Trans. of the Society of Instrument and Control Engineers
Vol.E-3, No.1, 82/90 (2004)

An Autonomous Distributed Route Planning Method for Multiple Mobile Robots†

Masakazu ANDO*, Tatsushi NISHI**,
Masami KONISHI*** and Jun IMAI***

Route planning of multiple AGVs (Automated Guided Vehicles) is expected to minimize the transportation time without collision and deadlock among the AGVs in many transportation systems. In this paper, we propose an autonomous distributed route planning method for multiple mobile robots. The proposed method has a characteristic that each mobile robot individually creates a near optimal route through the repetitive data exchange among the robots and the local search of route using Dijkstra's algorithm. The proposed method is applied to several transportation route planning problems. The optimality of the solution generated by the proposed method is evaluated using the duality gap derived by Lagrangian relaxation method. A near optimal route plan within 5% of duality gap for a large scale transportation system consisting of 143 nodes and 15 AGVs can be obtained with five seconds of computation time by using Pentium III (1GHz) processor. Moreover, it is also shown that the proposed method is effective for various types of problems despite the fact that each route for AGV is created without optimizing the entire objective function even when the velocity of each AGV is different.

Key Words: autonomous distributed system, route planning, AGV, transportation, Dijkstra's algorithm

1. Introduction

Multiple mobile robots are widely used for automated guided vehicles in clean room for semiconductor manufacturing, material handling system in production systems, or material transportation system in medical center, etc. It is necessary to generate collision-free route planning for multiple AGVs efficiently so that the total transportation time is minimized. Conventional route planning method has been configured on the assumption that the entire route planning is determined by single centralized decision making system [1) 2)]. In the field of motion planning for multiple mobile robots, there have been a wide variety of studies focusing on path planning methods such as probabilistic roadmap [3)], motion planning using super-graph [4)] or coordination method [5)] in field environments without specified lanes. This paper deals with route planning problems for multiple AGVs in various transportation systems with guided paths in ladder structure.

In recent years, route planning problems for multiple AGVs is increasingly complex with regard to relationship with rapid expansion of AGV transportation systems. Thus, it is intractable to generate collusion-free route planning for multiple AGVs in real time. Moreover, it is also quite difficult for conventional centralized route planning system to adjust itself for unforeseen circumstances such as disturbances or the change of its environment.

In such situations, autonomous distributed systems have received much attention from the viewpoint of its fault tolerance and flexibility. Autonomous distributed system consists of some of agents. Each agent has its individual criterion and has the capability to achieve global objective without having centralized controller. These systems are widely applied to production scheduling or transportation systems control [6)∼8)].

For autonomous distributed motion planning methods, a number of studies have been previously reported [9) 10)]. Gou et al. [10)] decomposed the motion planning problem into path planning and velocity planning for dynamic environments. However, the method may deteriorate the performance of solution when the coordinated solution is quickly derived.

In this paper, we propose an autonomous distributed route planning method for multiple AGVs. In the proposed method, each AGV generates the solution to minimize the objective function for itself, while repeating the communication between other AGVs and replanning for each AGV. The penalties for violating collision-free constraints are added in the objective function for each AGV. The weighting factor for the penalty function is gradually

* Mitsubishi Materials Corporation, Naoshima Works, Kagawa
** Osaka University, Graduate School of Engineering Science, Toyonaka city, Osaka
*** Okayama University, The Graduate School of Natural Science and Technology, Okayama city, Okayama

increased until a feasible solution fo the entire AGV is derived. The performance of the proposed method and its computation time are investigated in this study.

Since the conventional autonomous distributed methods are configured to derive a feasible solution in a shorter computation time based on the use of heuristics for coordination without using the entire information, it is still difficult to obtain near-optimal solution for autonomous distributed methods. From the reason, the optimality of the solution for the proposed method is evaluated by lower bound in this study.

Furthermore, in order to evaluate the flexibility of the algorithm, the proposed method is applied to 3-dimensional transportation system with different AGV speed. The performances of the proposed method on a variety of different conditions are also studied.

This paper is organized as follows. Section 2 defines the transportation system model and problem formulation. In Section 3, we propose an autonomous distributed algorithm for route planning problems. In Section 4, the proposed method is applied to example problems. The performance of the proposed method is compared with that of conventional methods. The optimality of solutions is evaluated. Section 5 studies flexibility of the proposed algorithm for 3-dimentional transportation system with different AGV speed. The effectiveness of the autonomous distributed method is demonstrated. Section 6 concludes our study and mentions future works.

2. Problem definition

2.1 Transportation system model

The two-dimensional transportation system is shown in **Fig. 1**. The transportation system consists of nodes and edges. Each node represents a place where each AGV can stop or turn, while each edge represents a lane for traveling between places. The following conditions are assumed for route planning problem treated in this section.

