Conflict-Free Routing of AGVs on the Mesh Topology Based on a Discrete-Time Model

Zeng Jianyang
zengjy321@pmail.ntu.edu.sg

Wen-Jing Hsu
hsu@ntu.edu.sg

Center for Advanced Information Systems
School of Computer Engineering
Nanyang Technological University, Singapore 639798

Abstract:

Automated Guided Vehicles (or AGVs for short) have become an important option in material handling. In many applications, such as container terminals, the service area is often arranged into rectangular blocks, which leads to a mesh-like path topology. Therefore, developing efficient algorithms for AGV routing on the mesh topology has become an important research topic. In this paper, we present a discrete time model, based on which a simple routing algorithm on the mesh topology is presented. The algorithm works by carefully choosing suitable parameters such that the vehicles using a same junction will arrive at different points in time, and hence no conflicts will occur during the routing; meanwhile, high routing performance can be achieved. Analyses of the task completion time and the requirements on timing control during the AGV routing are also presented.

1. Introduction:

Automated Guided Vehicles (or AGVs for short) have become an important option in material handling [1-7, 9-11]. In many applications, such as container terminals[1, 9-11], the service area is often arranged into rectangular blocks, which leads to a mesh-like path topology. Therefore, developing efficient algorithms for AGV routing on this topology has become an important research topic.

There are many existing results about AGV [5]. However, relatively little has been known about routing on the mesh topology. [2-3] gave the analysis of time and space complexities for some basic AGV routing operations on 2D-mesh topology. The upper bounds of time and space complexities for AGV routing are $\Theta(n^2)$ and $\Theta(n^3)$ respectively, where n denotes the number of nodes in the path topology. However, the paper does not give the details of the routing algorithms and techniques to avoid congestion, conflicts, deadlocks, etc.

[6-7] presented different methods to schedule and route simultaneously in an $n \times n$ mesh-like path topology. The algorithms can schedule and route simultaneously up to $4n^2$ AGVs concurrently at one time. In these papers, the routing process is formulated as a sorting problem. Although there are no conflicts during the permutation, it requires $3n$ steps of well-defined physical moves, which requires AGVs to travel extra distance and consume extra energy to finish the tasks.

In this paper, we present a discrete time model on mesh topology for AGV routing. Based on this model, the routing algorithm is presented and time control requirement is analyzed. The key idea lies in making use of the regularity of the mesh, and hence the regularity of points of time when AGVs arrive at the intersections. By choosing a suitable speed for the AGVs along different directions, we can ensure that no conflicts among any AGVs will occur. We also analyze the algorithm in terms of bounds on the task completion time and requirements on the routing precision controls. By our design, the AGVs can advance directly to their destinations, unlike in [6-7], and hence high performance can also be ensured.

The remainer of the paper is organized as follows. Section 2 describes the mesh layout and the routing model. Section 3 gives the routing algorithm and the time control criteria to avoid conflicts. In Section 4, we analyze the performance of the routing algorithm. Section 5 gives an explicit method to derive the timing controls. Finally, Section 6 discusses possibilities of relaxing certain constraints and points out directions of future study.

2. Description of mesh layout model

In order to describe the marrow of our routing process clearly, we start with a simple but general model in which one yard block has only one station near an intersection of pathways (refer to Figure 1). In this mesh

layout, there are in total $N \times N$ blocks, namely N blocks in each column and N blocks in each row. Each block has the same size. There are two paths with different directions between two adjacent blocks. Each Block has one Pick up-Drop off station(or P/D station for short), located at the upper right and upper top corner of the block. On the left-top side, there is a vehicle park where all AGVs are stationed initially and to which they will return upon completion of all tasks.

