2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)
Bari, Italy. October 6-9, 2019

Just-in-Time Routing and Scheduling for Multiple Automated Guided Vehicles

Kousei Nishida[1], Tatsushi Nishi[1], Hideki Kaname[2], Kenji Kumagai[2], Toshimitsu Higashi[2]

***Abstract*— For multiple Automated Guided Vehicles (AGVs) system, it is required to realize just-in-time delivery to minimize the earliness and tardiness of the specified delivery time not only to minimize the total traveling time. In this paper, we propose a heuristic algorithm for just-in-time and conflict-free routing problem of AGVs. The proposed method can derive a near-optimal task assignment and routing and scheduling of multiple AGVs to improve both the JIT performance and the total completion time. In the proposed method, a strategy to avoid collisions ensuring feasibility is proposed. The effectiveness of the proposed method is demonstrated by comparing the performance with CPLEX from computational results. The results show that the proposed method can obtain a better solution which is close to an optimal solution with a significantly shorter computational time than that derived from CPLEX.**

I. Introduction

With the spread of smart factories, automation of transportation and product delivery is progressed in flexible manufacturing systems to cope with the diversification of customer needs. Multiple AGVs are getting more and more utilized in various fields [1]. In the fourth industrial revolution era as stated in Industrie 4.0, many companies are focusing on the development of AGV transport systems in their factories to develop smarter factories and to enhance manufacturing competitiveness due to the computerization and intellectualization of products and services [2].

For the recent requests of online shopping, AGV transport systems are widely used not only transportation of wafers in a conventional semiconductor factory, but also the transportation of raw materials, parts and products in production plant, storage and shipping of products in distribution centers. In the future, when automated cars using AI and optimization are introduced, it is expected that the use of automated mobile robots such as drones and AGVs will become widespread not only in factory automation but also in public transportation and traffic systems.

The control of AGVs generally consists of two phases: the one is the routing and scheduling and the second one is the real time motion control. Previous researches on routing in a guide path layout have been concentrated to minimize the total completion time [3], [6], [7] and [8]. However, in recent years, to cope with the diversification of customer needs, not only to minimize the total transportation time, but also to minimize the earliness and tardiness to achieve just-in-time transportation is required. In many real factories with multiple production processes, the shortest time transportation may sometimes cause 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.

Just-in-time concept is often used in production management and service. In order to reduce inventory costs by producing only if necessary, and to increase customer satisfaction by supplying products just-in-time, a solution method for the just-in-time flexible job shop scheduling problem has been proposed [4]. Fazlollahtabar et al. (2015) proposed a routing method that considers the transportation deadline without AGV collisions [5]. However, the minimization of the total completion time is not considered in their study.

On the other hand, the task assignment problem is closely related to the routing of all AGVs. In the task assignment problem to minimize the transportation time, the nearest neighborhood method that finds the lower bound value of the time taken to arrive at the initial node of the task newly generated, and assigns the task to the AGV with the smallest value is used [6]. The simultaneous optimization problem of the production scheduling and routing, and task assignment, is formulated as a mixed integer programming problem and a Lagrangian relaxation method and cut generation algorithm has been proposed [7].

The objective of our paper is to propose a heuristic algorithm to optimize task assignment and conflict-free routing to improve the JIT performance and the total completion time. The effectiveness of the proposed method is demonstrated by comparing the performance with CPLEX from computational results. We examine the trade-off relationship in the bi-objective optimization of the JIT performance and the total completion time.