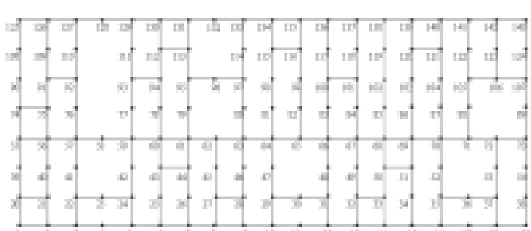


Fig. 1 2D layout model of a transportation system

- The length of edges are all equivalent.
- Each AGV can travel on an edge in bi-directional way.
- Each AGV can stop or turn only at a node.
- Requests for transportations are generated only when each of AGV is on a node.

The collision avoidance constraints can be stated as follows.

(i) Two or more than two AGVs cannot travel into a node at the same time (collision avoidance constraint on a node).

(ii) Two or more than two AGVs cannot travel on a node at the same time (collision avoidance constraint on an edge).

Given one pair of a loading node and an unloading node for each AGV and the initial node for each AGV, the problem is to determine the route planning to minimize the total traveling time for AGVs including waiting time satisfying collision avoidance constraints.

2.2 Integer Programming Problem Formulation

Let H be total time horizon and minimum traveling time be one time period where time period t ($t = 1, 2, \ldots, H$) is from time $t-1$ to time t. The dynamics of AGVs are represented by discrete time when the velocity for traveling between adjacent nodes is equal. Let $x^k_{i,j,t}$ denote a binary variable that takes the value 1 when AGV k travels from node i to node j in time period t, otherwise it takes the value of 0. The route for all AGVs can be determined by $x^k_{i,j,t}$ $(\forall k, i, j, t)$. AGV k stops at node i in time period t when $x^k_{i,i,t} = 1$. Let N be the set of nodes in the transportation system, N_i be the set of nodes adjacent to node i, V be the set of AGVs. It is assumed that neighborhood structure defined by N_i is symmetric. The constraints for route planning problem are explained as follows.

$$\sum_{j \notin N_i} x^k_{i,j,t} = 0 \quad (k \in V,\ i \in N;\ t = 1, \ldots, H) \tag{1}$$

$$\sum_{j \in N_i} x^k_{i,j,t} \leq 1 \quad (k \in V,\ i \in N;\ t = 1, \ldots, H) \tag{2}$$

$$\sum_{j \in N_i} x^k_{j,i,t} = \sum_{n \in N_i} x^k_{i,n,t+1} \quad (k \in V,\ i \in N;\ t = 1, \ldots, H) \tag{3}$$

$$\sum_{j \in N_{S_k}} x^k_{S_k,j,0} = 1 \quad (k \in V) \tag{4}$$

S_k denotes the initial node for AGV k. (1) restricts that AGV k cannot travel from node $i \in N$ to the node j which does not belong to N_i in a time period. (2) ensures that AGV k can travel from node i to only one node $j \in N_i$. (3) is related to time continuity constraints of traveling route for AGV k. (4) denotes the initial condition for the position of AGVs. The collision avoidance constraints of (i), (ii) can be written as (5) and (6), respectively.

$$\sum_{k\in V}\sum_{j\in N} x^k_{j,i,t} \le 1 \quad (i\in N;\ t=1,\ldots,H) \qquad (5)$$

$$\sum_{k\in V}(x^k_{i,j,t}+x^k_{j,i,t}) \le 1 \quad (i\in N,\ j\in N_i;\ t=1,\ldots,H) \qquad (6)$$

(5) represents the capacity constraint of each node which implies that for every node $j\in N$, two or more than two AGVs cannot travel into a node j at the same time. (6) represents the capacity constraint of each edge which indicates that two or more than two AGVs cannot travel on an edge at a time. A binary variable $\delta_{k,t}$ is defined to describe the transportation time. $\delta_{k,t}\in\{1,0\}$ takes the value of 1 if AGV k has not arrived at the ending node in time period t, otherwise it takes the value of zero. This condition can be written as the following constraints.

$$\sum_{i\in N_{G_k}} x^k_{i,G_k,t} \le M(1-\delta_{k,t}) \quad (k\in V;\ t=1,\ldots,H) \qquad (7)$$

$$\sum_{i\in N_{G_k}} x^k_{i,G_k,t} \ge 1-\delta_{k,t} \quad (k\in V;\ t=1,\ldots,H) \qquad (8)$$

where G_k is the goal node for AGV k and M is the upper bound of the left-hand side of (7)($M=1$ from (5)). If the AGV has arrived to the goal node, the AGV stops at the node. Thus, the following constraints must be satisfied.

$$-\delta_{k,t}+\delta_{k,t+1}\le 0 \quad (k\in V;\ t=1,\ldots,H) \qquad (9)$$

From the above constraints, the route planning problem is formulated as

$$\min_{\{x^k_{i,j,t}\}} \sum_k\sum_t \delta_{k,t} \qquad (10)$$

subject to (1)-(9).

