International Symposium on Scheduling 2013
July 18-20, 2013, Tokyo, Japan

4B1

Column Generation Heuristics to Route Planning Problems for Automated Guided Vehicles with Acceleration and Deceleration

S. Matsushita[1], T. Nishi[1]
[1] Graduate School of Engineering Science, Osaka University
1-3 Machikaneyama-cho, Toyonaka city 560-8531, Japan
matsushita@inulab.sys.es.osaka-u.ac.jp, nishi@sys.es.osaka-u.ac.jp
S. Kataoka[2], T. Hisano[2], K. Matsumura[2]
[2]Daifuku Institute of Technology and Development Co., Ltd.
1225 Nakazaiji Hino-cho, Gamogun, Shiga Prefecture, Japan
{satomi_kataoka, takashi_hisano, katsuhiko_kataoka}@ha.daifuku.co.jp

Abstract

In this paper, we propose a column generation approach to solve the route planning problem for automated guided vehicles with acceleration and deceleration. The transportation model is discretized into regular intervals. A network model is created by taking into account the acceleration and deceleration motions of AGVs. Column generation heuristics is developed to find a near-optimal solution. The pricing problem is represented by a resource-constrained shortest path problem, which is effectively solved by a labeling algorithm. By comparing the performance of the conventional method, the effectiveness of the proposed method is demonstrated.

Keywords: automated guided vehicle, column generation, routing, labeling algorithm, heuristics

1. INTRODUCTION

In transportation systems such as semiconductor plants, container terminals, and Flexible Manufacturing Systems (FMS), multiple automated guided vehicles (AGVs) are widely used. Transportation tasks are performed loading-points and unloading-points. In practice it is required to generate conflict-free routes for multiple AGVs, which minimize the total delivery time.

In previous research, there are many researches of vehicle route planning problem Bunte and Kliewer, (2009). A branch-and-cut-and-price algorithm for vehicle route planning problem using column generation approach was proposed in Bettinelli, *et al.*, (2011). Reveliotis and Roszkowska (2011) regard the vehicle route planning problem as a resource allocation system and develop a deadlock avoidance policy. In Feillet, *et al.*, (2004), the vehicle routing problem (VRP) is regarded as an elementary shortest path problem with resource constraints and a labeling algorithm and column generation approach are used. Recently, route planning problems for AGV have been widely studied (Le-Anh and Koster, 2006). Nishi, *et al.*, (2005) used a Lagrangian decomposition technique for AGV route planning problem. Tanaka, *et al.*, (2010) developed a Petri net decomposition approach with deadlock avoidance to dynamically solve the AGV route planning problem. Desaulniers, *et al.*, (2003) used a column generation approach for AGV route planning problems. With the column generation approach a tight lower bound can be derived by solving the restricted master problem which is a linear programming problem with huge number of columns. In previous studies, for simplicity the AGV speed is constant. However, in a real delivery system AGVs use acceleration and deceleration at moving, stopping and turning. It is therefore extremely important point to take into consideration acceleration and deceleration during the route planning phase.

In this paper we propose a column generation heuristics to solve the route planning problem for AGVs with acceleration and deceleration. In the column generation approach, a tight lower bound with good accuracy is derived by solving the restricted master problem and pricing problem repeatedly. In the restricted master problem, the dual variables for the restricted master problem for a limited set of columns are updated by solving a linear programming problem which takes into consideration collision avoidance constraints. In the pricing problem, a route that satisfies the constraints of speed and task assignment is generated for each AGV. The solution of the pricing problem with a minimum reduced cost is added to the set of columns in the restricted master problem. The column generation approach is effective for solving linear programming problem with a huge number of columns because it is not necessary to create all column candidates. The pricing problem can be solved effectively by a labeling algorithm using dominance (Feillet, *et al.*, 2004). We can obtain a solution efficiently by removing non-optimal labels by dominance relation.

The solution obtained by column generation approach is generally infeasible because it is equivalent with the continuous relaxation problem of the Dantzig-Wolfe reformulation of the original problem. In order to create a feasible solution, heuristics are required to modify the infeasible solution. In this paper, we propose an efficient heuristic algorithm to generate a feasible solution. Computational results demonstrate that the proposed method can create a better upper bound than the conventional method.