II. Problem Definition

A. Transportation system

A guide path in transportation system is represented by the set of nodes and arcs as shown in Fig. 1. Each node represents a place where each vehicle can stop or turn. Each arc represents a lane which AGV can travel. It is assumed that there are multiple vehicles in the system and a set of tasks is given in advance. Each AGV has its starting position. Each task is given by a pick node up and a delivery node. Each task is assigned to one of AGVs. Each AGV can transport only one task at the same time and completes the assigned tasks without collision with other AGVs. In order to optimize the AGV system, we assume that the AGV

[1]Mathematical Science for Social System, Graduate School of Engineering Science Osaka University, 1-3 Machikaneyama-cho, Toyonaka City, Osaka, Japan nishida@inulab.sys.es.osaka-u.ac.jp, nishi@sys.es.osaka-u.ac.jp

[2]Murata Machinery, LTD., 146 Takedamukaishiro-cho, Fushimiku, Kyoto, Japan

978-1-7281-4569-3/19/$31.00 ©2019 IEEE

motion is discrete, and the time of traveling to adjacent node is one unit time. Under the assumption, at time step t ($t = 0, 1, 2, ...$), all AGVs can be on each node and not on arcs. Two or more AGVs cannot travel on the same node at the same time (restriction of no collision). Two or more AGVs cannot travel on the same arc at the same time (restriction of no passing). AGV can travel to the adjacent node only if there is no other AGV in that node (restriction of no simultaneous transition).

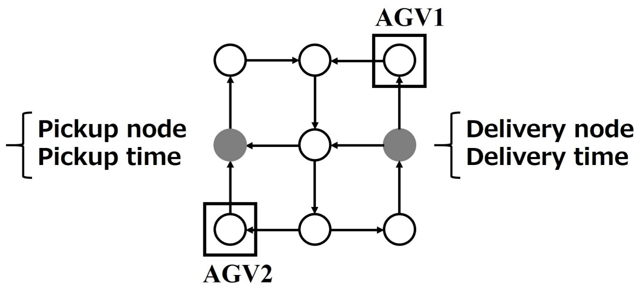

Fig. 1. Just-in-time routing problem of AGVs

B. Just-in-time routing and scheduling problem

The definition of transportation time is illustrated in Fig. 2. Once each task is assigned, each AGV travels from its initial node to the pickup node, then the AGV travels to the delivery node. The duration time between the task assignment time and the completion time of pickup and delivery is defined as the completion time.

The sum of the absolute value of the difference between the delivery completion time and the specified delivery time and the absolute value of the difference between the pickup completion time and the specified pickup time is called the total earliness and tardiness. JIT performance is the sum of earliness and tardiness. To transport just at the specified times is called the JIT transportation. The just-in-time routing and scheduling problem is defined as the problem of determining the task assignment and conflict-free routing of multiple AGVs to minimize the weighted sum of the total earliness and tardiness and the total completion time under the conflict-free constraints for AGVs.

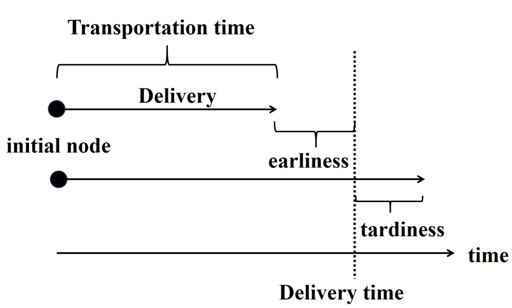

Fig. 2. Just-in-time routing problem of AGVs

C. Integer Programming Problem Formulation

The simultaneous task assignment and routing problem for JIT transportation (P) is formulated as an integer programming problem. In [6], the constraints on AGVs traveling ((2)-(6), (8) and (9)) are considered. However, earliness and tardiness in the objective function and the constraints on task assignment and pickup and delivery time are not considered.

[Sets]
M : set of vehicles
N : set of nodes
A_i : set of adjacent nodes from node i, which includes node i
T : set of time period during planning horizon
L : set of tasks

[Decision variables]
$x^m_{i,j,t}$: 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^l_t : 1 if delivery of task l is not completed at time t, and 0 otherwise
K^l_t : 1 if pickup of task l is not completed at time t, and 0 otherwise
y^m_l : 1 if task l is assigned to AGV 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 AGV m at the time 0
$u_l \in N$: pickup node of task l
$g_l \in N$: delivery node of task l
$J_{u_l} \in T$: pickup time of task l
$J_{g_l} \in T$: delivery time of task l
α : weighting factor for JIT performance
β : weighting factor for the complete time

