Dynamic Optimization of Conflict-Free Routing of Automated Guided Vehicles for Just-in-Time Delivery

Kosei Nishida and Tatsushi Nishi, *Senior Member, IEEE*

***Abstract*— This paper studies the solution procedure to the problem of conflict-free route planning for automated guided vehicles for just-in-time delivery to minimize the earliness and tardiness of a specified delivery time while minimizing the total completion time. The characteristic of the problem is that a vehicle can have an idle time at any nodes considering dispatching of dynamic task arrivals. This problem is modeled by a time-space network where the pickup and delivery tasks are dynamically scheduled. A heuristic solution procedure is developed to derive a conflict-free routing problem such that earliness and tardiness penalties are minimized while the total completion time of each task is also minimized under dynamic task arrivals. The effectiveness of the proposed method is demonstrated by comparing the performance with Gurobi from computational results. The results show that the proposed method can obtain a better solution that is close to an optimal solution with significantly shorter computational time than that derived from Gurobi. The trade-off relationship between earliness/tardiness penalties and the total completion time is investigated in various situations. The effects of the interval of task assignment on the performance of earliness/tardiness and the total completion time are studied. The computational results show the conditions when the trade-off relationship can be obtained.**

***Note to Practitioners*—The multiple AGVs are used in public transportation and social systems with rapid development of artificial intelligence and optimization methods. The benefits of AGV affect the reduction of congestion, traffic police, legal services, and economic savings in the manufacturing and logistics industries. In order to satisfy the customer's demands, it is requested to deliver the products just-in-time for the desired time without decreasing the total system performance. This paper presents the dispatching and conflict-free routing of AGVs for just-in-time transportation while minimizing the total completion time of the tasks. In many real factories with multiple production processes, the shortest time transportation may sometimes cause the congestions of AGVs waiting for handling or overstocked. Just-in-time transportation is required not only to reduce inventories but also to supply only when it is needed. Computational results show the situations when the just-in-time delivery and the minimization of the total completion time can be achieved.**

***Index Terms*—Automated guided vehicles, conflict-free routing, just-in-time delivery, tradeoff analysis, heuristic solution procedure.**

I. Introduction

WITH the widespread use of smart factories, automation of transportation and distribution has progressed in flexible manufacturing systems to cope with the diversification of customer needs. In Industrie 4.0/Logistics 4.0, several companies are focusing on the development of automated guided vehicles (AGVs) systems in their factories to enhance manufacturing competitiveness due to computerization and intellectualization of products and services [5]. There are many advantages of using AGVs such as its versatility, flexibility concerning tool arrangements, lower infrastructure costs, and lower risks of virus infection to operate in the same workspace like human operators [29].

Multiple AGVs systems are getting more utilized in various fields such as car manufacturing, chemical plants, energy power plants, hospitals, pharmaceutical industry, and food delivery, which have mostly positive impacts on many different sectors [38]. To satisfy the recent needs for online shopping, automated transport systems are also widely used in warehousing for e-commerce where small orders with a variety of items are repeated many times. These delivery schedules are so tight to keep the delivery in the next day or even same-day deliveries. This is an elementary promise of many online retailers, especially in the business to customers (B2C) segments. In conventional warehouses, orders are collected by human operators, however, it is unproductive in recent situations. Automating the warehousing system by using the AGV system can reduce walking times of human operators [3].

The driving of automated vehicles such as drones and cars will be automated in near future. Multiple automated vehicles or drones will be used in public transportation when artificial intelligence and optimization methods are introduced. The demonstration tests of distribution using drones or automated cars are conducted around the world to cope with an increase in demand for delivery and shortages of drivers.

Manuscript received 13 March 2022; revised 11 June 2022; accepted 15 July 2022. Date of publication 3 August 2022; date of current version 3 July 2023. This article was recommended for publication by Associate Editor M. Robba and Editor M. Dotoli upon evaluation of the reviewers' comments. This work was supported by the Japan Society for the Promotion of Science (JSPS) under the Grant-in-Aid for Scientific Research (A) (18HO3826). An earlier version of this paper was presented in part at the 2019 IEEE International Conference on Systems, Man and Cybernetics, Bari, in October 2019 [DOI: 10.1109/SMC.2019.8914493]. *(Corresponding author: Tatsushi Nishi.)*

Kosei Nishida is with the Division of Mathematical Science for Social Systems, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka 560-8531, Japan.

Tatsushi Nishi is with the Division of Mechanical and Systems Engineering, Graduate School of Natural Science and Technology, Okayama University, Okayama 700-8530, Japan (e-mail: nishi.tatsushi@okayama-u.ac.jp).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2022.3194082.

Digital Object Identifier 10.1109/TASE.2022.3194082

The benefit of AGV affects the routine life of society and businesses such as reduction of congestion, traffic police, legal services, and economic savings in medical, insurance and law under hands-free driving environment [32]. In order to meet customer's demands, it is requested to deliver the products in just-in-time to minimize earliness and tardiness from a specified delivery time without decreasing the total system performance. For instance, perishable goods or cold materials need to be delivered just on time since they maintain chemical reactions which the rate can be mostly mitigated with appropriate temperatures. Since earliness and tardiness cause extra costs to maintain such environments, minimizing earliness and tardiness without significantly increasing the total completion time is one of the significant issues in real-world transportation systems.

This paper studies the dispatching and conflict-free routing problem of automated guided vehicles for just-in-time delivery (JIT-DCFRP) while minimizing the total completion time of the deliveries of dynamically given tasks. In many real factories with multiple production processes, the shortest time transportation may sometimes cause congestions of vehicles when the frequency of transportations is increased. JIT transportation is required to reduce inventories and to supply them when it is necessary. For instance, for assembly production lines in the automobile industry, some important parts need to be delivered in just-in-time because there is a risk of stopping the line which causes huge losses. And a large number of products cannot be inventoried when the idle space between stations is narrow. Therefore, JIT supply system needs to deliver the required parts to the right place at the right time [27].

The concept of just-in-time (JIT) has been used in production management and service [15]. In order to reduce inventory costs by producing only if necessary to increase customer satisfaction by supplying products in just-in-time, a solution method for the JIT flexible job shop scheduling problem has been proposed [4]. Extensive research has been addressed on vehicle routing problems for JIT delivery. Vaidyanathan *et al.* [37] addressed a capacitated vehicle routing problem for JIT delivery where the quantity to be delivered at each demand node is a function of the route taken by the vehicle assigned to serve that node so that a vehicle will have no idle time in order to minimize inventories at the demand point. A heuristic solution procedure and a lower bound relaxation were proposed. The model in this paper is different in that JIT delivery is conducted to minimize earliness and tardiness from the specified delivery time so that a vehicle can have idle time and conflict-free constraints between vehicles must be satisfied. The problem has been addressed in the classical vehicle routing problem with time windows (VRPTW). Mahmoudi and Zhou [18] formulated the vehicle routing problem with pickup and delivery for transportation systems through time-space networks to consider time window requirements. Rahman and Nielsen [28] studied the automated transport vehicle scheduling problem to minimize the total earliness and tardiness penalties. However, different from the scope of VRPTW, collision avoidance constraints for vehicles were not considered in their studies. Adamo *et al.* [1] studied the problem to determine the vehicle paths and speeds on each arc of the path in such way that no conflicts arise, the time window was met, and the total energy consumption was minimized. Fazlollahtabar *et al.* [11] addressed the earliness/tardiness minimization in a jobshop scheduling problem with the selection of AGVs. Hu *et al.* [12] presented a three-stage decomposition method for multi-AGV dispatching and routing problem. However, in those papers, collision avoidance and conflicts between vehicles were not considered in their models at the routing decisions. The dispatching and conflict-free routing problem for JIT delivery (JIT-DCFRP) has not been addressed as far as the authors know. The difficulty in the JIT routing problem is the requirement of larger solution space considering vehicle waiting places for conflict avoidance, waiting states for coordinating timing with other vehicles, and the multi-objective function with earliness/tardiness penalties together with minimization of delivery completion time under dynamic task arrivals.

In this paper, we present a time-space network modeling of the problem formulation and a heuristic solution procedure using the specific problem structure [26]. The tradeoff relationship between earliness/tardiness and the total delivery completion time is investigated to understand in what situations the JIT performance can be easily performed. The effectiveness of the proposed heuristic solution method is demonstrated by comparing the performance with an exact solution procedure from computational results. The effects of the specified delivery time and the interval of task assignment time are examined in several dynamic environments.

The contributions of this paper are as follows.

- An integer programming formulation and a heuristic solution procedure for JIT-DCFRP are developed.
- The dynamic optimization method under dynamic task arrivals is developed by using the proposed heuristic procedure.
- The effects of task frequency and the task assignment intervals on the performance of earliness/tardiness and the total delivery completion time are studied.

The rest of this paper consists of the following sections. Section II introduces literature review. Section III describes the problem definition and the mathematical formulation of the problem. Section IV develops a heuristic algorithm for JIT conflict-free routing problems for AGVs under dynamic transportation requests. In Section V, the computational results are shown to investigate the performance of the proposed method for the dynamic transport problem and the tradeoff analysis is conducted. Section VI concludes this study and states our future study.

II. Literature Review

The control algorithms and techniques of AGVs have been extensively reviewed in [31]. The routing problems of multiple AGVs have been studied for many years. The difficulties of the AGV routing problem are to determine the route for multiple AGVs to optimize various objectives such as total traveling time, throughput, earliness and tardiness, fuel consumption and so on without deadlock, conflict and congestions. Vehicle routing problems with time windows (VRPTW) have widely studied [41], [42], however, conflicts-free constraints are not considered in the VRPTW models. The routing methods for guide path based AGV systems are classified into static routing and dynamic (online) routing.

In static routing problems, the information of the transportation requests is known in advance and the path to a destination point is determined and fixed until the requests are completed. The system is modeled as the static routing problem of AGVs. Yoo *et al.* [48] used a model of the graph-theoretic approach and presented an AGV deadlock avoidance algorithm. Wu and Zhou [43] introduced the coloured resource-oriented Petri net model to find the shortest time routes for bidirectional AGV, while both deadlock and blocking were avoided. Riazi and Lennsartson [30] addressed a Benders decomposition approach using CP/SMT solvers for scheduling and conflict-free routing of AGVs.

To solve the routing problem for AGVs, various solution approaches have been developed. To obtain an optimal solution, an exact method is used for the routing problem. Desaulniers *et al.* [9] proposed an exact method that enabled to solve instances with up to four vehicles. Their approach combined a greedy search heuristic, column generation and a branch and cut procedure. Adamo *et al.* [1] proposed a branch-and-bound algorithm for a conflict-free pickup and delivery problem in which a set of pickup and delivery requests was given within time windows.

Mathematical programming approaches were also applied to the routing problem. Corréa *et al.* [7] proposed a hybrid approach consisting of a decomposition method where the scheduling problem was modeled with constraint programming and the conflict-free routing problem with mixed integer programming to solve a problem of dispatching and conflict-free routing of AGVs in an FMS. Zhang *et al.* [49] developed a greedy upper bounding and Lagrangean relaxation algorithm to solve the rerouting problem modeled as a multicommodity flow problem with link capacity design for minimization total expected travel time. Nishi *et al.* [21] addressed a bilevel decomposition approach for simultaneous production scheduling and conflict-free routing for AGVs.