The paper is organized by follows. Section 2 describes the problem definition and task assignment. Section 3 explains the modelling of the problem. Section 4 explains the algorithm of column generation and the heuristic algorithm for generating a feasible solution. In section 5 we provide the computational results of a case study. Section 6 states the summary and conclusions.

2. ROUTE PLANNING PROBLEM FOR AGVS

In this section, we define the route planning problem for AGVs. Consider the situation where multiple AGVs $k \in K$ are traveling in a delivery system which consists of unidirectional arcs. Each AGV $k \in K$ has a uniformly accelerated motion. Each AGV has acceleration and deceleration when stopping and starting. The time to change direction can be ignored. An initial position is assigned to each AGV in advance. The load and unload points of tasks are also statically given in advance. Each AGV can have only one task at the same time. The route planning problem for AGVs is to determine the route plan from an initial position to the loading point, and the route plan from the loading point to the unloading point, with the shortest total delivery time and without collisions among AGVs.

To avoid collisions, we should exclude the situations that multiple AGVs are at the same point at the same time or that an AGV overtakes another AGV by accelerating or decelerating.

First, tasks are assigned to AGVs. A nearest neighbor method is used to assign the tasks (Eda, *et al.*, (2012)). The method is designed so that the task is assigned to the AGV which has the least estimated traveling time. E_{ik}, the estimated completion time of task i by AGV k, can be obtained by the following equation:

$$E_{ik} = T^{k}_{free} + T^{ik}_{\min path} \quad (1)$$

T^{k}_{free} is the estimated completion time in which AGV k completes all assigned tasks without considering collisions with other AGVs. $T^{ik}_{\min path}$ is the minimum traveling time from the unload point of final task of AGV k to the load point of task i, without considering collisions with other AGVs.

3. MATHEMATICAL MODEL

In this section, we explain the mathematical model of the route planning problem. First, we divide the delivery system into regular intervals. Nodes are defined in these areas. The collision avoidance condition should be satisfied in all area. In order to express this problem, we define traveling in those areas by using the following 4 patterns (Fig. 1). Acceleration (i) means that the AGV accelerates and then moves at a constant speed. Deceleration (ii) means that the AGV travels at a constant speed, then decelerates and stops. Constant speed (iii) means that the AGV moves at a constant speed. Acceleration and deceleration (iv) means that the AGV accelerates, moves at a constant speed, decelerates and stops. Additionally, we define 2 actions: wait and task. Wait means that the AGV waits on the same node. Task means that the AGV is performing a load or unload task on the same node. By using these 6 actions, we can create route planning models for AGVs with acceleration and deceleration.

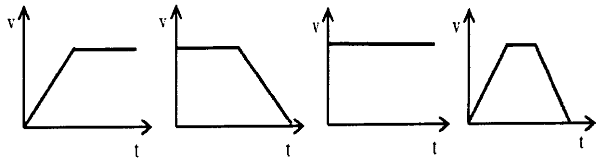

Fig. 1 Definition of AGV actions with acceleration and deceleration

The time necessary for the 4 travel actions is calculated by the following equations. In this research, we consider that AGVs follow uniformly accelerated motion.

Acceleration

$$SP/a + (L - a(SP/a)^2/2)/SP \quad (2)$$

Deceleration

$$SP/d + (L - SP(SP/d) + d(SP/d)^2/2)/SP \quad (3)$$

Constant speed

$$L/SP \quad (4)$$

Acceleration and deceleration

$$SP/a + SP/d + (L - a(SP/a)^2/2 - SP(SP/d) + d(SP/d)^2/2)/SP \quad (5)$$

L is the length of each area. SP is the maximum speed of the AGV. a is acceleration of the AGV. d is the deceleration of the AGV. The time necessary for wait and task can be set freely.

4. COLUMN GENERATION APPROACH

The column generation approach is an effective decomposition method based on the simplex method. This method solves the linear programming problem with huge columns. The derived dual variables are used in the pricing problem. The pricing problem involves deriving a column which has the minimum reduced cost.