[Objective function and constraints]

$$(P) \quad \min \quad J = \alpha \sum_{l \in L} (f_l + e_l) + \beta \sum_{l \in L} \sum_{t \in T} D^l_t \quad (1)$$

$$\sum_{j \in A_i} x^m_{i,j,t} \leq 1 \qquad (i \in N,\ m \in M,\ t \in T) \quad (2)$$

$$\sum_{i \in N} \sum_{j \in A_i} x^m_{i,j,t} = 1 \qquad (m \in M,\ t \in T) \quad (3)$$

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

The objective function (1) consists of the weighted sum of the total earliness and tardiness and the total completion time where α and β are the weighting factors. Equation (2) states that each vehicle can only travel to one of the adjacent nodes which is connected with node i. Equation (3) ensures that each vehicle chooses only one arc to travel or node to stop. Equation (4) 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.

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

$$\sum_{m\in M} (x^m_{i,j,t} + x^m_{j,i,t}) \le 1 \quad (i,\, j\in N,\, i\ne j,\, t\in T) \tag{6}$$

$$\sum_{j\in A_i} x^{m_1}_{j,i,t} + \sum_{m_2\in M\setminus\{m_1\}}\sum_{k\in A_i} x^{m_2}_{i,k,t} \le 1$$
$$(i\in N,\, m_1\in M,\, t\in T) \tag{7}$$

Equation (5) ensures that only one AGV can travel to node i at the same time. Equation (6) means that only one AGV can travel between node i and node j at the same time. Equation (7) represents that AGVs which travel to node i and from node i at the same time cannot be on the same node simultaneously.

$$\sum_{j\in A_{s_m}} x^m_{s_m,j,0} = 1 \qquad (m\in M) \tag{8}$$

$$\sum_{i\in N\setminus\{s_m\}}\sum_{j\in A_i} x^m_{i,j,0} = 0 \qquad (m\in M) \tag{9}$$

Equations (8) and (9) indicate that AGV m can travel from only the initial node s_m at the time 0.

$$\sum_{m\in M} y^m_l = 1 \qquad (l\in L) \tag{10}$$

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

$$K^l_t - \sum_{m\in M}(y^m_l x^m_{u_l,u_l,t}) \le K^l_{t+1} \qquad (l\in L,\, t\in T) \tag{12}$$

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

$$D^l_t - \sum_{m\in M}(y^m_l x^m_{g_l,g_l,t}) \le D^l_{t+1} \qquad (l\in L,\, t\in T) \tag{14}$$

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

$$K^l_0 = 1,\, D^l_0 = 1 \qquad (l\in L) \tag{16}$$

$$K^l_{t+1} \le K^l_t,\, D^l_{t+1} \le D^l_t,\, K^l_t \le D^l_t \quad (l\in L,\, t\in T) \tag{17}$$

Equations (10) and (11) indicate that each task is exactly assigned to one AGV and one task, respectively. Equations (12) and (13) state that the binary variable K^l_t takes a value of until vehicle $m\in M$ completes the pickup for task l, and 0 when the vehicle complete pickup. Equations (14) and (15) 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 complete delivery. Equation (16) ensure that variables K, D take the value of 1 at $t=0$. Equation (17) indicates that once K^l_t and D^l_t take a value of 0, 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.

$$-f_l \le \sum_{t\in T} D^l_t - J_{g_l} \le f_l \qquad (l\in L) \tag{18}$$

$$-e_l \le \sum_{t\in T} K^l_t - J_{u_l} \le e_l \qquad (l\in L) \tag{19}$$

Equations (18) and (19) indicate that the absolute value of the difference between the delivery completion time and the delivery time is f_l, and the absolute value of the difference between the pickup completion time and the pickup time is e_l.

$$x^m_{i,j,t} \in \{0,1\} \qquad (i,j\in N,\, m\in M,\, t\in T) \tag{20}$$

$$D^l_t,\, K^l_t \in \{0,1\} \qquad (l\in L,\, t\in T) \tag{21}$$

$$y^m_l \in \{0,1\} \qquad (l\in L,\, m\in M) \tag{22}$$

$$f_l,\, e_l \in \mathbb{N}^+_0 \qquad (l\in L) \tag{23}$$

Equations (20)-(23) are the constraints for decision variables where $\mathbb{N}^+_0$ is defined as the set of non-negative integers.