To derive a near-optimal solution in a short time, a heuristic algorithm is used to solve the routing problem. Fazlollahtabar *et al.* [11] proposed a scheduling problem for multiple automated guided vehicles (AGVs) to minimize the penalized earliness and tardiness in a manufacturing system and a heuristic search algorithm and a solution methodology based on network concepts were developed. Zhang *et al.* [51] proposed a collision-free routing method for AGVs based on collision classification. The initial route of each task was predetermined by improved Dijkstra's algorithm. Daugherty *et al.* [8] presented a heuristic algorithm for minimizing the makespan required to route a set of agents inhabiting a shared guide-path network while ensuring the safety and the integrity of the generated traffic. Solichudin *et al.* [33] proposed the combined approach of the Dijkstra and Floyd-Warshall algorithm to overcome the problem of conflict-free AGV routes using time windows. Hu *et al.* [12] developed a three-stage decomposition method to the problem by combining the advantages of pre-planning algorithm and real-time planning algorithm, which combines A* algorithm with the principle of time window to plan the path of each AGV in time order [28].

Meta-heuristic algorithms have been also widely used. Umar *et al.* [36] proposed an algorithm for integrated scheduling, dispatching, and conflict-free routing of jobs and AGVs in an FMS environment using a hybrid genetic algorithm. Lyu *et al.* [17] proposed a genetic algorithm combined with the Dijkstra algorithm that was based on a time window for machine and AGV scheduling problems that considered both the number of AGVs and the conflict-free routing problem. Rahman and Nielsen [28] proposed a mixed-integer programming model and two meta-heuristic-based algorithms for an ATV materials delivery scheduling problem. Yi *et al.* [47] presented a mathematical programming model for the scheduling and collision-free routing problem of AGVs, and proposed a two-stage improved ant colony algorithm. Jeon *et al.* [14] suggested a routing method for automated guided vehicles in port terminals that uses the Q-learning technique to determine shortest-time routes inclusive of the expected waiting times. In addition, the Q-learning technique is also used in a routing method for automated guided vehicles in port terminals to determine shortest-time routes inclusive of the expected waiting times [14].

The transportation tasks are given in real time. The dynamic environment was also considered to deal with dynamic task arrivals and uncertainties in many studies [23]. In dynamic routing problems, the route selection depends on traffic conditions. If there is heavy congestion on the shortest route, then the route might be determined again by a dynamic routing system to avoid delays [13]. The resource-oriented Petri net model was constructed dynamically for the deadlock and conflict control problem of the considered AGV systems [24], [44], [45]. Xing *et al.* [39] developed a collision and deadlock avoidance policy in multi-robot systems based on the concept of glued nodes. Smolic-Rocak *et al.* [34] presented a dynamic routing method for supervisory control of AGVs that were traveling by using time windows to find candidate paths, check their feasibility and resolve time window conflicts. Nishi and Maeno [22] developed a Petri net decomposition approach for optimization of conflict-free routing problems for AGVs. Nishi and Tanaka [25], [35] solved the dynamic problem by defining the static problem every time when the set of tasks was given. Xin *et al.* [46] proposed a receding horizon planning approach for path planning of AGVs using a time-space network (TSN) model. This approach decomposes the global planning problem into smaller local planning problems using a receding horizon way to improve computational efficiency and deal with uncertainties. Hwang and Jang [13] developed a reinforcement learning-based dynamic routing algorithm with a Boltzmann softmax policy and a reward function based on the Q(λ) learning method. Chen *et al.* [6] addressed a two-stage congestion-minimizing routing method for AGVs. They utilized an improved A* algorithm and dynamic path altering algorithm for dynamic routing for AGVs.

In dynamic routing literature, several dispatching strategies have been addressed. Bilge *et al.* [2] developed multi-attribute responsive dispatching strategies for AGVs. Two approaches:

a parametric approach where the weights were computed at the beginning of each planning horizon and a dynamic approach where the system was allowed to update the weights utilizing some system statistics were compared.

The optimization approaches have been also used for multi-AGVs dispatching problems. Zou *et al.* [52] presented an effective discrete artificial bee colony algorithm for the problem. However, conflicts between multi-AGVs and JIT performance were not considered in the dispatching problem where the transportation cost including travel cost, penalty cost for violating time and AGV cost were minimized.

The minimization of the total completion time and dynamic environment were not considered in their study. Two objectives of minimizing the completion time and improving the JIT performance have a trade-off relationship. Therefore, it is important to analyze how JIT performance can be improved and how the total completion time is minimized for dynamic transportation environments.

III. Problem Description

A. Transportation Network Model

We consider an AGV system with bidirectional guide path lanes. A guide path is represented by $G = (V, E)$ where V is the set of nodes and E is the set of arcs as shown in Fig. 1. Each node $n \in V$ represents a place or a physical region where each vehicle can stop or turn. Each arc $e \in E$ represents a connection between adjacent nodes where a vehicle can travel. At each time, vehicles cannot exist on an arc. There are a specified number of vehicles where each vehicle has its starting place in the system. A delivery task consists of a pickup node, a delivery node and its desired pickup time P_l^p and the desired delivery time P_l^d for task l. The set of tasks is assigned to each vehicle such that each task can be assigned to only one vehicle at the same time and there can be some idle vehicles. No more than two vehicles can travel on the same node and the same arc at the same time. Loading and unloading are immediately started once a vehicle arrives at the pickup and delivery nodes. Therefore, no inventories are placed on the loading and unloading nodes. All tasks are completed by the AGV delivery without collisions. In order to optimize the dynamics of the AGV system, the motion of the vehicles is modeled by the discrete-time representation.

Each vehicle exists on each node at a discrete-time step in $\{0, 1, 2, \ldots, |H|\}$ where H is the set of planning periods. The position of a vehicle can be placed at a node because each node represents a physical region partitioning transport layout. Therefore, the assumption that at each time, vehicles can exit only at a node and it cannot exist at an arc is valid.

B. Dispatching and Conflict-Free Routing Problem for JIT Delivery

To conduct a delivery just at the specified time is called the JIT delivery. The static JIT-DCFRP is stated as follows. There is an AGV system with multiple vehicles with initial nodes. Given a set of delivery tasks R, the problem is to find a dispatching of tasks and conflict-free routing such that the objective function is minimized. The objective function is the weighted sum of the earliness/tardiness and the total completion time.

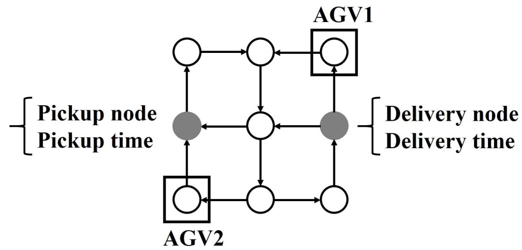

Fig. 1. Graph model of an AGV transportation system.

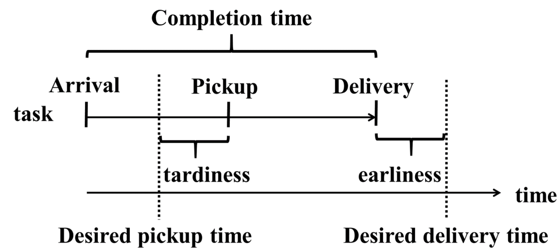

Fig. 2. Terminology defined for the JIT-DCFRP.

Fig. 2 shows an example of a JIT routing during the periods from when a task is given at a certain time until the task is completed. Once each task is assigned, each vehicle travels from its initial node to the pickup node (loading place), then travels to the delivery node (unloading place). The duration between the task arrival time and the delivery completion time is defined as the total completion time. The time difference between the pickup completion time and the desired pickup time is defined as the pickup earliness or pickup tardiness, also the time difference between the delivery completion time and the desired delivery time is called the delivery earliness or delivery tardiness. The JIT objective function is calculated by the sum of the delivery and pickup earliness/tardiness penalties. The objective function J is the weighted sum of the JIT objective function value and the total delivery completion time as shown in (1).

$$J = \alpha \sum_{l \in R} (|E_l^p - P_l^p| + |E_l^d - P_l^d|) + \beta \sum_{l \in R} E_l^d \tag{1}$$

where E_l^p is the pickup completion time of task l, E_l^d is the delivery completion time of task l, α and β are the weighting factors.

We set the objective function as the weighted sum of the earliness and tardiness from the desired delivery time from JIT perspective, and the sum of the delivery completion time to minimize the total completion time. The minimization of the sum of the delivery completion time is the same meaning as the minimization of the total completion time.

C. Dynamic Conflict-Free Routing Problem

In an actual AGV system, tasks are given dynamically in dynamic transportation environments. In this paper, the arrival time of each task and the number of tasks are randomly given.

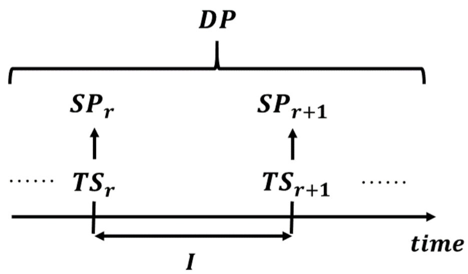

Fig. 3. The relation between static problem and tasks.

The information of the pickup/delivery nodes and the desired pickup/delivery time cannot be available until the task arrival time. The dynamic JIT-DCFRP (DP) is defined as follows. DP is the problem that the information on all tasks may not be always available at any time and the task information becomes available as time elapses. The total number of tasks that are given for the problem (DP) during the planning horizon is $|R|$. In the dynamic environment, the arrived tasks are queued in the dispatching server. After a duration of task assignment interval I time, the stocked tasks are dispatched to the AGV system. Then the assignment of uncompleted tasks and new tasks and the conflict-free routing of AGVs are determined. The task assignment interval I is the parameter from the current time to the next task assignment. The static JIT-DCFRP is solved to determine the task assignment and routing every task assignment time. r is the number of rescheduling from time step 0, etc. $r = 1$ is the initial scheduling, $r = 2, 3, \ldots, R$ is the second, the third, $\ldots$, and rth re-optimization after the assignment interval I. The set of tasks given for the rth task assignments is defined TS_r. Note that no new tasks are dispatched between the time when TS_r is given and the time when TS_{r+1} is given. Therefore, when TS_r is dispatched, the rth problem can be regarded as a static JIT-DCFRP (SP_r). The relation between the problem DP and rth static problem (SP_r) is shown in Fig. 3.

In dynamic environments, the information of all tasks is not always available at the current time step. Therefore, in the static problem (SP_r) ($r = 1, 2, \ldots, |H|$), the information of tasks which have been already given at the current time ($\bigcup_{r'=1}^{r} TS_{r'}$) is used. The static problem SP_r at a time step $h \in H$ is defined as follows.

- Input: Uncompleted tasks which are dispatched until the current time h, the current node of vehicles at time h which are determined in (SP_{r-1})
- Objective function: Minimization the weighted sum of the total earliness and tardiness and the total completion time
- Decision variables: Task assignment and conflict-free routing during $[h, h + |T|]$ where $|T|$ is the planning horizon of the static problem SP_r

The set of uncompleted tasks consists of the tasks for which pickup has not yet been completed and tasks for which pickup has been already completed. In the problem (SP_r), the task assignment cannot be changed for the tasks in which the pickup has been already completed. The dynamic problem DP is solved by solving the static problem (SP_r) ($r = 1, 2, \ldots$) in the sequential order.