3. Autonomous Distributed Route Planning Method

3.1 Route Planning Algorithm

An autonomous distributed route planning algorithm is proposed in this study. In the proposed algorithm, each AGV is regarded as a subsystem. Each subsystem repeats the generation of route and data exchange among the subsystems. **Fig. 2** shows the flowchart of the proposed algorithm. The detailed algorithm for each AGV k consists of the following steps.

Step 1 **Initial generation of route planning**

Each AGV generates the shortest route by using Dijkstra's algorithm without taking into account the route of other AGVs.

Step 2 **Communication with other AGVs**

Each AGV communicates with other AGVs and gets the information of tentative route plan $\{\bar{x}^l_{i,j,t}\}$ from all of other AGVs.

Step 3 **Evaluation of convergence**

If the routing generated by each AGV satisfies both of the following conditions, each AGV stops the algorithm and the derived route plan is regarded as a final solution.

- Each AGV generates the same route as that derived in a previous iteration.
- The route generated by each AGV has no interference with that of other AGVs.

Step 4 **Judgment on whether the replanning is skipped or not**

In the algorithm, each AGV generates its routing concurrently. In some cases, the same route plans are generated cyclically. In that case, one of the AGV skips the replanning of Step 5. In this step, each AGV determines whether the Step 5 is skipped or not with a certain probability (20%-40%). The detail of skipping is explained in Section 3.2.

Step 5 **Route replanning**

Each AGV generates the routing to optimize its own objective function I_k of (11) by using the route planning data of other AGVs.

$$I_k = \sum_t \delta_{k,t} + \sum_{l\in V, l\ne k} \alpha_{k,l}(r)(C^1_{k,l}+C^2_{k,l}) \qquad (11)$$

$$C^1_{k,l} = \sum_t \sum_{i\in N_{P(k,t)}} \bar{x}^l_{i,P(k,t),t} \qquad (12)$$

$$C^2_{k,l} = \sum_t \bar{x}^l_{P(k,t),P(k,t-1),t} \qquad (13)$$

$\{\bar{x}^l_{i,j,t}\}$ denotes the tentative route planning result derived by AGV l in a previous iteration. $P(k,t)$ is the node to which AGV k arrives in the end of time period t, $\alpha_{k,l}(r)$ is the weighting factor for violating collision avoidance constraints at rth iteration between AGV k and other AGV l, $C^1_{k,l}$ is the infeasibility for violating collision avoidance constraint for AGV k with other AGVl on each node. $C^2_{k,l}$ is the infeasibility for violating collision avoidance constraint for AGV k with other AGV l on each edge. The route replanning problem is singel AGV route planning problem which can be solved by Dijkstra's algorithm.

Step 6 **Updating weighting factor for penalty function**

If the derived route at Step 5 is infeasible, weighting factor for penalty function $\alpha_{k,l}(r+1)$ is increased for each AGV by (14) and then return to Step 2.

$$\alpha_{k,l}(r+1) = \alpha_{k.l}(r) + \Delta\alpha \sum_{l \neq k} (C^1_{k,l} + C^2_{k,l}) \qquad (14)$$

At the fist step of the proposed algorithm, each AGV generates the shortest route plan without considering other AGVs. If each AGV creates the route plan individually, the derived solution is infeasible for the entire AGVs. Thus, at the second step of the algorithm, the penalties violating collision avoidance constraints are added in the objective function. The communication between AGVs and reroute planning are repeated until a feasible solution for the entire AGVs is derived. Each subsystem can adopt the same optimization algorithm. Therefore, it is possible to construct the entire route planning system by using the same algorithm for each AGV in the multiple AGVs transportation systems.

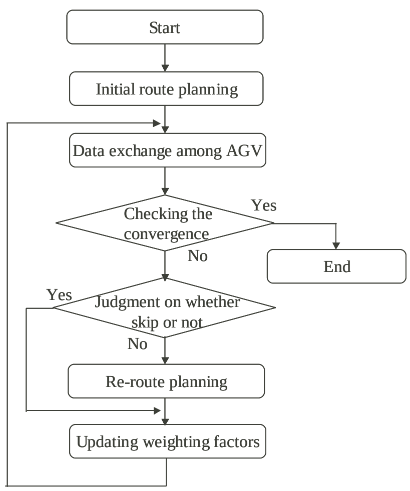


Fig. 2 Distributed route planning algorithm for each AGV

3. 2 Skipping of route replanning

Consider a simple example of 2 AGVs system where requests are given as arrows shown in **Fig. 3**.