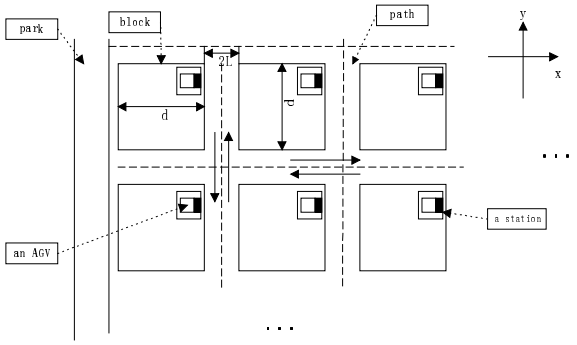

Figure 1. Realistic mesh layout

Although there are some important details for AGV routing, such as the size of the junction, the radius of 90° turn, the length of the AGV, etc[4-7], it is reasonable and realistic for us to simplify the mesh model for convenience of analysis and discussion. In the simplified mesh layout model, shown as Figure 2, there are N^2 junctions of pathways. A junction and the associated neighboring station are collectively regarded as a node. Each node is to assign it with the coordinates (x,y) as its address or ID, where x and y represent respectively the row and column IDs. This mesh layout is modeled by a graph $G=(V,E)$. The $N \times N$ vertices of the graph represent junction nodes, and the bi-directional edges represent two paths between two adjacent junction nodes, and the length of each edge is a constant.

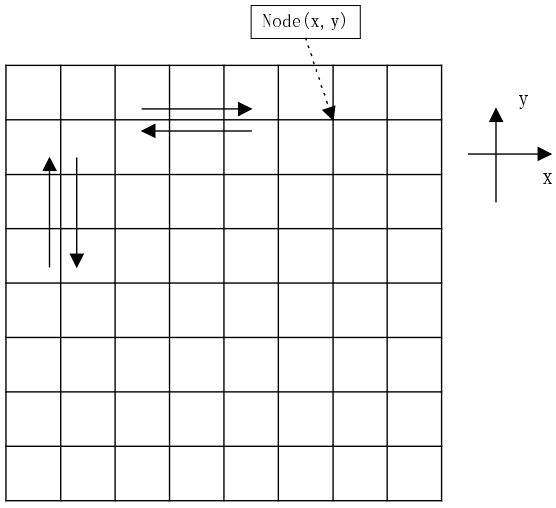

Figure 2. Simplified mesh routing model

We divide the AGV movements into three phases. In the first phase, let AGVs set out from the park to their pick up stations. In the second phase, let AGVs pick up loads and travel to their destinations and drop-off loads. In the third phase, let AGVs return to the park from their drop-off stations. Because it is easy for us to dispatch the AGV moving without any conflict in the first phase and the third phase, we can focus only on the second phase when the loaded AGVs move on the mesh layout.

We assume that the time can be divided into discrete units of time, and that each AGV always reaches every junction node at some discrete point on time. It is reasonable for us to make this assumption because the distance between two adjacent nodes is a constant, and we can adjust the speed of the AGVs to let them arrive at the junctions at multiples of the unit time.

We assume that when an AGV reaches its destination, it enters the buffer and leaves the mesh grid.

Based on the mesh layout model, we formally define the following.

Definition: *$(x_1,y_1)=(x_2,y_2)$ if and only if $x_1=x_2$ and $y_1=y_2$.*

Definition (Job): *A job is identified by an ordered pair ((PX,PY),(DX,DY)), where(PX,PY) represenst the address of the pickup station, (DX,DY) represents the address of the drop-off station, and $(PX,PY) \neq (DX,DY)$.*

Assume that each job has a distinct origin and also a distinct(but different) destination, and an AGV is given only one job and any job is assigned to only one AGV.

Definition (Job Set): *A job set M denoting a set of k jobs, where $2 \le k \le \left\lfloor \frac{N^2}{2} \right\rfloor$, is defined as follows:*

$M=\{((PX_i, PY_i), (DX_i, DY_i))| 1 \le PX_i, PY_i, DX_i, DY_i \le N,$ for $i=1, 2, \ldots, k\}$.