4.1 Column generation heuristics

The algorithm of column generation heuristics consists of the following steps:

Step 1: Generate an initial feasible solution
 Generate an initial feasible solution which is conflict-free and create $\overline{R}$ which is a set of AGV routes.
Step 2: Renew dual variables
 Solve the restricted master problem and renew dual variables.

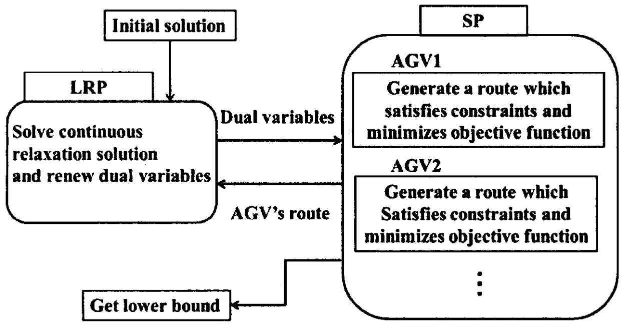

Fig. 2 Outline of column generation approach

Step 3: Solve pricing problem
Solve pricing problem and create a route for each AGV with a minimum reduced cost.
Step 4: Evaluation of convergence
If there are no negative values among the reduced costs of all the AGV routes, go to Step 6. Otherwise go to Step 5.
Step 5: Add columns to the restricted master problem
Add the column generated at Step 3 with the negative reduced cost to $\overline{R}$, and go to Step 2.
Step 6: Derive a lower bound
We obtain a lower bound from the limited set of columns.
Step 7: Use a heuristic algorithm to generate a feasible solution with a limited set of columns. (Generally, branch-and-bound method is used for solving the restricted master problem with integer constraints)

4.2 Generation of an initial feasible solution

In order to execute column generation, we need to generate an initial feasible solution. After the shortest paths are generated by Dijkstra's algorithm for all AGVs to execute assigned task, an initial feasible solution avoiding collision is generated by heuristics (Tanaka et al., 2010). First, we classify the AGV states as traveling, temporary stop and final stop. If there are multiple AGVs in an area at the same time, as change the AGV's route and avoid collision. If the state of one of the AGVs is final stop, we set a temporary destination for it. Otherwise we change and delay the route of AGV that arrived into this area later using the heuristics rule with acceleration and deceleration (Fig. 3). By using the heuristics rule, we change the AGV's action in order to satisfy the speed constraint. We explain the heuristic algorithm which creates a feasible solution with acceleration and deceleration.

In cases (i) and (ii) in Fig. 3, we add the wait action before the acceleration and deceleration, or before the acceleration action. In cases (iii) and (iv), the AGV decelerates. In case (iii), we change acceleration to acceleration and deceleration. Then we change deceleration to acceleration and deceleration. In case (iv), we change constant speed to deceleration, and we change deceleration to acceleration and deceleration. In cases (v) and (vi), the AGV travels with constant speed. In case (v), we change the acceleration to acceleration and deceleration. Then, we change the constant speed to acceleration. In case (vi), we change the constant speed to deceleration and acceleration. In this way, we consider all actions patterns and change the actions for AGVs to satisfy the speed constraint, and we can delay the AGV's movements.