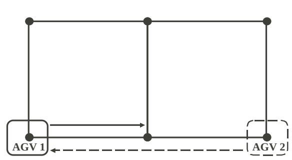
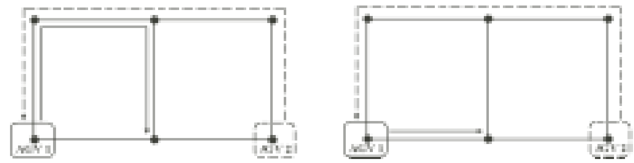


Fig. 3 The transportation request and its initial route plan

For the problem, an example of route planning result for the proposed method is shown in **Fig. 4**(a) and Fig. 4(b). At the first step of the algorithm shown in Fig. 2, each AGV generates the shortest route planning without considering the collision with other AGVs(Fig. 3). The derived route plan is exchanged between AGVs and each



(a) Infeasible route plan (b) Feasible route plan

Fig. 4 An example of an intermediate result obtained by the proposed method

AGV executes replanning considering the other AGVs. However, both of AGVs generates the infeasible route plan as shown in Fig. 4(a) to reduce the penalties violating collision avoidance constraints with the route derived in a previous iteration. Then, the infeasible route plan shown in Fig. 3 is generated again in the next iteration. In such situations, the same solutions are cyclically generated, making it difficult to converge for the proposed algorithm. To avoid this situation, the step of the replanning is skipped randomly for each AGV. This procedure is called as skipping which can improve the convergence of the proposed algorithm. The appropriate value of skipping probability is studied in Section 4. 2.

3. 3 Route planning method for each AGV

Each AGV generates the route by the shortest path algorithm as explained in Section 3. 1. Dijkstra's algorithm can derive an optimal route to minimize the objective function of (11) for each AGV. The cost of each edge consists of the traveling time between the adjacent nodes and penalty costs for violating the collision avoidance constraints on each node and each edge. The penalty costs for each AGV are calculated on the assumption that the route for other AGVs is fixed as the tentative solution which is derived in a previous iteration.

4. Numerical experiments

4. 1 Example 1 : 2-dimensional transportation system

In this section, we treat an example problem for transportation system with 143 nodes and 7 AGVs shown in Fig. 1. **Table 1** shows the starting node (S node) and ending node (E node) which are given to each AGV. The velocity for all AGVs is equal and the traveling time between adjacent two nodes is one time period. Total planning horizon H is 100 and the parameters are set as $\Delta\alpha = 0.8$, and the skipping probability is 25%. The results of an example route planning is shown in **Fig. 5**. From the results of Fig. 5, an optimal solution satisfying collision avoidance constraints without any waiting time is derived by the proposed method. The total computation time is 1 second when a Pentium III 1GHz processor with 256MB memory is used.

Table 1 Transportation request for each AGV

AGV	1	2	3	4	5	6	7
S node	1	125	96	105	52	104	26
E node	99	38	49	92	23	109	123

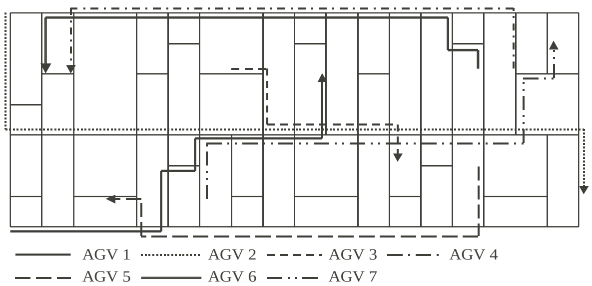


Fig. 5 The results of route planning for the example problem

4.2 Effects of skipping probability to the performance of the proposed algorithm

In this section, the effects of skipping probability to the performance of the proposed method are investigated. The transportation system model with 143 nodes and 7 AGVs as shown in Fig. 1 is used for numerical experiments. Ten types of requests are generated randomly without duplication. The average results for one hundred of computational experiments are derived. The relationship between the average number of data exchange and the value of objective function (total sum of transportation time for AGVs) is shown in **Fig. 6**. From the

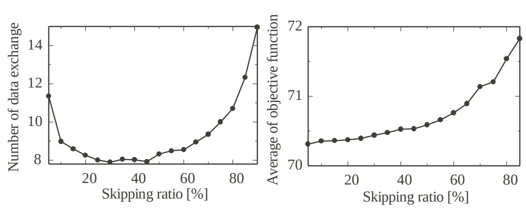


Fig. 6 The effects of skipping ratio

results of Fig. 6, the average value of objective function is increased when the skipping probability is increased. However, the average number of data exchange has minimum point when the skipping probability is almost 40%. The total computation time is increased when the average number of data exchange is increased. Thus, there is tradeoff relationship between total computation time and the quality of solution. From this consideration, the skipping probability should be selected from 25% to 45% because the average number of data exchange is almost constant. In this study, he skipping probability is set to 25%.

4.3 Optimality of solution and computation time for the proposed method

In this section, the effectiveness of the proposed method is investigated from the viewpoint of optimality of solutions and total computation time. At the first step, the optimality of solution for the proposed method is studied.