III. Heuristic Algorithm for Just-in-Time Routing Problem

A. Outline of the proposed algorithm

The integer programming problem can be solved using a general-purpose solver such as CPLEX if the problem size is sufficiently small. However, if the number of AGVs and tasks are increased, it takes extremely much computational time to solve, which is not practical considering the actual transport system. Therefore, we propose a heuristic algorithm to obtain a solution close to the optimal solution in a short time. In the proposed method, the routing of each AGV for JIT transportation is determined by a shortest path algorithm with a time-space network. Each state in the network has a node that represents where the AGV is located and the current time. At this time, a penalty to avoid collision between AGVs is added to minimize collisions with other AGVs. If the routing of each AGVs is infeasible, a conflict-free routing algorithm is conducted. For task assignment, after the initial assignment is determined, another task assignment is searched for by changing the assignment locally according to the objective value of each task. The outline of the heuristic algorithm is as follows.

Input : the parameters of the problem: the number of iterations θ_1 and θ_2, the weighting factors α, β and γ

Output : task assignment, each AGV routing, the value of the objective function

STEP 1 Initial assignment considering JIT performance.

STEP 2 Randomly select an AGV that does not have a routing plan.

STEP 3 JIT routing is conducted for the selected AGV.

STEP 4 If the routing for all AGVs is not determined, go to STEP 2.

STEP 5 If there are conflicts between AGVs, the conflict-free routing algorithm is conducted.

STEP 6 The solution derived at STEP 5 is updated if it is the best solution in the previous solutions. If the solution has not been updated a fixed number of times (θ_1), then go to STEP 7; otherwise go to STEP 2.

STEP 7 The solution derived at STEP 6 is updated if it is the best solution. If the solution has not been updated a fixed number of times (θ_2), then the algorithm is completed.

STEP 8 Based on the evaluation of the current task assignment, change task assignment again, and go to STEP 2.

Fig. 3 shows the flowchart of the proposed algorithm.

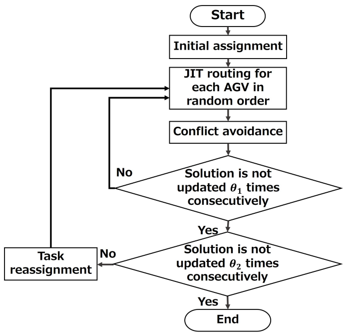

Fig. 3. The flowchart of the proposed algorithm

B. JIT routing method

We define the set of states $\mathcal{P}$ wherein each state represents that an AGV exists at node n at time t as shown in (24).

$$P = \{\sigma | \sigma = (t,\, n),\, t \in T,\, n \in N\} \tag{24}$$

$$d_{\sigma_a \sigma_b} = \begin{cases} \alpha \max(0, T_{set} - t_{\sigma_b}) \\ \quad +\gamma \max(0, t_{\sigma_b} - T_{set}) \quad (n_{\sigma_a} \neq n_{\sigma_b}) \\ \beta(t_{\sigma_b} - t_0) + \gamma \max(0, t_{\sigma_b} - T_{set}) \\ \qquad\qquad\qquad\qquad (n_{\sigma_a} = n_{\sigma_b}) \end{cases} \tag{25}$$

Equation (25) 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 AGV should arrive its delivery node to complete task at delivery time.

The case when $n_{\sigma_a} \neq n_{\sigma_b}$ of (24) 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 $n_{\sigma_a} = n_{\sigma_b}$ of (24) 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 α increases, the cost to travel other node increases as time step is increased, and JIT performance is improved. As β increases, the cost to stop the same node increases as time step is increased, and the completion time is improved. 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 routing of each AGV considering JIT performance without considering the collision between AGVs.