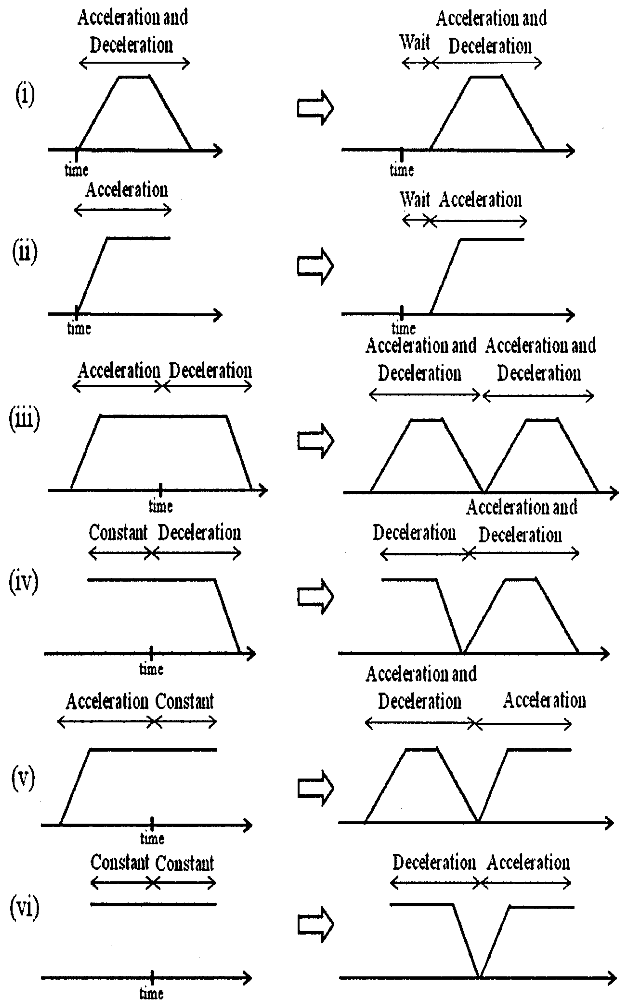

Fig. 3 Steps (i)-(vi) to generate a feasible solution in the heuristics with acceleration and deceleration

4.3 Restricted master problem

The restricted master problem can be formulated as a linear programming problem such as:

Sets

K : the set of AGVs
R^k : the set of possible routes for vehicle $k \in K$
$\overline{R^k}$: the limited set of possible routes for vehicle $k \in K$
N : the set of nodes
T : the set of times delimiting periods
D : the set of tasks

Parameters

c_r : the delivery time generated by route r of AGV $k \in K$, $r \in R^k$

$e_r^{n,t}$: a binary parameter equal to 1 if a vehicle using route r is on node $n \in N$ at time $t \in T$, and to 0 otherwise

τ_r^d : a binary parameter equal to 1 if a vehicle using route r has task $d \in D$, and to 0 otherwise

Δt : the time duration to avoid collisions in each area

Decision variables

$\theta_{r,k}$: binary variable equal to 1 if route $r \in R^k$ is selected by vehicle $k \in K$

Problem formulation

$$\min \sum_{k\in K}\sum_{r\in R^k} c_r \theta_{r,k} \quad (6)$$

This objective function minimizes the total delivery time.

subject to

$$\sum_{k\in K}\sum_{r\in R^k} e_r^{n,\Delta ti}\theta_{r,k} \le 1 \ \ (\forall n \in N, \forall i = 1,2,\ldots,T/\Delta t) \quad (7)$$

This constraint ensures that there is at most one vehicle existing on a node at each time. With this constraint we can avoid collisions among AGVs.

$$\sum_{k\in K}\sum_{r\in R^k} \tau_r^d \theta_{r,k} = 1 \ (\forall d \in D) \quad (8)$$

This constraint ensures that all tasks are executed by AGVs.

$$\sum_{r\in R^k} \theta_{r,k} = 1 \ (\forall k \in K) \quad (9)$$

This constraint enforces the selection of exactly one route for each vehicle.

$$\theta_{r,k} \ge 0 \ (\forall k \in K, \forall r \in \overline{R^k}) \quad (10)$$

This constraint specifies the binary character of the variable $\theta_{r,k}$.

4.4 Pricing problem

We regard the pricing problem as a resource constraint shortest path problem, and a labeling method is applied to solve it. The labeling method is an algorithm for finding a path from source to sink with a minimum cost, with resource constraints. The method stores the information about states such as the cost and the resource consumption as labels. The states and labels are renewed during the search. We define a label as the following. l is a number of tasks.

$$L = (c, s, \tau, h_1, \cdots, h_l, t) \quad (11)$$

L is a label on node n . c is the reduced cost from the start node to node n . s is the speed of AGV on node n at time t . τ is the task number of the AGV at time t . h_i are binary variables equal to 1 if the AGV finishes task i by time t and 0 otherwise. t is the time of this label. A label represents one state from start node to that node. There may be multiple labels at the same node.

4.4.1 Renewing labels

We renew a label $L = (c, s, \tau, h_1, \cdots, h_l, t)$ on node n , and create label $L' = (c', s', \tau', h_1', \cdots, h_l', t')$ on node n' . Node n' can be reached from node n .