D. Problem Formulation

The static problem (SP_r) at time step $h \in H$ is formulated as an integer programming problem as follows. The constraints on the traveling (3)-(7), (9) and (10) of each vehicle are considered in [38].

[SETS]

M:	Set of vehicles.
N:	Set of nodes.
A_i:	Set of adjacent nodes of node i, which includes node i itself (stopping at node i).
T:	Set of time periods during static planning horizon.
$L \subset \bigcup_{r'=1}^{r} TS_{r'}$:	Set of tasks that has been already completed.
$L^p \subset L$:	Set of tasks in which pickup is not completed.
$L^d \subset L$:	Set of tasks in which pickup is completed The information of L_1 and L_2 is obtained by solutions of SP_{r-1}.

[DECISION VARIABLES]

$x_{i,j,t}^m$:	1 If vehicle $m \in M$ travels from node $i \in N$ to node $j \in N$ at time $t \in T$, and 0 otherwise.
D_t^l:	1 if delivery of task l is not completed at time t, and 0 otherwise.
K_t^l:	1 if pickup of task l is not completed at time t, and 0 otherwise.
y_l^m:	1 if task l is assigned to vehicle m, 0 otherwise.
f_l:	Absolute value of difference of delivery complete time and delivery time.
e_l:	Absolute value of difference of pickup complete time and pickup time.

[PARAMETERS]

$s_m \in N$:	Initial node of vehicle m.
$v_l \in M$:	Vehicle to which task $l \in L^d$ is assigned.
$u_l \in N$:	Pickup node of task l.
$g_l \in N$:	Delivery node of task l.
$P_l^p \in T$:	Desired pickup time of task l.
$P_l^d \in T$:	Desired delivery time of task l.
α :	Weighting factor for total earliness and tardiness.
β :	Weighting factor for the total complete time.

The information of s_m is obtained by solutions of the problem SP_{i-1}. The problem SP_r is formulated as follows.

$$(SP_r) \quad \min\ J$$

$$\text{where} \quad J = \alpha \sum_{l \in L} (f_l + e_l) + \beta \sum_{l \in L} \sum_{t \in T} D_t^l \tag{2}$$

$$\sum_{j \in A_i} x_{i,j,t}^m \leq 1 (i \in N, m \in M, t \in T) \tag{3}$$

$$\sum_{i\in N}\sum_{j\in A_i} x^m_{i,j,t} = 1(m\in M, t\in T) \tag{4}$$

$$\sum_{j\in A_i} x^m_{j,i,t} = \sum_{k\in A_i} x^m_{i,k,t+1}(i\in N, m\in M, t\in T) \tag{5}$$

The objective function (2) consists of the weighted sum of the total earliness/tardiness penalties and the total delivery completion time where α and β are the weighting factors. Equation (3) states that each vehicle can only travel to one of the adjacent nodes which are connected with node i. Equation (4) ensures that each vehicle chooses only one arc to travel or node to stop. Equation (5) indicates that each vehicle travels to node k that is connected with node i at a time $t+1$ if it travels to node i at time t.

$$\sum_{m\in M}\sum_{j\in A_i} x^m_{j,i,t} \leq 1 \ (i\in N, t\in T) \tag{6}$$

$$\sum_{m\in M}(x^m_{i,j,t} + x^m_{j,i,t}) \leq 1 \quad (i,j\in N, i\neq j, t\in T) \tag{7}$$

$$\sum_{j\in A_i} x^{m_1}_{j,i,t} + \sum_{m_2\in M\backslash\{m_1\}}\sum_{k\in A_i} x^{m_2}_{i,k,t} \leq 1 \ (i\in N, m_1\in M, t\in T) \tag{8}$$

Equation (6) ensures that only one vehicle can travel to node i at the same time. Equation (7) means that only one vehicle can travel between node i and node j at the same time. Equation (8) represents that the vehicle that travels to node i and that from node i at the same time cannot be on the same node simultaneously.

$$\sum_{j\in A_{s_m}} x^m_{s_m,j,0} = 1(m\in M) \tag{9}$$

$$\sum_{i\in N\backslash\{s_m\}}\sum_{j\in A_i} x^m_{i,j,0} = 0(m\in M) \tag{10}$$

Equations (9) and (10) ensure that vehicle m can travel from only the initial node s_m at the time 0.

$$\sum_{m\in M} y^m_l = 1(l\in L) \tag{11}$$

$$\sum_{l\in L} y^m_l = 1(m\in M) \tag{12}$$

$$K^l_t - \sum_{m\in M}(y^m_l x^m_{u_l,u_l,t}) \leq K^l_{t+1}(l\in L, t\in T) \tag{13}$$

$$\sum_{m\in M}(y^m_l x^m_{u_l,u_l,t}) + K^l_{t+1} \leq 1(l\in L, t\in T) \tag{14}$$

$$D^l_t - \sum_{m\in M}(y^m_l x^m_{g_l,g_l,t}) \leq D^l_{t+1}(l\in L, t\in T) \tag{15}$$

$$\sum_{m\in M}(y^m_l x^m_{g_l,g_l,t}) + D^l_{t+1} \leq 1(l\in L, t\in T) \tag{16}$$

$$K^l_0 = 1,\ D^l_0 = 1(l\in L) \tag{17}$$

$$K^l_{t+1} \leq K^l_t, D^l_{t+1} \leq D^l_t, K^l_t \leq D^l_t(l\in L, t\in T) \tag{18}$$

Equations (11) and (12) indicate that each task is exactly assigned to one vehicle and one task, respectively. Equations (13) and (14) state that a binary variable K^l_t takes a value of 1 until vehicle $m\in M$ completes the pickup for task l, and 0 when the vehicle completes its pickup. Equations (15) and (16) denote that D^l_t takes a value of 1 until vehicle $m\in M$ completes the delivery for task l, and 0 when the vehicle completes its delivery. Equation (17) ensure that variables K, D take the value of 1 at $t=0$. Equation (18) indicates that once K^l_t and D^l_t take a value of 0, then they take only 0 at subsequent times, and variable D^l_t cannot take 1 until variable K^l_t takes a value of 0.

$$-f_l \leq \sum_{t\in T} D^l_t - P^d_l \leq f_l(l\in L) \tag{19}$$

$$-e_l \leq \sum_{t\in T} K^l_t - P^p_l \leq e_l(l\in L) \tag{20}$$

Equations (19) and (20) indicate that the absolute value of the difference between the delivery completion time and the desired delivery time is f_l, and the absolute value of the difference between the pickup completion time and the desired pickup time is e_l.

$$x^m_{i,j,t} \in \{0,1\}(i,j\in N,\ m\in M,\ t\in T) \tag{21}$$

$$D^l_t,\ K^l_t \in \{0,1\}(l\in L,\ t\in T) \tag{22}$$

$$y^m_l \in \{0,1\}(l\in L,\ m\in M) \tag{23}$$

$$f_l,\ e_l \in \mathbb{N}^+_0(l\in L) \tag{24}$$

Equations (21)–(24) are the constraints for decision variables where $\mathbb{N}^+_0$ is defined as the set of non-negative integers.

Because (13)-(16) are nonlinear, these terms should be converted into linear programming formulations to solve the integer linear program by using Gurobi or CPLEX. The decision variables and constraints are added to convert them by setting $y_{l,m}x_{m,u_l,u_l,t} = B_{l,t,m}$, $y_{m,l}x_{m,g_l,g_l,t} = C_{l,t,m}$

$B_{l,t,m}$: 1 if $y_{m,l}x_{m,u_l,u_l,t}$ is 1, and 0 otherwise
$C_{l,t,m}$: 1 if $y_{m,l}x_{m,g_l,g_l,t}$ is 1, and 0 otherwise

$$1 - x_{m,u_l,u_l,t} - y_{m,l} + B_{l,t,m} \geq 0 \tag{25}$$

$$x_{m,u_l,u_l,t} - B_{l,t,m} \geq 0 \tag{26}$$

$$y_{m,l} - B_{l,t,m} \geq 0 \tag{27}$$

$$1 - x_{m,g_l,g_l,t} - y_{m,l} + C_{l,t,m} \geq 0 \tag{28}$$

$$x_{m,g_l,g_l,t} - C_{l,t,m} \geq 0 \tag{29}$$

$$y^m_l - C_{l,t,m} \geq 0 \tag{30}$$

$$B_{l,t,m} \in \{0,1\},\ C_{l,t,m} \in \{0,1\} \tag{31}$$

$$x_{m,u_l,u_l,t} - B_{l,t,m} \geq 0 \tag{32}$$

$$(l\in L,\ m\in M,\ t\in T) \tag{33}$$

The formulation of (SP_r) has $|M||N|^2|T|+2|L||T|+|L||M|$ 0-1 variables. The solution space of the problem increases exponentially with $|N|$, $|M|$, and $|T|$. A heuristic solution procedure is required since (SP_r) $r=1,\cdots,|H|$ is repeatedly solved to derive a feasible solution of DP.

IV. Heuristic Solution Procedure for the Static JIT-DCFRP

A. Outline of the Proposed Algorithm

The integer programming problem (SP_r) can be solved using a general-purpose solver such as CPLEX or Gurobi solver if the scale of the problem is sufficiently small, however, it requires too much computational time to solve SP_r if the number of vehicles and tasks is increased. It is not practical to solve the problem under dynamic task arrivals. Therefore, we propose a heuristic solution procedure to obtain an approximate solution close to an optimal solution in a shorter computing time. From the formulation provided above, the problem structure can be analyzed mathematically. Once the decision variables for task assignment are fixed and the conflict-free constraints (6)-(8) are eliminated, the problem (SP_r) can be independently solved for each vehicle. The main idea of the heuristic procedure is that the routing problem is solved for each vehicle with a given task assignment without considering conflicts to minimize the objective function for each task. In general, the shortest path problem of one vehicle can be easily solved by Dijkstra's algorithm. The benefit of the proposed heuristic procedure is that the problem can be solved in $\mathcal{O}(|N|^2|T|^2)$ for each vehicle by using Dijkstra's algorithm without enumerating an exponential computational order of task dispatching, routing and timing. However, in the JIT routing problem, the decision variable represents the routing of vehicles including time-index because it is required to consider the waiting/idling of vehicles at a node. The time-space network [18] can represent the nodes where the vehicle stays at a node and the time when the vehicle leaves from the node. The JIT routing problem for each vehicle (SP_r^m) ($m \in M$) can be reformulated as the optimization problem for each vehicle by using the time-space network.

If there is a conflict between vehicles, a deadlock avoidance heuristic (priority rule) [25] is applied to the current solution to satisfy conflict-free constraints after the routing for each vehicle. The penalty of the degree of constraint violations for the collision on each node and each lane is embedded into the objective function as a penalty function. The outline of the heuristic algorithm to solve (SP_r) given at time step h is as follows. The set of M is the set of vehicles, and the function $f(S)$ is the value of the objective function for solution S.