The interference or waiting time between AGVs is increased to avoid collision if the number of AGVs is increased for a transportation system. For such situations, it is extremely difficult to derive an optimal solution by using branch and bound method or dynamic programming. Therefore, we derive a lower bound of the original problem by Lagrangian relaxation method for the problem. Then the optimality of solution for the proposed method is evaluated by

$$\text{duality gap}: \ D = \frac{J - L^*}{L^*} \times 100[\%] \tag{15}$$

The duality gap is used to evaluate the optimality of solution quantitatively by using the upper bound and the lower bound of the problem. The lower bound is computed by Lagrangian relaxation technique shown in the appendix. The duality gap is calculated by (15) for each problem. J is the value of objective function, L^* is the value of lower bound obtained by using Lagrangian relaxation. The duality gap is close to zero if the solution of the proposed method is close to optimum. If the duality gap is D[%], it means that there is no feasible solution whose objective function is under D[%] of the current optimal solution. For numerical experiments explained in Section 4.1, the derived solution of Fig. 5 is optimal because D is zero.

The optimality of solutions are investigated when the number of AGVs is changed from 2 to 15 for the transportation system shown in Fig. 1. Ten types of requests are generated randomly. The lower bound is derived by Lagrangian relaxation for each problem. The average duality gap for ten times of computational results is derived. **Fig. 7** shows the computational results. From the results of Fig. 7, the duality gap is gradually increased when the number of AGVs is increased. This is due to the fact that it becomes more difficult to derive optimal solution for the route planning problems when the number of AGVs is increased. It is demonstrated from the numerical experiments that near optimal solution with 5% of duality gap can be derived for 15 AGVs problems. The proposed method can derive near optimal solution for every case.

At the second step, the computation time for the proposed method is studied. To derive a feasible route planning is one of the most important factor for the application to semiconductor fabrication bays. It is requested that the route planning should be executed in a few seconds for dynamically given transportation requests. The computation time to derive a feasible solution for the proposed algorithm is evaluated and the performance is compared with a conventional route planning method.

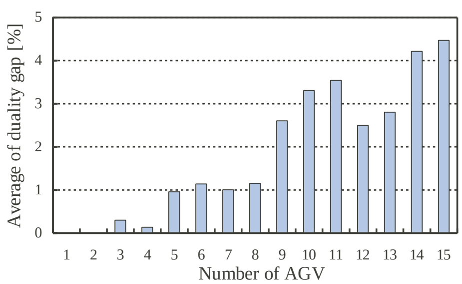


Fig. 7 Relationship between the number of AGV and average of duality gap

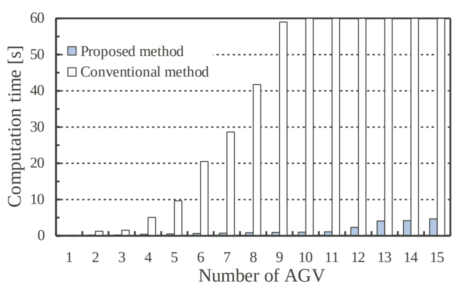


Fig. 8 Comparison of average of computation time

The algorithm of the conventional method is based on the simulated annealing method where a feasible route planning for the entire system of AGVs is always generated by successively improving neighborhood solution. The neighborhood solution is effectively generated according to the probability of selecting the succeeding route planning node, which is calculated by the position of other AGVs in the same way as the numerical potential fields, in order not to generate non-promising solutions. The main difference between the proposed method and the conventional method is that the infeasible solution for the entire system is successively improved for the proposed method because each AGV individually derives the route planning for itself. **Fig. 8** shows the total computation time for both methods when the number of AGVs is changed. For the conventional method, the total computation time is determined so that the duality gaps for both methods are equal. From the results of Fig. 8, the difference of computation time for both methods is small when the number of AGVs is less than 3. This is because if the number of AGVs is small, the optimal solution can easily derived by both methods. However, for the case of problems when the number of AGVs is more than 4, the difference of computation time for both methods becomes larger and the difference increases with respect to the increase of the number of AGVs. From these results, it is demonstrated that the proposed method can derive a near optimal solution in a shorter computation time compared with that of conventional method.

Finally, the optimal solution is derived by branch and bound method using commercial integer programming solver CPLEX (ILOG(C)) by using the formulation in Section 2. 2. It is conformed that the CPLEX can derive the same solution for 29 node and 7 AGVs. However, the optimal solution cannot be derived for 143 node problems due to the increase of binary variables. From these results, it is demonstrated that the proposed method can generate near optimal solution for 2-dimensional transportation system shown in Fig. 1 within 5 seconds of practical computation time.

5. Flexibility of autonomous distributed route planning algorithm for a variety of probelems

In this section, the performance of the proposed algorithm is investigated for a variety of problems. We study the flexibility of the proposed method for applying various types of problems.