The reduced cost c' can be renewed by the following equation:

$$c' = c - \pi_{n'} \quad (12)$$

$\pi_{n'}$ is the cost necessary for the transition from node n to node n' .

If the action is acceleration or constant speed, the speed s' is 1. If the action is deceleration, acceleration and deceleration, wait or task, the speed s' is 0. If the action is task and the task is loading, the task number τ' is the number of the task. If the action is task and the task is unloading, the task number τ' is 0. Otherwise $\tau' = \tau$.

If the action is task and the task is unloading task i , h_i' is 1. Otherwise $h_1' = h_1', \cdots, h_l' = h_l$.

Time t' can be renewed by the following equation:

$$t' = t + t_{transit} \quad (13)$$

$t_{transit}$ is the time the AGV to move in an area.

4.4.2 Dominance

If two labels on node n

$L^1 = (c^1, s^1, \tau^1, h_1^1, \cdots, h_l^1, t^1)$

$L^2 = (c^2, s^2, \tau^2, h_1^2, \cdots, h_l^2, t^2)$

satisfy the following equations,

$$\begin{gathered} c^1 \le c^2, s^1 = s^2, \tau^1 = \tau^2, \\ h_1^1 = h_1^2, \cdots, h_l^1 = h_l^2, t^1 = t^2 \end{gathered} \quad (14)$$

the reduced cost of L^2 cannot be lower than the reduced cost of L^1 in the future. If L^1 and L^2 are not the same, L^1 dominates L^2 , and there is no need to renew L^2 .

Proposition 1

If $c^1 \le c^2, s^1 = s^2, \tau^1 = \tau^2, h_1^1 = h_1^2, \cdots,$ $h_l^1 = h_l^2, t^1 = t^2$ and L^1 and L^2 are not the same, label L^1 dominates label L^2 .

The proof of the proposition 1 is omitted due to space limitations.

4.4.3 Labelling algorithm

Step 1: Generate an initial label

Generate a label on a given initial position of the AGV, and set time $t = 0$ and node number $n = 0$.

Step 2: Renew label

If there is even one label which is not renewed on node n, renew the label to node n' to which the AGV can transit from node n. Generate unrenewed labels on node n'. If the renewing time is over the planning horizon, we generate a renewed label instead of an unrenewed label.

Step 3: Delete label

After renewing labels, we delete the labels dominated by other labels using dominance.

Step 4: Change variables

If $n = |N|$ and there are no unrenewed labels on any of the nodes, go to Step 5. If $n = |N|$ and there are unrenewed labels on any node, set $n = 0$ and go to Step 2. Otherwise set $n = n + 1$ and go to Step 2.

Step 5: Search route

Search a route with the least reduced cost among the routes which are not deleted. If the reduced cost is negative, add the route to the limited column set and the algorithm is finished.

4.5. Column generation heuristics

The solution obtained by the column generation approach is the optimal solution of the continuous relaxation problem, but it may be infeasible. In this research, we propose a heuristic algorithm to generate a feasible solution to provide a good upper bound.

Heuristic algorithm

Step 1: Generate an initial feasible solution

Generate an initial feasible solution by using heuristics.

Step 2: Column Generation

Obtain an optimal continuous relaxation solution

Step 3: Fix a column

Search the highest $\theta_{r,k}$ in the optimal continuous relaxation solution. Fix the column to the AGV.

Step 4: End judgment

If we fixed columns to all AGVs, we finish and we can obtain an upper bound. Otherwise go to Step5.

Step 5: Check feasibility

If it is infeasible, create a feasible solution by using heuristics (see Fig. 3) with fixed AGVs and infeasible AGVs, then return to Step 2.

If the solution is infeasible in the middle of this algorithm, we can generate a feasible solution by using heuristics. By using this algorithm, a good feasible solution can be derived.

5. COMPUTATIONAL RESULTS

In order to investigate the effectiveness of our proposed method, we conducted a simulation on a small-scale transportation system. We coded the program with Microsoft Visual C++ 2008 Express Edition. The branch and bound method with IBM ILOG CPLEX12.1 was used for the solving linear programming problem. An Intel(R) Core(TM) i7 2.80GHz with 3.46GB memory was used for computations.