Input : The parameters of the problem: the number of iterations θ_1 and θ_2, the weighting factors (α, β and γ) and uncompleted delivery tasks in the tasks which are given up to time step h and node where vehicles exist at time step h, which are determined in SP_{r-1}.

Output : task assignment, each vehicle routing from time steps h, the value of the objective function.

Step 1 The initial task assignment is optimized by the greedy search with θ_1 times of iterations.

Step 2 The set of vehicles that does not have a routing plan M' is set to $\emptyset$.

Step 3 Randomly select vehicle $m \in M'$ and the routing problem for vehicle m (SP_r^m) is solved by Dijkstra's algorithm by using the time-space network. $M' = M' \cup \{m\}$.

Step 4 If $M' \neq M$, return to Step 3.

Step 5 If there are conflicts between vehicles, the deadlock avoidance algorithm (Steps (a)–(c)) is conducted as follows to find a feasible solution S^{new}.

[Deadlock avoidance heuristic] [25]

Step a) Collision avoidance with permanently stopping vehicle
Detect a collision with a vehicle that is permanently stopping at a node. If it is found, a permanently stopping vehicle is moved to an adjacent node.

Step b) Collision avoidance with traveling vehicle
Detect a collision with a vehicle that is traveling into the same node. If it is found, a traveling vehicle is stopped one time based on a priority rule.

Step c) Collision avoidance with temporarily stopping vehicle
Detect a collision with a vehicle that is temporarily stopping at a node. If it is found, the traveling vehicle is delayed to avoid a collision.

Step 6 If $f(S^{new}) < f(S)$ then $S = S^{new}$

Step 7 If the number of iterations reaches a fixed number of times (θ_2), then the algorithm is completed. Otherwise, the task assignment is changed, and return to STEP 2.

The proposed algorithm has the following advantages:

- The routing problem is solved for each vehicle using the time-space network. It can drastically reduce the computing time for the total AGV system.
- The task assignment and routing for each vehicle are repeated until a near-optimal solution is derived.
- A deadlock avoidance algorithm is conducted to ensure the feasibility of routing derived at the routing for each vehicle.
- The proposed algorithm can be easily implemented via a parallel processing environment into the computation of each vehicle for real-time computations.

B. Initial Task Assignment

The task assignment is related to the overall objective function value, it is assigned such that the total completion time is not too fast or too late from the desired pickup and delivery time. The initial task assignment is determined by using two steps local search based on the simple JIT evaluation. The first search is repeated a fixed number of times (θ_1). The simple JIT evaluation is the earliness and tardiness for each task based on the shortest path without any collisions. The computational time does not increase even if many iterations are performed because the routing considering conflicts is not required. A near-optimal task assignment may be found by changing the task assignment by a neighborhood search. The initial routing procedure is prepared to minimize the computation time not to decrease the number of collisions.

C. JIT Routing Optimization Using Time-Space Network

The set of states $\mathcal{P}$ is defined where each state represents that a vehicle exists at node n at time t as shown in (34).

$$\mathcal{P} = \{\sigma | \sigma = (t, n), t \in T, n \in N\} \tag{34}$$

$$d_{\sigma_a \sigma_b} = \begin{cases} \alpha \max(0, T_{set} - t_{\sigma_b}) \\ \quad + \gamma \max(0, t_{\sigma_b} - T_{set}) \quad (n_{\sigma_a} \neq n_{\sigma_b}) \\ \beta(t_{\sigma_b} - t_0) + \gamma \max(0, t_{\sigma_b} - T_{set}) \\ \qquad\qquad\qquad\qquad (n_{\sigma_a} = n_{\sigma_b}) \end{cases} \tag{35}$$

Equation (35) is the cost $d_{\sigma_a \sigma_b}$ from state σ_a to state σ_b in $\mathcal{P}$. n_{σ_i} is the node when the state is σ_i, t_{σ_i} is the time when the state is σ_i, and an initial state is σ_0. T_{set} represents the time when a vehicle should arrive at its delivery node to complete the task at the desired delivery time. The case when $n_{\sigma_a} \neq n_{\sigma_b}$ represents the cost of traveling to another node. Due to the first term of $\max(0, T_{set} - t_{\sigma_b})$, the cost of traveling to another node increases as the time of the state after the transition is earlier than time T_{set}. The case $n_{\sigma_a} = n_{\sigma_b}$ represents the cost of stopping at the same node. Due to the term of $t_{\sigma_b} - t_0$, the cost of stopping at the same node increases as the time of the state after the transition is later than the time of the initial state. The term of $\max(0, t_{\sigma_b} - T_{set})$ increases the cost when the time of the state after the transition is later than time t. α, β, γ are the weighting factors. As γ increases, the cost increases as the time after the transition is later than T_{set}. The cost $d_{\sigma_a \sigma_b}$ from state σ_a to state σ_b is replaced by the path length from node σ_a to node σ_b. By using Dijkstra's algorithm, it is possible to obtain the routing of each vehicle considering JIT performance without considering the collision between vehicles.

D. Deadlock Avoidance Algorithm

The individual generation of the routing for a vehicle may be infeasible for the AGV system because it does not consider collisions with other vehicles. Therefore, it is necessary to use a deadlock-free algorithm (DFA) to generate a feasible solution. Nishi and Tanaka [25] developed a deadlock and blocking avoidance algorithm. An initial routing of each vehicle is compared sequentially from time step 0 to check whether there are conflicts or not. If there are conflicts, one vehicle stops and delays traveling to avoid collisions based on their priority rules. It is repeated until there are no conflicts at all-time step. At Step 5 of the algorithm, we use the algorithm to guarantee feasibility.

The steps of the construction of a feasible solution by DFA are as follows.

Step 1 Shortest path routing from an initial point to its unloading/loading points
For the tentative solution, the shortest path without considering conflict and blocking to its unloading/loading points is added to the routing for the associated vehicles when some vehicles do not arrive to its unloading/loading points by the tentative solution.

Step 2 Route generation to avoid blocking
Blocking exists if the following conditions are satisfied. There is no conflict-free route without the generation of a route to avoid blocking.

a) The destination node for vehicle i $(1 \leq i \leq m)$ is the same with that of other vehicle j $(i \neq j)$.

b) The route of vehicle i $(1 \leq i \leq m)$ is included in the route of vehicle j $(i \neq j)$.

To eliminate physically infeasible situations, the additional routing for vehicle i is generated. The new destination for vehicle i is selected from the waiting places where the node is not included in the route for all vehicles. The shortest path to its new destination is added to the route of vehicle i. The step 2 is repeated until all the blocking situations are eliminated.

Step 3 Priority determination
If the destination for vehicle i is included in the routing for other vehicles, conflict occurs at the destination node for vehicle i. To resolve the conflict, priority is given to the associated vehicles. A transition to the conflicting nodes is restricted by using the priority rule.

Step 4 Conflict-free routing generation
The routing derived at the previous steps is feasible when the conflicts of nodes and edges are eliminated. A feasible schedule for AGVs can be obtained by delaying the routing schedule corresponding to each vehicle route.

Deadlock prevention policies have been actively studied by siphon analysis and liveness analysis on the systems modeled by Petri nets [10], [16], [40]. Wu and Zhou [44], [45] addressed a monitor addition method to the markings on the Petri net, which consists of the transport routes of each AGV to determine if collisions or deadlocks occur. In our study as well, collisions and deadlocks are determined by monitoring the created transport route at each time step, and avoidance measures are applied. The major differences from the conventional deadlock avoidance algorithm are described below.

- For Wu *et al.* [44], a bidirectional lane is permanently unidirectionalized when avoiding a collision on the lane (passing collision) by reroute planning. On the other hand, in our method [25], the parameter of time window length is used and the lane is temporarily made unidirectional so that the lane can be used again in both directions.
- For Wu *et al.* [44], the travel from node x to node y is prohibited for the lane connecting node x and the adjacent node y, another transport route from node x to node y is created. However, if node y is not the loading/unloading point of the assigned task, the AGV does not necessarily have to pass through node y. In this study, when reroute planning, the route is efficiently planned by considering the nodes that the AGV should reach.

E. Procedure to Decrease the Number of Collisions on the Initial Routing

If the strategy of collision avoidance is used, the delay is caused in most cases. If the number of collisions on the initial route of each vehicle increases, the difference between the pickup time and delivery time may increase. A penalty term is added to the transition cost to drive a conflict-free shortest path algorithm such that the number of collisions is reduced.

$$C(m, t, n) = \sum_{m' \in M'} \delta \times (x_{i,n,t-1}^{m'} + x_{i,n,t}^{m'} + x_{i,n,t+1}^{m'})(i \in A_n, m' \in M') \tag{36}$$

Eq. (36) is the penalty cost for the transition to $\sigma = (t, n)$ of vehicle m. M' is defined as the set of vehicles that have already determined the routing before vehicle m. δ is a constant representing the weighting factor for collision avoidance.

This procedure is added in the algorithm because the routing of each AGV becomes different from already determined routings by incurring the penalty function.

Proposition 1: The penalty function for vehicle m at time t on node n is represented by $C(m, t, n) = \sum_{m' \in M'}(x^{m'}_{i,n,t-1} + x^{m'}_{i,n,t} + x^{m'}_{i,n,t+1})$

Proof: Let M' be defined as the set of vehicles that have already determined the routing before vehicle m. If $\sum_{m' \in M'} x^{m'}_{i,n,t} = 1$, to satisfy (6), $\sum_{m \in M \setminus M'} x^{m}_{i,n,t}$ should be 0. Else if $\sum_{m' \in M'} x^{m'}_{i,n,t-1} = 1$, $\sum_{m' \in M'} x^{m'}_{n,j,t}$ $(j \in A_n)$ is set to 1 from (5). In this case, if $\sum_{m \in M \setminus M'} x^{m}_{i,n,t} = 1$, (8) cannot be satisfied. If $\sum_{m' \in M'} x^{m'}_{i,n,t+1} = 1$, $\sum_{m' \in M'} x^{m'}_{j,n,t}$ $(j \in A_n)$ is 1 from (5). In this case, if $\sum_{m \in M \setminus M'} x^{m}_{i,n,t} = 1$, (8) cannot be satisfied. Therefore, In the case of $\sum_{m' \in M'} x^{m'}_{i,n,t}$ or $\sum_{m' \in M'} x^{m'}_{i,n,t-1}$ or $\sum_{m' \in M'} x^{m'}_{i,n,t+1}$ is 1, $\sum_{m \in M \setminus M'} x_{m,i,n,t}$ must to be 0 to satisfy conflict-free constraints with already determined routings by $m' \in M'$. ■

F. Near-Optimality and Computational Complexity of the Proposed Algorithm

In the proposed algorithm, the optimal task assignment can be obtained because all combinations of task assignment are enumerated in a finite time of iterations if θ_2 is sufficiently large. If the optimal task assignment is found, the optimal routing for each vehicle is obtained with no considering conflicts under the task assignment. The conflicts are eliminated in the smallest delay. Therefore, it is guaranteed that the approximate solution is obtained. For an initial assignment, the matrix which is composed of the shortest traveling time from one node to another node is used. The computational complexity of the simple JIT evaluation (total earliness/tardiness) is $\mathcal{O}(|L|)$ because the shortest traveling time is obtained from the matrix the number of iterations is $|L|$ to task completion time prediction for all of the tasks. Therefore, the computational complexity of the initial task assignment is determined by the greedy search using the simple JIT evaluation is $\mathcal{O}(\theta_1|L|)$. The computational complexity of the JIT routing of one vehicle is $\mathcal{O}(|N|^2|T|^2)$ because the number of the states in the time-space network is $|N||T|$ and Dijkstra's algorithm is conducted in the time-space network to determine the routing of one vehicle. Therefore, the computational complexity of JIT routing of all vehicles is $\mathcal{O}(|L||N|^2|T|^2)$. Due to the JIT routing of all vehicles is repeated θ_2 times, the computational complexity of task reassignments is $\mathcal{O}(\theta_2|L||N|^2|T|^2)$. The complexity of the deadlock avoidance algorithm for bidirectional layout is $\mathcal{O}(|E||M||T|^2|N|^2)$ for eliminating conflicts on each arc and $\mathcal{O}(|M|^2|T|^2|N^2|)$ for eliminating conflicts on each node in worst cases.