5. 1 Example 2 : Application to 3-dimensional transportation system

In this section, 3-dimensional transportation system representative for hospitals or buildings are treated. For 3-dimensional transportation systems, the number of branching for route planning is larger than that of 2-dimensional transportation systems. Therefore, it is predicted that it is not easy for autonomous distributed algorithm to derive a feasible solution. It is assumed that the velocity is all the same in Section 3 but the vertical velocity is different from the horizontal velocity for traveling for 3-dimensional transportation systems. It is necessary to generate a collision free route planning for AGVs with different velocity. There are some cases that one AGV is traveling on a node and the other is traveling on an edge. In order to consider practical situations, tuning time is also taken into account in the model. Each AGV stops on a node in a fixed time when it is turning at a node. Example problem for 3-dimensional transportation sys-

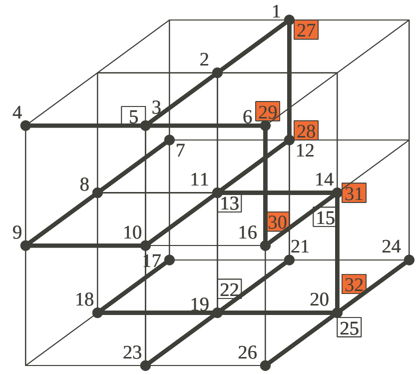


Fig. 9 3D layout model of a transportation system

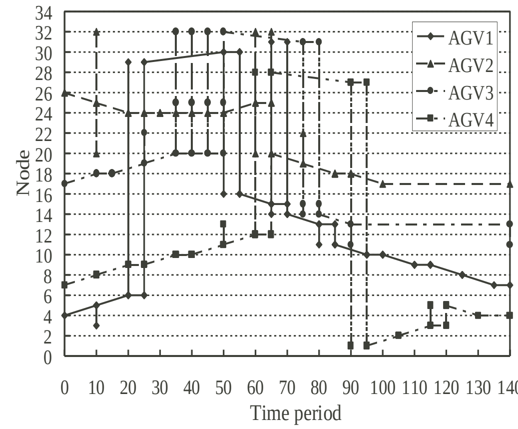


Fig. 10 A time chart representation of the result for example 2 obtained by the proposed method

tem is explained. There are 4 AGVs in the 3-dimensional transportation system in **Fig. 9**. The vertical and horizontal traveling time between the adjacent two nodes, turning time are shown in **Table 2**. The transportation requests shown in **Table 3** are given to each AGV. The route planning results for the proposed method are shown in **Fig. 10**. Fig. 10 is the time chart illustrating time on x-axis and the node number at each time on y-axis. Note that each AGV has more than 2 nodes at the same time on the chart in some cases since the node number is duplicated in Fig. 9 and Fig. 10. It is easy to observe the deadlock between AGVs duplicating the node number because crossing occurs in the time chart if some AGVs are crossing against on an edge. It is confirmed that a feasible route planning without interference between AGVs is successfully derived by the proposed method from Fig. 10. The computation time for the proposed method is 0.15 sec. On the other hand, the simulated annealing method takes about 380 times of computation time to solve the problem. The conventional method generates candidates of route planning randomly. A lot of computation time is required to derive a feasible solution for the simulated annealing method.

Table 2 Traveling time required to perform each action

Action	Required time
Horizontal movement (between the neighbor two nodes)	10
Vertical movement (between the neighbor two nodes)	25
90 degree turn	5
180 degree turn	5

Table 3 Transportation request for each AGV

AGV	1	2	3	4
S node	4	26	17	7
E node	7	17	11	4

5.2 Flexibility study for different velocity conditions

Table 4 shows the average results of 100 times of computations when the velocity for all AGVs is constant (Case 1) and when the velocity of AGV 3 is 1/2 (Case 2). The total transportation time for AGV 3 in the route planning results for Case 2 (the velocity of AGV 3 is 1/2) is supposed to 2 times of the route planning results for Case 1 if each AGV creates its route planning individually. However, the transportation time for AGV 2 in Case 2 is less than two times of transportation time for AGV 2 in Case 1 from the results in Table 4. The results can be interpreted that AGV 2 requested to change route planning for other AGVs in order to minimize the increase of transportation time caused from the bottleneck of AGV 2. **Fig. 11** shows the transition of objective function during the iteration for AGV 2 and AGV 3 in the proposed algorithm. The objective function for each AGV is gradually increasing when the penalties for collision avoidance are increased when the number of data exchange is increased. After 5 times of data exchange, AGV 2 takes another route to avoid collision. Then, the objective function for AGV 3 is decreased. Thus, a feasible route planning can be derived. **Table 5** shows the comparison results of total transportation time (Trans. time) and total computation time (Comp. time) with conventional method using simulated annealing method. The average results of 10 times computations for the proposed method and SA method are shown in Table 5. From the results of Table 5, it is demonstrated that the proposed method can derive better solutions in shorter computation time compared with simulated annealing method even though only local information is used to derive a solution for the proposed method for all cases. From these results it can be said that the proposed algorithm has an autonomous function for deriving the solution where the bottleneck AGV is placed much importance by adopting the algorithm of gradually increasing the penalties for violating interferences between other AGVs during the data exchange between AGVs.

