negenborn, R.R., Lodewijks, G., 2014. Energy-aware control for automated container terminals using integrated flow shop scheduling and optimal control. Transp. Res. C 44, 214–230.

Xin, J., Negenborn, R., Lodewijks, G., 2015b. Event-driven receding horizon control for energy-efficient container handling. Control Eng. Pract. 39, 45–55.

Yi, G., Feng, Z., Mei, T., Li, P., Jin, W., Chen, S., 2019. Multi-agvs path planning based on improved ant colony algorithm. J. Supercomput. 75, 5898–5913.

Yin, S., Xin, J., 2019. Path planning of multiple agvs using a time-space network model. In: 2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 73–78.

Young, K., Wang, C., Le, Y.W., Kai, S., 2013. Electric Vehicle Battery Technologies. Springer, New York.

Young, C.T., Zheng, Y., Yeh, C.W., Jang, S.S., 2007. Information-guided genetic algorithm approach to the solution of minlp problems. Ind. Eng. Chem. Res. 46, 1527–1537.

Yue, L., Fan, H., Ma, M., 2021. Optimizing configuration and scheduling of double 40 ft dual-trolley quay cranes and agvs for improving container terminal services. J. Clean. Prod. 292, 126019.

Zhang, Y., Peng, Q., Yao, Y., Zhang, X., Zhou, X., 2019. Solving cyclic train timetabling problem through model reformulation: Extended time-space network construct and alternating direction method of multipliers methods. Transp. Res. B 128, 344–379.

Zou, W.Q., Pan, Q.K., Meng, T., Gao, L., Wang, Y.L., 2020. An effective discrete artificial bee colony algorithm for multi-agvs dispatching problem in a matrix manufacturing workshop. Expert Syst. Appl. 161, 113675.