V. Computational Study

The performance of the proposed method in the static problems and the dynamic problems is compared with that of an exact solution method by Gurobi to confirm the effectiveness of the proposed method. Then the trade-off relationship between the JIT performance and the total completion time is investigated under different task assignment intervals and task arrival frequency in order to evaluate the effect of the situation on JIT delivery.

TABLE I
Initial Node for 5 Vehicles for an Illustrative Example

Vehicle number	Initial node
1	1
2	3
3	25
4	12
5	7

TABLE II
Task Data for 10 Tasks Example Problem

Task	Arrival time	Pick up node	Delivery node	Pick up time	Delivery time
1	1	6	11	11	21
2	4	8	13	14	24
3	6	30	4	16	26
4	8	17	29	18	28
5	11	12	17	21	31
6	13	20	25	23	33
7	14	24	21	24	34
8	15	3	16	25	35
9	17	5	10	27	37
10	20	26	8	30	40

A. Illustrative Example

An illustrative example is provided to show the results of the proposed heuristic procedure for a dynamic problem. The information of initial nodes of 5 vehicles and 10 tasks is shown in Table I and Table II. The assignment time intervals are set to 5 and both weighting factors α and β are set to 1. The program was implemented with Microsoft Visual C++ 2019 and an Intel(R) Core(TM) i7-6700 3.40GHz with 8.0GB memory was used in the computational study. A general-purpose solver Gurobi 9.00 was used for the comparison.

SP_r is defined at time step k, the task information which is given before time step k is available, however, tasks which are given after time step k is not available. For example, when SP_r is defined at time step 10, the information of tasks between task 1 and task 4 is available, however, the information of tasks between task 5 and task 10 is not available. Firstly, a static problem SP_1 is defined, when 5 steps elapse. In the SP_1, task assignment of task 1 and task 2, which are given during $t = 0$ and $t = 5$, and routing for all vehicles are determined. In this example problem, task 1 is assigned to vehicle 1 and task 2 is assigned to vehicle 4. The routing result of each vehicle is shown in Fig. 4 showing the vehicle number on the same node at the same time. Then, a static problem SP_2 is defined at step 10. In the SP_2, task assignment of task 1 and task 2 in which the pickup tasks have not been completed, and task 3 and task 4, which are given during $t = 5$ and $t = 10$.

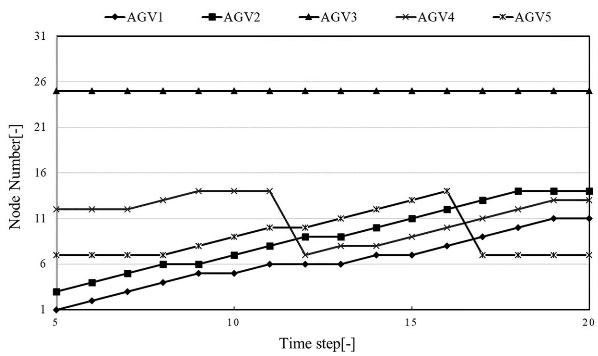

Fig. 4. The route of vehicles determined in SP_1.

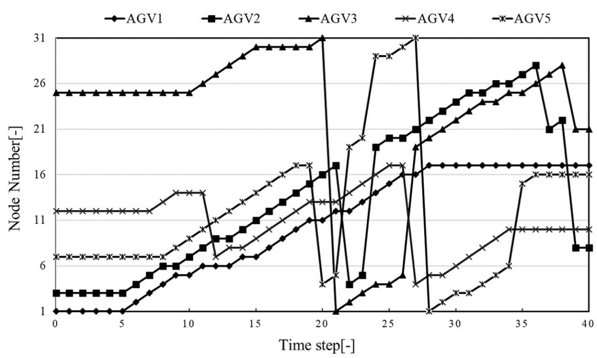

Fig. 5. The route of vehicles determined in DP.

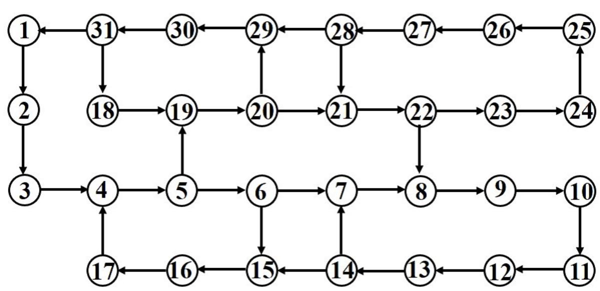

Fig. 6. Small scale transportation system layout.

The dynamic problem DP is solved by solving the static problems $SP_i (i = 1, 2, \ldots)$ in order until all tasks are completed its delivery. The route for each vehicle obtained by solving the dynamic problem is shown in Fig. 5.

TABLE III

COMPUTATIONAL RESULTS OF THE PROPOSED METHOD IN STATIC PROBLEM ($|M|, |L| = 5, \alpha = 1, \beta = 1$)

	Proposed method		Gurobi	
Case	Obj.	Time(s)	Obj.	Time(s)
1	101	1.22	100	63.601
2	88	1.187	88	135.64
3	125	1.209	125	37.237
4	117	1.245	115	65.229
5	138	1.229	130	431.355
6	130	1.225	126	1575.877
7	140	1.196	132	243.134
8	128	1.197	121	130.222
9	132	1.123	128	130.641
10	122	1.166	113	90.164
Ave.	122.1	1.1997	117.8	290.31

TABLE IV

COMPUTATIONAL RESULTS OF THE PROPOSED METHOD IN STATIC PROBLEM ($|M|, |L| = 7, \alpha = 1, \beta = 1$)

	Proposed method		Gurobi	
Case	Obj.	Time(s)	Obj.	Time(s)
1	144	1.587	140	274.204
2	128	1.523	122	500.411
3	190	1.604	176	781.704
4	167	1.628	163	2448.494
5	193	1.478	181	1930.19
6	180	1.576	171	3600
7	189	1.568	184	2092.471
8	185	1.476	166	3105.319
9	198	1.569	184	1888.95
10	178	1.563	160	497.908
Ave.	175.2	1.5572	164.7	1711.965

TABLE V

COMPUTATIONAL RESULTS OF THE PROPOSED METHOD IN STATIC PROBLEM ($|M|, |L| = 9, \alpha = 1, \beta = 1$)

	Proposed method		Gurobi	
Case	Obj.	Time(s)	Obj.	Time(s)
1	208	2.121	181	797.334
2	170	2.11	160	3600
3	238	2.028	231	3600
4	228	2.002	221	3600
5	260	2.018	325	3600
6	249	2.394	238	3600
7	272	2.175	471	3600
8	297	2.462	1243	3600
9	268	2.07	252	3600
10	236	2.144	260	3600
Ave.	242.6	2.1524	358.2	3319.733

B. Performance of the Proposed Method

We investigate the performance of the proposed method by comparing it with the results of solving the static problem using a general-purpose solver (Gurobi). The computation is stopped and the best solution is derived when the computation time of Gurobi reaches 3600 seconds. A transportation layout with 31 nodes and 38 arcs in Fig. 6 was used.

We compare the performance of the proposed method with that of Gurobi to solve the static problem. The weighting factors α, β and γ are set to 1, 1 and 5000. The penalty term for decreasing collisions δ is set to 10000. The number of repetitions $\theta_1 = 500$ and $\theta_2 = 100$. The objective function value and the computational time for 10 cases are shown in Table III-V.

The proposed method can derive feasible solutions with a significantly shorter time and the value of the objective function is close to those of Gurobi. Furthermore, in the case of 9 vehicles, the average of objective function values is smaller in the proposed method than those derived by Gurobi. Gurobi could not find a good solution within 3600 seconds because the number of combinations of task assignment is increased and the problem becomes extremely difficult when the number of vehicles is increased. However, the solutions of the proposed method are not better compared to the other cases in 2 cases

TABLE VI
COMPUTATIONAL RESULTS OF THE PROPOSED METHOD IN DYNAMIC PROBLEM ($|M| = 5, |R| = 10, \alpha = 1, \beta = 1$)

	Proposed method		Gurobi	
Case	Obj.	Time(s)	Obj.	Time(s)
1	243	5.096	232	373.787
2	204	4.715	229	151.652
3	316	6.196	310	4222.601
4	309	4.899	314	1164.666
5	287	6.062	275	911.675
6	328	4.857	294	1073.689
7	285	4.845	299	1765.093
8	310	4.887	321	2970.17
9	283	7.01	270	473.766
10	290	6.095	275	1673.883
Ave.	285.5	5.4662	281.9	1478.0982

TABLE VII
COMPUTATIONAL RESULTS OF THE PROPOSED METHOD IN DYNAMIC PROBLEM ($|M| = 7, |R| = 10, \alpha = 1, \beta = 1$)

	Proposed method		Gurobi	
Case	Obj.	Time(s)	Obj.	Time(s)
1	222	5.936	213	765.452
2	207	5.918	203	298.275
3	269	8.26	258	1462.35
4	267	6.056	255	1312.7
5	285	7.59	261	1072.817
6	261	5.959	262	1496.099
7	274	7.408	261	1014.824
8	269	7.536	278	1113.478
9	260	7.373	251	674.721
10	297	7.485	277	1312.865
Ave.	261.1	6.9521	251.9	1052.3581

for 7 vehicles. The reason is that the solution is trapped into a bad local optimum solution. Then, the performance of the proposed method for the dynamic problem DP is investigated. We compare the results obtained by solving each static problem SP_r ($r = 1, 2, \ldots$) by using a general-purpose solver (Gurobi) with that by using the proposed algorithm. A transportation layout with 31 nodes and 38 arcs in Fig. 6 was used. The weighting factors α, β and γ are set to 1, 1 and 5000. The penalty for decreasing collisions δ is set to 10000. The number of repetitions $\theta_1 = 500$ and $\theta_2 = 100$. The objective value and the computational time for 10 cases are shown in Table VI and VII.