C. Conflict-free routing algorithm

To determine the routing for each AGV may be infeasible because it does not consider collisions with other AGVs. Therefore, it is necessary to use a conflict avoidance strategy to generate a feasible solution. Tanaka et al. (2010) developed a deadlock and blocking avoidance algorithm [8]. An initial route of each AGV is compared sequentially from time step 0 to check whether there are conflicts or not. If there are conflicts, one AGV stops and delays the 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 in subsection A of section Ⅲ, we use the algorithm to guarantee the feasibility.

D. 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. Therefore, as the number of collisions on the initial route of each AGV increases, the difference from the pickup time and delivery time may increase. For this reason, a penalty is added in the transition cost to drive a conflict-free shortest path algorithm so that the number of collisions is reduced.

$$C(m,t,n) = \sum_{m' \in M'} \delta(x_{i,n,t-1}^{m'} + x_{i,n,t}^{m'} + x_{i,n,t+1}^{m'}) \qquad (i \in A_n,\, m' \in M') \tag{26}$$

Equation (26) is the penalty cost of transition to $\sigma = (t, n)$ of AGV m. M' is defined as the set of AGVs which has already determined the routing before AGV m. Fig. 4 shows a time-space network when a penalty is added to the state according to the routing of AGV which has already determined the routing previously, then the state with a penalty cost is avoided. In this example, the previous shortest path is $\sigma = (4, 2)$ and $(5, 3)$, and the penalized states are $\sigma = (4, 2)$, $(5, 3)$, $(3, 2)$, $(5, 2)$, $(4, 3)$ and $(6, 3)$.

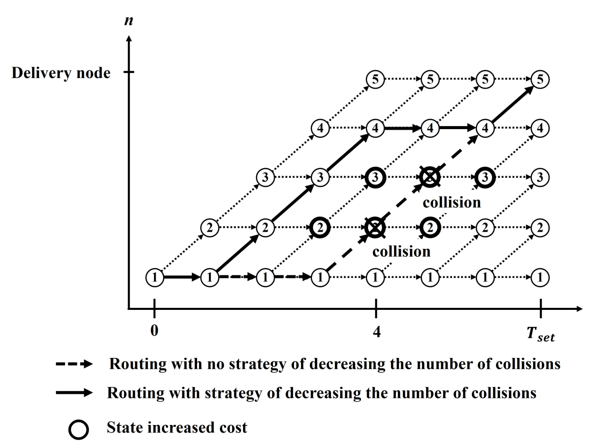

Fig. 4. The strategy of decreasing the number of collisions on the initial routing

E. Task assignment method

The task assignment is related to the overall objective value, and it is assigned such that the completion time is not too early or too late from the specified time. However, it is difficult to determine the task assignment considering conflict-free routing in a short computational time. Therefore, the initial task assignment is determined based on the time up to the specified time and the time it takes to transport to the destination in the shortest time. Each task is assigned to one of AGVs which completes in time for the specified time assuming that there no conflicts and it transports in the shortest time. There may be a better task assignment than the initial assignment. Therefore, the best task assignment is searched by changing the task assignment via neighborhood search. Then, we insert it into the first task of each AGV. We assign it to the first order in which the AGV with the best the objection value completes the task. The search is repeated until the solution is not updated a fixed number of times (θ_2).

F. Computational complexity

When an initial assignment is determined, Dijkstra's algorithm is used for the shortest path. The computational complexity of obtaining the matrix is $\mathcal{O}(|M||L||N|^2)$ and that of the row of the matrix recalculation is $\mathcal{O}(|L|^2|N|^2)$. Therefore, the computational complexity of the initial task assignment is $\mathcal{O}(|M||L||N|^2) + \mathcal{O}(|L|^2|N|^2)$. The number of the states is $|N||T|$ and the computational complexity of JIT routing of one AGV is $\mathcal{O}(|N|^2|T|^2)$. Therefore, the computational complexity of JIT routing of all AGVs is $\mathcal{O}(|M||N|^2|T|^2)$. Due to the JIT routing of all AGVs is repeated $|M|$ times, the computational complexity of task reassignment is $\mathcal{O}(|M|^2|N|^2|T|^2)$.