Table 4 Transportation time for each AGV

Case	AGV Number	Transportation time
1	1	135.2
	2	65.65
	3	114
	4	130
2	1	135
	2	85
	3	145
	4	130

Table 5 Comparison of the proposed method and the SA method

Case	Performance	The proposed method	SA method
1	Trans. time	466	440
	Comp. time	0.2250	57.063
2	Trans. time	495	495
	Comp. time	0.1242	26.218

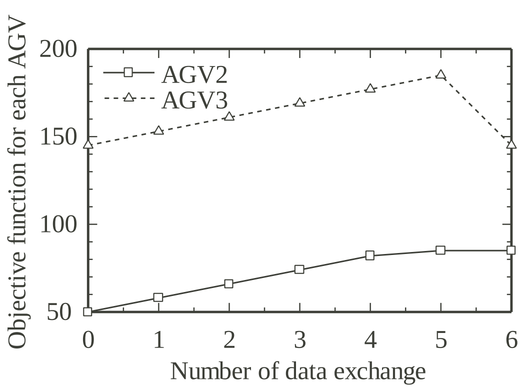


Fig. 11 Transition of evaluation(AGV2,AGV3)

6. Conclusion

In this paper, an autonomous distributed route planning method for multiple mobile robots has been proposed. In the proposed algorithm, each AGV repeats the data exchange and regeneration of route planning for itself to minimize total transportation time. The proposed algorithm can derive near optimal solution in short computation time avoiding deadlock and interferences between AGVs.

In this study, the lower bound is derived by Lagrangian relaxation method and it is used to derive duality gap for the evaluation of optimality of solutions derived by the proposed method.

The results demonstrate that the proposed method can derive near optimal solution with 5% of duality gap within 5 seconds of computation time for transportation system with 143 nodes and 15 AGVs.

In order to show the applicability of the proposed method to a variety of problems, the proposed method is applied to 3-dimentional transportation system consisting of AGVs with different velocity. The results demonstrate that the proposed algorithm has a function to place much importance on the bottleneck to derive a solution even though each AGV creates its route planning to minimize the objective function for itself.

There is a limitation that the algorithm can be applied to minimize the additive objective function for each AGV. This problem dependency is caused by the ease of minimizing the entire objective function for setting the minimization of additive objective function such as total transportation time for each AGV. However, in practice, there are other types of problems to minimize maximum completion time for AGVs transportation route planning problems even though there exists a lot of problems with additive objective function.

Therefore, further study should be devoted to consider the applicability to another types of objective function such as max-min type for the proposed methodology. Moreover, the proposed algorithm can be easily implemented to parallel processing system to reduce computation time since each AGV has its objective function and constraints. The implementation of the algorithm using multiple processing systems and its investigation and the realization of distributed system for large-scaled problems in dynamic situation is our future study.

Acknowledgment

The authors would like to thank the funding provided by Electric Technology Research Foundation of Chugoku.

References

1) S. Endo and M. Konishi: Motion Planning for Multiple Mobile Robots by a Petri Net approach, *Proceedings of 1998 Japan-USA Symposium on Flexible Automation*, 587/594 (1998)
2) S. Endo, M. Konishi, T. Moriwaki, and M. Yoshida: Motion Planning Method for Multiple Mobile Robots in Large Scale Transportation System by Genetic Algorithms, *Trans. Inst. Syst. Control, Inf. Eng.*, **13**-3, 115/123 (2000)
3) J.C. Latombe: Robot Motion Planning, *Kluwer Academic Publishers*, Boston (1991)
4) P. Svestka and M.H. Overmars: Coordinated Path Planning for Multiple Robots, *Robotics and Autonomous Systems*, **23**, 125/152 (1998)