It is also confirmed that the proposed method can obtain near-optimal solutions with a significantly shorter time than Gurobi in all cases and the derived solutions are very close to those of Gurobi. Furthermore, the value of the objective function is smaller in the proposed method than that in Gurobi in 4 cases for $|M| = 5, |R| = 10$ and 2 cases for $|M| = 7, |R| = 10$. Because the worse optimal solution for the later arrival tasks from some optimal solutions is selected, the value of the objective function is smaller in the proposed method than that in Gurobi in some cases. In the dynamic problem, when the static problem is solved, the information of tasks that are given after the current time cannot be available. Therefore, it cannot be predicted which optimal solution should be selected from some optimal solutions in order to improve the JIT objective and the completion time of tasks that are given after the current time.

TABLE VIII
CASE STUDY FOR THE DYNAMIC PROBLEM INSTANCES

	layout	assignment time interval
Case 1	1	30
Case 2	1	10
Case 3	2	30
Case 4	2	10

C. Computational Study on Trade-off Analysis Under Different Assignment Time Intervals

In this section, we investigate what situation and when the proposed JIT delivery is effective from the viewpoints of task dispatcher control and time intervals. Tasks may arrive randomly and they are immediately dispatched to AGVs to minimize tardiness, however, for JIT environments, earliness can be minimized when tasks are temporarily stored in the task dispatcher and the timing of dispatching the stored tasks is controlled without increasing tardiness. Therefore, it is possible to control the timing of dispatching the stored tasks to AGVs for JIT transportation environments. We investigate the relationship between the JIT objective (earliness/tardiness) and the total completion time under various desired delivery time and assignment time intervals. In the computational experiments, the proposed algorithm in Section IV-A and the same computational environment were used. A transportation layout with 39 nodes and 46 arcs (layout 1) and 28 nodes and 35 arcs (layout 2) were utilized. The problem was solved under a dynamic task arrival environment. The number of vehicles and tasks was 10 and 20. The initial node for each vehicle, the pair of pickup and delivery node and task generation time for each task were determined randomly. We obtain the trade-off relationship of the case of which the time from task arrival time to pickup time is 30 and 10. Two scenarios are prepared where the time from task arrival time to pickup time is 30, 10, respectively. The initial node for each vehicle, the pair of pickup and delivery node and task arrival time for each task are the same as prepared. Four cases (Case 1, 2, 3, 4) are prepared for each layout as shown in Figs. 7 and 8. Cases 1 and 2 are the problems for layout 1 and Cases 3 and 4 are the problems for layout 2 with different assignment time intervals. For each case, three instance problems are solved with different tasks. The instance number j is written as Case $i-(j)$ for Case i. The assignment time interval is fixed through a dynamic problem ($I_i = I, i = 1, 2, \ldots$). In each case, the trade-off relationship is derived under assignment time interval $I = 10$ and 30. The time horizon of the dynamic problem is set to $H = 600$.

The JIT objective values (total earliness and tardiness) and the total completion time are obtained by solving those dynamic problems. 21 types of combinations of different weighting factors and three assignment time intervals for each problem are shown in Figs. 7 and 8. The weighting factors γ is set 100. The penalty for decreasing collisions δ is set to 100. The number of repetitions are $\theta_1 = 500$ and $\theta_2 = 30$.

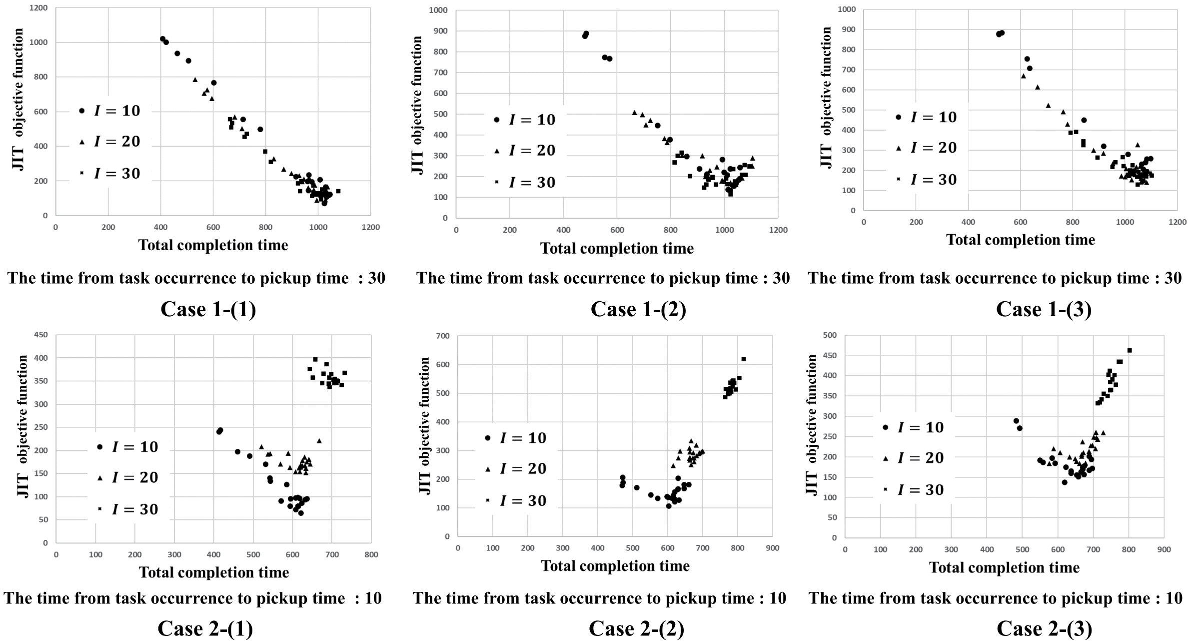

Fig. 7. Computational results of the total completion time and JIT performance under different l_i in layout 1.

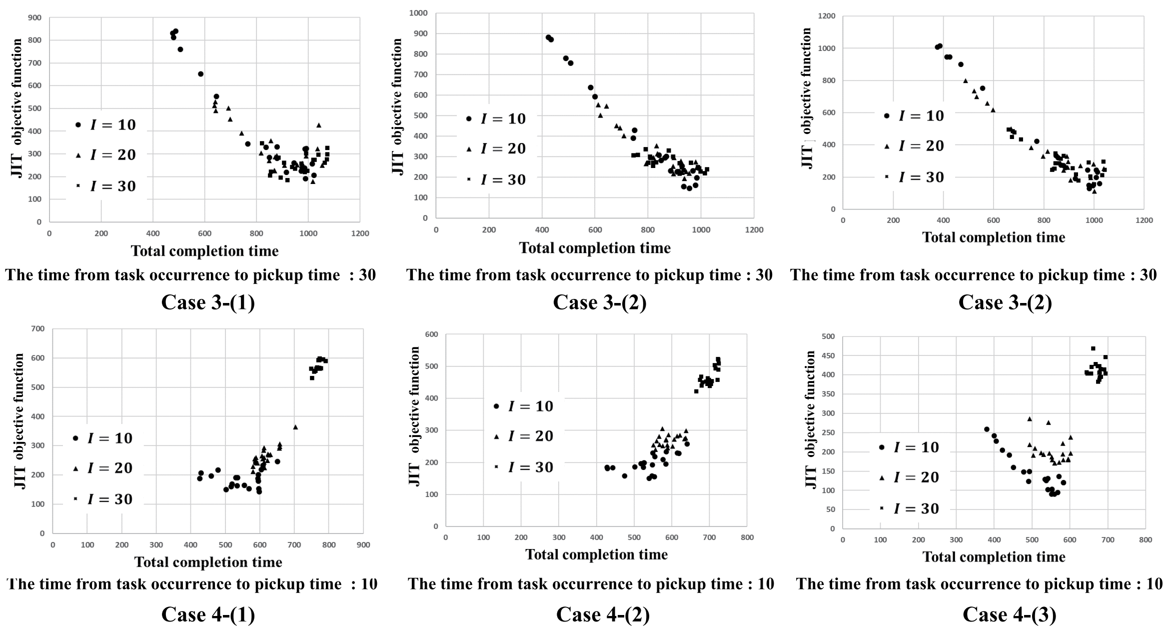

Fig. 8. Computational results of the total completion time and JIT performance under different l_i in layout 2.

From the computational results, the trade-off relationship between the total earliness/tardiness and the total completion time can be obtained especially in Case 1-(1), Case 1-(2), Case 3-(2) and Case 3-(3). However, there are some cases when the trade-off relationship cannot be obtained.

For those cases, the time duration between the task arrival time and the pickup specified time is sufficiently larger than the time required for the shortest path. In this case, it becomes easier to adjust the task completion time.

However, the solutions to improve the total completion time cannot be derived when $I = 30$ compared to the cases when $I = 10$. It is because the time duration between the task arrival time and task assignment time is sufficiently longer when $I = 30$ compared with the cases when $I = 10$. Then, it becomes extremely difficult to improve the total completion time.

In Case 2-(1) compared with Cases 1-(1), and Case 4-(3) with Case 3-(3), the JIT objective value of the solution when $I = 30$ becomes worse than that of the solution which has the same total completion time when $I = 10$. The time required for the shortest path is longer than the time duration between task assignment time and the pickup specified time when $I = 30$.

If $I = 10$ of Case 2-(2) and Case 2-(3), both the JIT objective value and the total completion time are larger when the weighting factor of the JIT performance is higher. It is because some tasks can improve JIT objective whereas there is no enough time for the tasks dispatched later to minimize the JIT objective values.

If $I = 30$ of Case 2-(3) and $I = 30$ of Case 3-(2), the trade-off relationship cannot be obtained. In the case, many tasks are tardy for the specified time at task assignment time and the routing for each vehicle is determined to minimize only the total completion time regardless of the weighting factors. Therefore, the relationship of trade-off cannot be obtained because both the JIT objective and the total completion time are improved such that collisions and congestions can be avoided.

Some solutions in Case 2-(1) have a trade-off relationship. However, in some solutions in Case 2-(1), the trade-off relationship cannot be obtained even when the time duration between task arrival time and the pickup specified time is the same as Case 1-(2). This is because so many tasks are given at the task assignment time and all tasks cannot be completed by a specified time. Then, both the JIT objective and the total completion time cannot be minimized when collisions and congestions cannot be avoided.

The following is a summary of the above results.

- Cases when $I = 30$
  - In Case 1-(1) in which 20 tasks are given during 70 steps, it is easy to obtain the relationship of trade-off.
  - In Case 1-(2) in which 20 tasks are given during 42 steps, both the JIT objective and the total completion time are larger when the weighting factor of the JIT objective value is higher.
  - The solution for minimizing the total completion time cannot be derived when $I = 30$ compared to that when $I = 10$.
  - The best JIT objective value when $I = 30$ is smaller than other cases and the best JIT objective value in the solutions of $I = 1$ is larger than those of others.
- Cases when $I = 10$
  - The best JIT objective value is the smallest when $I = 1$ and the largest when the assignment time interval is 30.
  - The JIT objective value when $I = 30$ is larger than the solution which has the same total completion time when $I = 10$.
  - The trade-off relationship cannot be obtained when $I = 30$ in Case 2-(2).

D. Computational Study on Trade-off Analysis Under Different Situations

1) Degree of Task Arrival Frequency: The frequency of task arrival affects the JIT objectives of the earliness/tardiness and total throughput of the AGV systems. In this section, we investigate the effects of the task arrival frequency on JIT delivery on the JIT objectives and the total completion time. The degree of task arrival frequency (D_{TAF}) defined by (37) is the rate of working time against the completion time of the final task for each vehicle when a dynamic problem is solved to minimize the total traveling time only.