IV. Computational results

We investigate the performance of the proposed method by comparing with the results of solving the model (P) by using a general-purpose solver (CPLEX). The program was implemented with Microsoft Visual C++ 2010 Express Edition. A general-purpose solver IBM ILOG CPLEX12.6.0.0 was used. An Intel(R) Core(TM) i7-6700 3.40GHz with 8.0GB memory was used for computations. The computation is stopped and the best solution is derived when the computation time of CPLEX reaches 3600 seconds. A transportation layout with 31 nodes and 38 arcs in Fig. 5 is used.

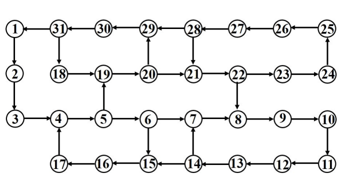

Fig. 5. Small scale transportation system layout

We first compare the performance of the proposed method with CPLEX in the case when the task assignment is fixed to verify the performance of the proposed routing method. The weighting factors α, β and γ are set to 1, 0 and 100. The penalty for decreasing collisions δ is set to 100. The number of repetitions $\theta_1 = 100$ and $\theta_2 = 0$ because the task assigned is fixed. The average of the solutions and the computational time for 10 cases are shown in Table I.

TABLE I
Comparison of the proposed method and CPLEX with fixed task assignment

	Proposed method		CPLEX	
AGVs	Obj.	Time(s)	Obj.	Time(s)
5	10.5	0.2	8.3	8.0
7	22.5	0.2	17.6	116.2
9	53.4	0.32	38.7	2270.7

It is confirmed that the proposed method can derive feasible solutions with a significantly shorter time. In the proposed method, collisions are avoided after routing of each AGV, so there is no need to consider collisions with other AGVs in routing.

Then, we compare the performance of the proposed method with that of CPLEX by setting $\theta_2 = 50$. The average of the objective value and the computational time for 10 cases are shown in Table II.

TABLE II
Computational results of the proposed method ($\alpha = 1, \beta = 0$)

	Proposed method		CPLEX	
AGVs	Obj.	Time(s)	Obj.	Time(s)
5	6.6	23.5	4.6	489.6
7	13.8	61.3	7.3	2126.3
9	25.6	142.8	46.1	3565.8

The proposed method can derive feasible solutions with a significantly shorter time and the value of the objective function is close those of CPLEX. Furthermore, in the case of 9 AGVs, the average of objective function values is better in the proposed method than those derived by CPLEX.

CPLEX 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 AGVs is increased. However, the solutions of the proposed method are not better compared to the other cases in 1 case for 5 AGVs and in 3 cases for 7 AGVs. The reason is that the solution is trapped into a bad local optimum solution.

To investigate the performance of the bi-objective problems, the weighting factors α, β and γ are set to 1, 1 and 100. θ_1 and θ_2 are set to 100 and 50. The average of the objective function value and the computational time for 10 cases are shown in Table III. The case where there is no value in the table indicates that a feasible solution could not be obtained within 3600 seconds. In that case, the average is calculated only for the cases which the feasible solution is obtained by CPLEX.

It is also confirmed that the proposed method can obtain feasible solutions with a significantly shorter time than CPLEX in all cases and the solution is very close those

TABLE III
COMPUTATIONAL RESULTS OF THE PROPOSED METHOD ($\alpha = 1, \beta = 1$)

	Proposed method		CPLEX	
AGVs	Obj.	Time(s)	Obj.	Time(s)
5	123.6	18.8	117.8	438.2
7	176.6	58.4	166.5	2512.1
9	206.0	132.6	208.0	3037.4

of CPLEX. Furthermore, even in the case of 9 AGVs, the average of objective function values is better in the proposed method than in CPLEX. However, for 5 AGVs and minimization of 7 AGVs, the difference with the solution of CPLEX is larger than in the case of $\alpha = 1$, $\beta = 0$. The reason is that routing of each AGVs is more difficult to determine routing that minimizes the two objective functions than to improve only JIT performance. CPLEX could not find even a feasible solution in Case 5 to Case 10 for 9 AGVs.