5) C. Ferrari, E. Pagello, J. Ota and T. Arai: Multirobot motion coordination in space and time, *Robotics and Autonomous Systems*, **25**, 219/229 (1998)
6) K. Sotobayashi, M. Konishi and T. Nishi: Application of decentralized agent method to dynamic route planning method for mobile robots, Proc. Production Scheduling Symposium 2001, 167/172 (2001)
7) T. Kitajima, H. Nishitani, S. Hasebe and I. Hashimoto: Performance Evaluation of the Distributed Scheduling System Using Message Passing, Transactions of the Society of Instrument and Control Engineers, **35**-4, 546/553 (1999)
8) Tatsushi Nishi, Shinji Hasebe and Iori Hashimoto: Machine-Oriented Decomposition of Flowshop Scheduling Problems Using Lagrangian Relaxation, *Trans. Inst. Syst. Control, Inf. Eng.*, **14**-5, 244/251 (2001)
9) D. M. Maniezzo and V. Colorni: Ant System Optimization by a Colony of Cooperating Agents, *IEEE Transactions on System, Man, and Cybernetics-Part B*, **26**-1, 1/13, (1996)
10) Y. Gou and L.E. Parker: A Distributed and Optimal Motion Planning Approach for Multiple Mobile Robots, *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, 2612/2619 (2002)
11) M.L. Fisher: The Lagrangian Relaxation Method for Solving Integer Programming Problems, *Management Science*, **27**-1, 1/18 (1981)

Appendix A. Lower bound derivation by Lagrangian relaxation method

The constraints of (5) and (6) are relaxed by using non-negative Lagrangian multiplier $\{\lambda_{i,t}\}$, $\{\phi_{i,j,t}\}$. The Lagrangian function L is written as

$$L = \sum_k \sum_t \delta_{k,t} + \sum_{i\in N} \sum_t \lambda_{i,t} \left(\sum_k \sum_{j\in N_i} x^k_{j,i,t} - 1 \right) + \sum_{i\in N} \sum_{j\in N} \sum_t \phi_{i,j,t} \left[\sum_k (x^k_{i,j,t} + x^k_{j,i,t}) - 1 \right] \quad \text{(A. 1)}$$

(A. 1) can be rewritten as

$$L = \sum_k L_k - \sum_t \sum_{i\in N} \lambda_{i,t} - \sum_t \sum_{i\in N} \sum_{j\in N} \phi_{i,j,t} \quad \text{(A. 2)}$$

where L_k is defined as

$$L_k = \sum_t \delta_{k,t} + \sum_t \lambda_{P_{(k,t)},t} + \sum_t (\phi_{P_{(k,t-1)},P_{(k,t)},t} + \phi_{P_{(k,t)},P_{(k,t-1)},t}) \quad \text{(A. 3)}$$

The Lagrangian relaxation problem to minimize L can be decomposed into each AGV subproblem to minimize L_k, which can be solved by Dijkstra's algorithm.

Step 1 Initialization of Lagrangian multipliers

$\lambda_{i,t} \leftarrow 0,\ \phi_{i,j,t} \leftarrow 0\ (\forall i,j,t)$

Step 2 Solving Lagrangian relaxation problem

The subproblem for each AGV is solved by minimizing (A. 3) where the Lagrangian multipliers $\{\lambda_{i,t}\}$, $\{\phi_{i,j,t}\}$ are fixed.

Step 3 Updating of lower bound

The value of L is calculated by (A. 2) from the solution at Step 2 and the lower bound is updated if the value is larger than previous one.

Step 4 Updating Lagrangian multipliers

The Lagrangian multipliers are updated by subgradient method for each node i, j at each time period.

Step 5 Evaluation of convergence

The convergence of lower bound is evaluated and the steps 2 to 4 are repeated if the lower bound is not converged.

Masakazu Ando

He received the B.E., M.E. degrees from Okayama University in 2002 and 2004, respectively. Since 2004, he joined the Mitsubishi Materials Corporation. Currently, he is with the Naoshima Works, Kagawa, Japan.

Tatsushi Nishi (Member)

He received the B.E., M.E. and Ph.D. degrees from Kyoto University in 1996, 1998 and 2001, respectively. Since 2001, he has been an Assistant Professor with the Department of Electrical and Electronic Engineering, Okayama University. Since 2006, he has been an Associate Professor with the Graduate School of Engineering Science, Osaka University. His research interests include discrete optimization, production scheduling and supply chain optimization. He is a member of IEEE, the Institute of System, Control and Information Engineers, and the Operations Research Society of Japan.

Masami Konishi (Member)

He received the M.E. and Ph.D. degrees from Kyoto University in 1969 and 1976. Since 1999, he has been a Professor with the Department of Electrical and Electronic Engineering, Okayama University. Before 1999, he was a Principal Researcher for the Technical Development Group, Kobe Steel, Ltd. His research interests are intelligent process control and decentralized scheduling. He is a member of IEEE, the Institute of System, Control and Information Engineers, and the Iron and Steel Institute of Japan.

Jun Imai (Member)

He received the B.E. and Ph.D. degrees from Kyushu University in 1987 and 1992, respectively. Since 2000, he has been a lecturer with the Department of Electrical and Electronic Engineering, Okayama University. His research interests include modeling and control of distributed parameter systems. He is a member of IEEE, the Institute of System, Control and Information Engineers.

Reprinted from Trans. of the SICE
Vol. 39 No. 8 759/766 2003