In this simulation we used the delivery system in Fig. 4. The number of nodes in this delivery system is 8. We set the length of each area is 10m, and we set the information of AGVs based on actual information. The number of AGVs is from 2 to 6 and the number of tasks is same as AGV number. We compare the upper bound and computational time of the Proposed Method (PM) and branch and bound method (BB). After the column generation method converges, we use BB to the limited set of routes. The computational results are shown in the Table 1.

Table 1 Computational results

Method	AGV num.	LB	UB	Time(s)
BB	2	34.3	34.3	1.1
	3	44.1	44.1	2.6
	4	75.8	84.1	17.8
	5	97.5	101.9	47.4
	6	123.3	147.7	84
PM	2	34.3	34.3	1
	3	44.1	44.1	2.6
	4	75.8	84.1	17.8
	5	97.5	99.8	58.6
	6	123.3	132.9	99.3

From these computational results, we can confirm that the proposed column generation heuristic (PM) can generate better upper bounds than those of BB. This is because PM can generate more columns to create a feasible solution than BB. Our proposed method takes more computational time than BB, because PM fixes the routes one by one. If the number of AGVs is small, we can obtain the same results for UB and LB. It means that an optimal solution can be obtained by the proposed algorithm. If the number of AGVs is large, the accuracy of the UB is declines. If there are many AGVs in the delivery system, we have to consider the interference among them. Therefore, it becomes difficult to generate UB with good accuracy.

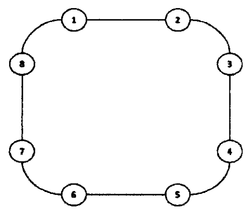

Fig. 4 Case study of a delivery system

6. SUMMARY AND CONCLUSION

In this paper, we proposed an efficient column generation approach for solving the route planning problem for AGVs with acceleration and deceleration. In the pricing problem, we applied a labeling algorithm using a dominance relation. In the restricted master problem, the collision avoidance constraint was considered in the case of acceleration and deceleration. From the computational results, we demonstrated that our method can generate better upper bound than the conventional column generation method with BB.

REFERENCES

Bunte, S. and N. Kliewer. (2009). "An overview on vehicle scheduling models", Public Transport, Vol. 1, pp. 299–317.

Bettinelli, A., A. Ceselli, and D. Righini. (2011). "A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows", Transportation Research - Part C: Emerging Technologies, Vol. 19, No. 5, pp. 723–740.

Desaulniers, G., A. Langevin, and D. Riopel. (2003). "Dispatching and Conflict-free Routing of Automated Guided Vehicles: An Exact Approach", International Journal of Flexible Manufacturing Systems, Vol. 15, pp. 309–331.

Eda, S., T. Nishi, T. Mariyama, S. Kataoka, K. Shoda, and K. Matsumura. (2012). "Petri Net Decomposition Approach for Bi-Objective Routing for AGV systems Minimizing Total Traveling Time and Equalizing Delivery Time", Journal of Advances Mechanical Design, Systems, and Manufacturing, Vol. 6, No. 5, pp. 672–686.

Feillet, D., P. Dejax, M. Gendreau, and C. Gueguen. (2004). "An Exact Algorithm for the Elementary Shortest Path Problem with Resource Constraints: Application to Some Vehicle Routing Problems", Networks, Vol. 44, No. 3, pp. 216–229.

Le-Anh, T. and M.B.M. De Koster. (2006). "A review of design and control of automated guided vehicle systems", European Journal of Operational Research, Vol. 171, pp. 1–23.

Nishi, T., M. Ando, and M. Konishi. (2005). "Distributed Route Planning for Multiple Mobile Robots using an Augmented Lagrangian Decomposition Technique", IEEE Transactions on Robotics, Vol. 22, No. 6, pp. 1191–1200.

Reveliotis, S. A. and E. Roszkowska. (2011). "Conflict Resolution in Free-Ranging Multivehicle Systems: A Resource Allocation Paradigm", IEEE Transactions on Robotics, Vol. 27, No. 2, pp. 283–296.

Tanaka, Y., T. Nishi, and M. Inuiguchi. (2010). "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.