V. TRADE-OFF BETWEEN JIT PERFORMANCE AND TOTAL COMPLETION TIME

In order to improve the JIT performance, the total completion time is larger because AGVs complete tasks to meet the specified time. However, if the difference in the completion time from the specified time is allowed, it is possible to reduce the total completion time instead of deteriorating the JIT performance. We example by numerical experiments whether various solutions can be obtained from the heuristic method by changing the weights α, β and the transition costs in the JIT routing method. The Case 1 and Case 9 for 5 AGVs and 5 tasks with various weights are solved and the results are shown in Figs. 6 and 7. By changing

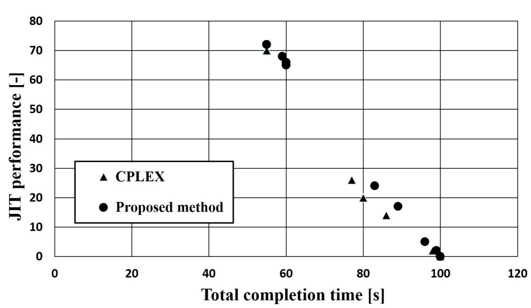

Fig. 6. Trade-off between JIT performance and total completion time (case1)

the weights, a variety of solutions could be obtained. It was also confirmed that the solutions obtained by the proposed method which are close to the Pareto optimal solution derived from CPLEX. In this study, we minimize the weighted sum of the JIT performance and the total completion time. Even the proposed method is based on the heuristic method, the near-optimal Pareto optimal solution can be obtained by changing the weighting factors in the objective function. This allows the user of the transport system to change the quality of routing depending on how much emphasis is placed on the completion time and the JIT performance of earliness/tardiness.

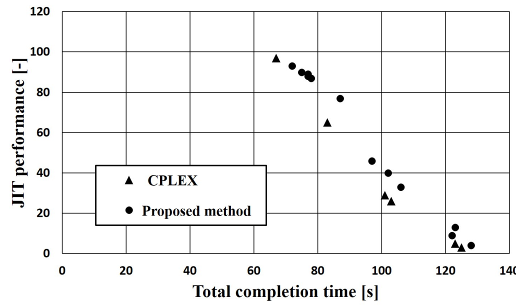

Fig. 7. Trade-off between JIT performance and total completion time (case9)

VI. CONCLUSION

We proposed a heuristic algorithm for the JIT routing problem of AGVs. The advantage and its usefulness of the proposed method was confirmed by comparing the performance with CPLEX from computational results. Also, near-optimal solutions can be obtained by changing the weight of the JIT performance and the total completion time. The proposed method can be applied to larger transport system layout with more AGVs and more tasks. Our future work is to verify the performance for large-scale real-world problems.

REFERENCES

[1] I. F. A. Vis, Survey of research in the design and control of automated guided vehicle systems, *European Journal of Operational Research*, vol. 170, no. 3, pp. 677–709, 2006.

[2] H. W. Cheong, H. Lee, Requirements of AGV (Automated Guided Vehicle) for SMEs (Small and Medium-sized Enterprises), *Procedia Computer Science*, vol. 139, pp. 91–94, 2018.

[3] G. Daugherty, S. A. Reveliotis, G. Mohler, Optimized multiagent routing for a class of guidepath-based transport system, *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 1, pp. 363–381, 2019.

[4] J. Chen, W. Weng, G. Rong, S. Fujimura, Integrating genetic algorithm with time control for just-in-time scheduling problems, *IFAC Symposium on Information Control Problems in Manufacturing*, vol. 48, no. 3, pp. 893–897, 2015.

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

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

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

[8] Y. Tanaka, T. Nishi, M. Inuiguchi, Dynamic optimization of simultaneous dispatching and conflict-free routing for automated guided vehicles - Petri net decomposition approach -, *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 4, no. 3, pp. 701–715, 2010.