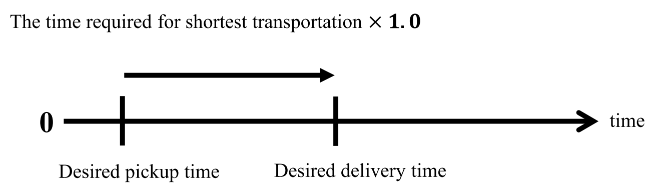

Fig. 9. Relation between the desired time and the duration for the shortest path when D_{idle} is 1.0.

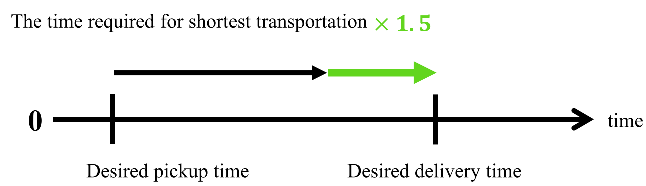

Fig. 10. Relation between the desired time and the duration for the shortest path when D_{idle} is 1.5.

[Notation]

M : set of vehicles

$T_{idle}(m)$: Idle time of vehicle m

$C_{max}(m)$: Completion time of final task for vehicle m

$$D_{TAF} \text{ for vehicle } m = 1 - \frac{\sum_{m\in M} T_{idle}(m)}{\sum_{m\in M} C_{max}(m)} \tag{37}$$

2) Setting of Desired Pickup and Delivery Time According to Degree of Idleness: We define the idle ratio to quantitatively represent the length of the time between task arrival time and the desired delivery time. The degree of idleness (D_{idle}) is defined as the ratio of the time between task arrival time and the desired time against the time required for the shortest transportation. The desired pickup time (P_l^p) and the desired delivery time (P_l^d) satisfy (38) and (39), respectively.

d_{n_1,n_2} : Shortest traveling time from node n_1 to node n_2

g_l : Delivery node of task l

O_l : Arrival time of task l

p : Loading and unloading time

s : starting node for vehicle when task l is dispatched

u_l : Pickup node of task l

$$P_l^p = O_l + d_{s,u_l} \times D_{idle} + p \tag{38}$$
$$P_l^d = P_l^p + d_{u_l,g_l} \times D_{idle} + p \tag{39}$$

The relation between the desired time when D_{idle} is 1.0 and 1.5 and the shortest traveling time is shown in Fig. 9 and 10. It is clear from Fig. 9 that, when D_{idle} is 1.0, the shortest traveling time is the same as the time until the desired time. On the other hand, when D_{idle} is 1.5, the time until the desired time is 1.5 times the shortest traveling time from Fig. 10.

We investigate the relationship between the JIT objectives and the total completion time in various degrees of idleness

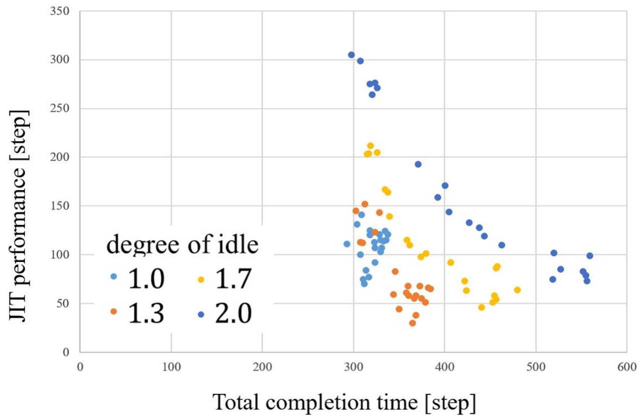

Fig. 11. Computational results when D_{taf} = 50% under different degree of idleness.

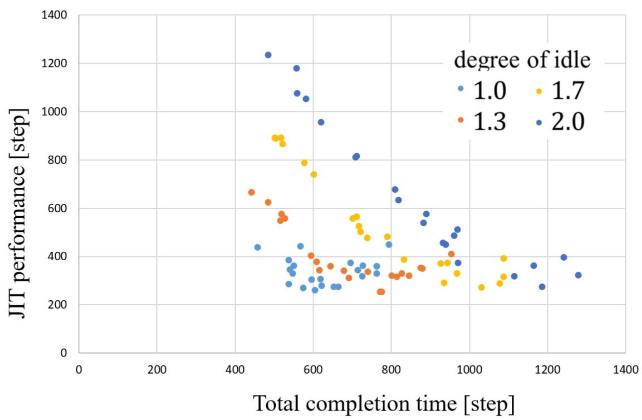

Fig. 12. Computational results when D_{taf} = 90% under different degree of idleness.

and degree of frequency. A transportation layout with 39 nodes and 46 arcs is used under the dynamic task arrival. The number of vehicles is 10 and the number of tasks is 20. The initial node for each vehicle, the pair of pickup and delivery nodes for each task is randomly determined. We obtain the trade-off relationship of the case of which task arrival time is set so that D_{taf} is 50% and 90%, and the desired time is set so that the degree of idleness is 1.0, 1.3, 1.7 and 2.0. The assignment task interval I is set to 1 so that task assignment and routing for each vehicle are determined at every task arrival time. The time horizon of the dynamic problem is set to $H = 600$.

The JIT objectives (earliness/tardiness) and the total completion time are obtained by solving the dynamic problem. 21 types of combinations of different weighting factors and four degrees of idleness for each are shown in Figs. 11 and 12. The weighting factor γ is set to 5000. The penalty for decreasing collisions δ is set to 10000. The number of iterations of the proposed method is determined by the parameters $\theta_1 = 500$ and $\theta_2 = 60$.

If $D_{taf} = 90\%$, the JIT objective could not be improved even when the weighting factor of the JIT objective is higher. It is because the completion time of tasks was so tardy that the traveling of vehicles could not meet the desired time when the degree of idleness is 2.0.

If $D_{taf} = 50\%$ and 90%, the best JIT objective value can be obtained when the degree of idleness is 1.3. It is because the waiting time of vehicles becomes so short that every vehicle can arrive at the destination at the desired time compared with $D_{idle} = 1.3$.

From the computational results, we have the following managerial implications.

- If the time duration between the task arrival time and the desired pickup time is sufficiently large, the JIT transportation is easily realized when the task assignment interval is $I = 1,\ 10,\ 20,\ 30$.
- The JIT objectives of earliness/tardiness can be easily minimized while minimizing the total completion time when the trade-off between the JIT objective and the total completion time can be obtained.
- If the time duration between the task arrival time and the desired pickup time is sufficiently large, the best JIT objective values in the solutions of $I = 30$ tend to be smaller than those of $I = 10$.
- In the $D_{taf} = 50\%$, the JIT objective can be easily improved when D_{idle} is around 1.3.
- In the $D_{taf} = 90\%$, the JIT objective cannot be improved regardless of the number of vehicles/congestions between vehicles.

These results are significant when the proposed method is implemented in the real transport environment. Especially, these results demonstrate the managerial implications when the JIT delivery can be achieved while minimizing the total completion time. Also, the setting of the parameters of D_{taf} and D_{idle} are used for the design of transport systems, warehouse layout [50] and several logistic systems in practice.

VI. Conclusion and Future Work

We have proposed a heuristic solution procedure for the JIT routing and scheduling problems for dynamic transportation. The effectiveness has been confirmed by comparing the performance with Gurobi from computational results. The advantage and usefulness are that the method can be applied to a dynamic environment for real time optimization. The trade-off relationship between the JIT objectives (earliness/tardiness) and the total completion time has been examined under different task arrival frequencies and different desired times. It has been revealed that when the JIT delivery can be achieved while minimizing the total completion time. We also obtained the implications that JIT deliveries cannot be achieved if the time duration between the task assignment and the desired time is too long compared with the traveling time required for the shortest transportation. Our future work is to investigate how the transportation layout [20], [50] affects the JIT objective values and the trade-off relationship between the JIT objective and the total completion time.

Acknowledgment

The authors would like to thank Dr. Kenji Kumagai and Hideki Kaname with Murata Machinery Ltd. For their suggestions and valuable comments to improve the paper.

REFERENCES

[1] T. Adamo, T. Bektaş, G. Ghiani, E. Guerriero, and E. Manni, "Path and speed optimization for conflict-free pickup and delivery under time windows," *Transp. Sci.*, vol. 52, no. 4, pp. 739–755, Aug. 2018.

[2] Ü. Bilge, G. Esenduran, N. Varol, Z. Öztürk, B. Aydın, and A. Alp, "Multi-attribute responsive dispatching strategies for automated guided vehicles," *Int. J. Prod. Econ.*, vol. 100, no. 1, pp. 65–75, Mar. 2006.

[3] N. Boysen, R. de Koster, and F. Weidinger, "Warehousing in the e-commerce era: A survey," *Eur. J. Oper. Res.*, vol. 277, no. 2, pp. 396–411, Sep. 2019.

[4] J. Chen, W. Weng, G. Rong, and S. Fujimura, "Integrating genetic algorithm with time control for just-in-time scheduling problems," in *Proc. IFAC Symp. Inf. Control Problems Manuf.*, vol. 48, no. 3, 2015, pp. 893–897.

[5] H.-W. Cheong and H. Lee, "Requirements of AGV (automated guided vehicle) for SMEs (small and medium-sized enterprises)," *Proc. Comput. Sci.*, vol. 139, pp. 91–94, Jan. 2018.

[6] X. Chen, L. Feng, X. Wang, W. Wu, and R. Hu, "A two-stage congestion-aware routing method for automated guided vehicles in warehouses," in *Proc. IEEE Int. Conf. Netw., Sens. Control (ICNSC)*, Dec. 2021, pp. 1–6, doi: 10.1109/ICNSC52481.2021.9702160.

[7] A. I. Corréa, A. Langevin, and L.-M. Rousseau, "Scheduling and routing of automated guided vehicles: A hybrid approach," *Comput. Oper. Res.*, vol. 34, no. 6, pp. 1688–1707, 2007.

[8] G. Daugherty, S. Reveliotis, and G. Mohler, "Optimized multiagent routing for a class of guidepath-based transport systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 363–381, Jan. 2019.

[9] G. Desaulniers, A. Langevin, D. Riopel, and B. Villeneuve, "Dispatching and conflict-free routing of automated guided vehicles: An exact approach," *Int. J. Flexible Manuf. Syst.*, vol. 15, no. 4, pp. 309–331, 2003.

[10] N. Du, H. Hu, and M. Zhou, "Robust deadlock avoidance and control of automated manufacturing systems with assembly operations using Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 4, pp. 1961–1975, Oct. 2020.

[11] H. Fazlollahtabar, M. Saidi-Mehrabad, and J. Balakrishnan, "Mathematical optimization for earliness/tardiness minimization in a multiple automated guided vehicle manufacturing system via integrated heuristic algorithms," *Robot. Auto. Syst.*, vol. 72, pp. 131–138, Oct. 2015.

[12] Y. Hu, L. Dong, and L. Xu, "Multi-AGV dispatching and routing problem based on a three-stage decomposition method," *Math. Biosci. Eng.*, vol. 17, no. 5, pp. 5150–5172, 2020.

[13] I. Hwang and Y. J. Jang, "Q(λ) learning-based dynamic route guidance algorithm for overhead hoist transport systems in semiconductor fabs," *Int. J. Prod. Res.*, vol. 58, no. 4, pp. 1199–1221, Feb. 2020.

[14] S. M. Jeon, K. H. Kim, and H. Kopfer, "Routing automated guided vehicles in container terminals through the Q-learning technique," *Logistics Res.*, vol. 3, no. 1, pp. 19–27, Apr. 2011.

[15] Z. Liu and T. Nishi, "Analyzing just-in-time purchasing strategy in supply chains using an evolutionary game approach," *J. Adv. Mech. Design, Syst., Manuf.*, vol. 14, no. 5, pp. 1–18, 2020.

[16] J. C. Luo, Z. Q. Liu, and M. C. Zhou, "A Petri net based deadlock avoidance policy for flexible manufacturing systems with assembly operations and multiple resource acquisition," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3379–3387, Jun. 2018.

[17] X. Lyu, Y. Song, C. He, Q. Lei, and W. Guo, "Approach to integrated scheduling problems considering optimal number of automated guided vehicles and conflict-free routing in flexible manufacturing systems," *IEEE Access*, vol. 7, pp. 74909–74924, 2019.

[18] M. Mahmoudi and X. Zhou, "Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations," *Transp. Res. B, Methodol.*, vol. 89, pp. 19–42, Jul. 2016.

[19] T. Nishi, M. Ando, and M. Konishi, "Distributed route planning for multiple mobile robots using an augmented Lagrangian decomposition and coordination technique," *IEEE Trans. Robot.*, vol. 21, no. 6, pp. 1191–1200, Dec. 2005.

[20] T. Nishi, S. Akiyama, T. Higashi, and K. Kumagai, "Cell-based local search heuristics for guide path design of automated guided vehicle systems with dynamic multicommodity flow," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 2, pp. 966–980, Apr. 2020.

[21] T. Nishi, Y. Hiranaka, and I. E. Grossmann, "A bilevel decomposition algorithm for simultaneous production scheduling and conflict-free routing for automated guided vehicles," *Comput. Oper. Res.*, vol. 38, no. 5, pp. 876–888, 2011.

[22] T. Nishi and R. Maeno, "Petri net decomposition approach to optimization of route planning problems for AGV systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 523–537, Jul. 2010.

[23] T. Nishi, S. Morinaka, and M. Konishi, "A distributed routing method for AGVs under motion delay disturbance," *Robot. Comput.-Integr. Manuf.*, vol. 23, no. 5, pp. 517–532, 2007.

[24] T. Nishi, K. Shimatani, and M. Inuiguchi, "Decomposition of Petri nets and Lagrangian relaxation for solving routing problems for AGVs," *Eur. J. Oper. Res.*, vol. 47, no. 14, pp. 3957–3977, 2009.

[25] T. Nishi and Y. Tanaka, "Petri net decomposition approach for dispatching and conflict-free routing of bidirectional automated guided vehicle systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 5, pp. 1230–1243, Sep. 2012.

[26] K. Nishida, T. Nishi, H. Kaname, K. Kumagai, and T. Higashi, "Just-in-time routing and scheduling for multiple automated guided vehicles," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2019, pp. 841–846.

[27] Y. Peng, T. Zeng, Y. Han, and B. Xia, "Scheduling just-in-time transport vehicles to feed parts for mixed model assembly lines," *Discrete Dyn. Nature Soc.*, vol. 2020, pp. 1–13, Apr. 2020.

[28] H. F. Rahman and I. Nielsen, "Scheduling automated transport vehicles for material distribution systems," *Appl. Soft Comput.*, vol. 82, Sep. 2019, Art. no. 105552.

[29] K.-B. Reith, P. Boden, M. Daumler, S. Rank, T. Schmidt, and R. Hupfer, "Evaluating automated guided vehicle system characteristics in semiconductor fab automated material handling systems," in *Proc. 30th Annu. SEMI Adv. Semiconductor Manuf. Conf. (ASMC)*, May 2019, pp. 1–6.

[30] S. Riazi and B. Lennartson, "Using CP/SMT solvers for scheduling and routing of AGVs," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 1, pp. 219–229, Aug. 2021.

[31] M. De Ryck, M. Versteyhe, and F. Debrouwere, "Automated guided vehicle systems, state-of-the-art control algorithms and techniques," *J. Manuf. Syst.*, vol. 54, pp. 152–173, Jan. 2020.

[32] S. Shah, I. Logiotatopouloh, and S. Menon, "Industry 4.0 and autonomous transportation: The impacts on supply chain management," *J. Intell. Transp. Syst.*, vol. 4, pp. 45–50, Sep. 2019.

[33] S. Solichudin, T. Aris, and R. Munawar, "Conflict-free dynamic route multi-AGV using Dijkstra Floyd–Warshall hybrid algorithm with time windows," *Int. J. Electr. Comput. Eng.*, vol. 10, pp. 3596–3604, Aug. 2020.

[34] N. Smolic-Rocak, S. Bogdan, Z. Kovacic, and T. Petrovic, "Time windows based dynamic routing in multi-AGV systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 1, pp. 151–155, Jan. 2010.

[35] Y. Tanaka, T. Nishi, and M. Inuiguchi, "Dynamic optimization of simultaneous dispatching and conflict-free routing for automated guided vehicles-Petri net decomposition approach," *J. Adv. Mech. Des., Syst., Manuf.*, vol. 4, no. 3, pp. 701–715, 2010.

[36] U. A. Umar, M. K. A. Ariffin, N. Ismail, and S. H. Tang, "Hybrid multiobjective genetic algorithms for integrated dynamic scheduling and routing of jobs and automated-guided vehicle (AGV) in flexible manufacturing systems (FMS) environment," *Int. J. Adv. Manuf. Technol.*, vol. 81, nos. 9–12, pp. 2123–2141, 2015.

[37] B. S. Vaidyanathan, J. O. Matson, D. M. Miller, and J. E. Matson, "A capacitated vehicle routing problem for just-in-time delivery," *IIE Trans.*, vol. 31, no. 11, pp. 1083–1092, Nov. 1999.

[38] I. F. A. Vis, "Survey of research in the design and control of automated guided vehicle systems," *Eur. J. Oper. Res.*, vol. 170, no. 3, pp. 677–709, May 2006.

[39] Z. Xing, X. Chen, X. Wang, and W. Wu, "Collision and deadlock avoidance in multi-robot systems," *IEEE/CAA J. Automat. Sinica*, vol. 9, no. 7, pp. 1327–1330, Mar. 2022.

[40] K. Xing, M. C. Zhou, H. Liu, and F. Tian, "Optimal Petri-net-based polynomial-complexity deadlock-avoidance policies for automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 1, pp. 188–199, Jan. 2009.

[41] L. Wang and J. Lu, "A memetic algorithm with competition for the capacitated green vehicle routing problem," *IEEE/CAA J. Automat. Sinica*, vol. 6, no. 2, pp. 516–526, Feb. 2019.

[42] J. Wang, Y. Sun, Z. Zhang, and S. Gao, "Solving multitrip pickup and delivery problem with time windows and manpower planning using multiobjective algorithms," *IEEE/CAA J. Automat. Sinica*, vol. 7, no. 4, pp. 1134–1153, Jul. 2020.

[43] N. Wu and M. Zhou, "Shortest routing of bidirectional automated guided vehicles avoiding deadlock and blocking," *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 1, pp. 63–72, Feb. 2007.

[44] N. Wu and M. Zhou, "Modeling and deadlock control of automated guided vehicle systems," *IEEE/ASME Trans. Mechatronics*, vol. 9, no. 1, pp. 50–57, Mar. 2004.

[45] N. Wu and W. Zeng, "Deadlock avoidance in an automated guidance vehicle system using a coloured Petri net model," *Int. J. Prod. Res.*, vol. 40, no. 1, pp. 223–238, Jan. 2002.

[46] J. Xin, L. Wei, D. Wang, and H. Xuan, "Receding horizon path planning of automated guided vehicles using a time-space network model," *Optim. Control Appl. Methods*, vol. 41, no. 6, pp. 1889–1903, Nov. 2020.

[47] G. Yi, Z. Feng, T. Mei, P. Li, W. Jin, and S. Chen, "Multi-AGVs path planning based on improved ant colony algorithm," *J. Supercomput.*, vol. 75, no. 9, pp. 5898–5913, Sep. 2019.

[48] J.-W. Yoo, E.-S. Sim, C. Cao, and J.-W. Park, "An algorithm for deadlock avoidance in an AGV system," *Int. J. Adv. Manuf. Technol.*, vol. 26, nos. 5–6, pp. 659–668, Sep. 2005.

[49] M. Zhang, R. Batta, and R. Nagi, "Modeling of workflow congestion and optimization of flow routing in a manufacturing/warehouse facility," *Manage. Sci.*, vol. 55, no. 2, pp. 267–280, 2009.

[50] G. Zhang, T. Nishi, S. D. O. Turner, K. Oga, and X. Li, "An integrated strategy for a production planning and warehouse layout problem: Modeling and solution approaches," *Omega*, vol. 68, no. 1, pp. 85–94, 2017.

[51] Z. Zhang, Q. Guo, J. Chen, and P. Yuan, "Collision-free route planning for multiple AGVs in an automated warehouse based on collision classification," *IEEE Access*, vol. 6, pp. 26022–26035, 2018.

[52] W.-Q. Zou, Q.-K. Pan, T. Meng, L. Gao, and Y.-L. Wang, "An effective discrete artificial bee colony algorithm for multi-AGVs dispatching problem in a matrix manufacturing workshop," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113675.

Kosei Nishida received the B.S. and M.S. degrees from the Division of Mathematical Science and Social Systems, Graduate School of Engineering Science, Osaka University, Japan, in 2019 and 2021, respectively. Since 2021, he has been with Hankyu Hanshin Holdings Inc., Japan. He was previously engaged in the study on the just-in-time delivery of AGV systems.

Tatsushi Nishi (Senior Member, IEEE) received the Ph.D. degree in process systems engineering from Kyoto University, Kyoto, Japan, in 2001. Since 2001, he has been a Research Associate with the Department of Electrical and Electronic Engineering, Okayama University, Okayama, Japan. From 2006 to 2020, he was an Associate Professor with the Department of Mathematical Science for Social Systems, Graduate School of Engineering Science, Osaka University, Osaka, Japan. He is currently a Full Professor with the Department of Intelligent Mechanical Systems Engineering, Faculty of Natural Science and Technology, Okayama University. He has authored or coauthored over 200 technical publications. His research interests include discrete optimization, discrete event systems, production scheduling, decentralized optimization algorithms, supply chain optimization, multiple mobile robots control, and manufacturing systems. He received several awards, including the National Instruments Corporation Young Investigator Award from the 2006 International Symposium on Flexible Automation; the 2010 Best Paper Award from the Society of Instrument and Control Engineers; the 2011 Best Paper Award from the Scheduling Society of Japan; the 2013 Best Paper Award from the Institute of Systems, Control and Information Engineers; and the Outstanding Paper Awards at the IEEE International Conference on Industrial Engineering and Engineering Management in 2014, 2015, and 2019.