According to the position of the origins and destinations of jobs, any given job set M can be divided into four subsets, denoted by M_x, M_y, M_{xy} respectively, such that

$$M_x=\{((PX_i,PY_i),(DX_i,DY_i))|DX_i \neq PX_i, DY_i = PY_i \text{ for } i=1,2,...,k\}.$$

$$M_y=\{((PX_i,PY_i),(DX_i,DY_i))|DY_i \neq PY, DX_i = PX_i, \text{ for } i=1,2,...,k\}.$$

$$M_{xy}=\{((PX_i,PY_i),(DX_i,DY_i))|DY_i \neq PY, DX_i \neq PX_i, \text{ for } i=1,2,...,k\}.$$

We also divide M_{xy} into two subsets, denoted by M_{xy+} and M_{xy-}, such that

$M_{xy+} = \{((PX_i, PY_i),(DX_i, DY_i)) | DY_i > PY, DX_i \neq PX_i,$
$for \ \ i = 1,2,...,k\}$.

$M_{xy-} = \{((PX_i, PY_i),(DX_i, DY_i)) | DY_i < PY, DX_i \neq PX_i,$
$for \ \ i = 1,2,...,k\}$.

Accordingly, we have the following notations:

A_x: the set of AGVs that carry out jobs in M_x;

A_y: the set of AGVs that carry out jobs in M_y;

A_{xy+}: the set of AGVs that carry out jobs in M_{xy+};

A_{xy-}: the set of AGVs that carry out jobs in M_{xy-};

Definition (Direction of AGVs): *$\vec{v}$ is a unit vector which represents the direction of a given AGV, where $\vec{v} \in \{+\vec{x}, -\vec{x}, +\vec{y}, -\vec{y}\}$. $\vec{v}_1 = \vec{v}_2$ when $\vec{v}_1$ is in the same direction as $\vec{v}_2$. $\vec{v}_1 = -\vec{v}_2$ when $\vec{v}_1$ is in the opposite direction of $\vec{v}_2$. $\vec{v}_1 \cdot \vec{v}_2 = 0$ when $\vec{v}_1$ is in a vertical direction of $\vec{v}_2$.*

Definition (State of AGVs): *$((x,y),t,\vec{v})$ is the state of an AGV, where (x,y) represents the location in the mesh layout, and t represents the discrete time points of the AGV, and $\vec{v} \in \{+\vec{x}, -\vec{x}, +\vec{y}, -\vec{y}\}$.*

Definition (Collision): *$((x_1,y_1),t_1,\vec{v}_1)$ is the state of AGV1, and $((x_2,y_2),t_2,\vec{v}_2)$ is the status of AGV2. When $t_1 = t_2$, $(x_1,y_1) = (x_2,y_2)$ and $\vec{v}_1 \in \{+\vec{x}, -\vec{x}, +\vec{y}, -\vec{y}\} - \{-\vec{v}_2\}$. In this case, we say that AGV1 and AGV2 have a collision at (x_1,y_1)or(x_2,y_2) when $t = t_1$ on the mesh layout.*

3. Conflict-free routing algorithm

Based on the simplified mesh layout and the discrete time, the routing algorithm is presented as follows.

Let all AGVs set out from their pick up stations at the same time, when $t_0 = 0$.

Case a In the job set M_x. In this case, let the AGV travel along the row PY_i from (PX_i, PY_i) station to (DX_i, DY_i).

Case b In the job set M_y. In this case, let the AGV travel along the column PX_i from (PX_i, PY_i) station to (DX_i, DY_i).

Case c In the job set M_{xy+}. In this case, let the AGV firstly travel along the column PX_i from (PX_i, PY_i) station to (DX_i, DY_i) station. Then let it travel along the row DY_i from (DX_i, PY_i) station to (DX_i, DY_i) station.

Case d In the job set M_{xy-}. In this case, let the AGV firstly travel along the column PX_i from (PX_i, PY_i) station to (DX_i, DY_i). Then let it travel along the row DY_i from (DX_i, PY_i) station to (DX_i, DY_i) station.

The routing algorithm looks simple, and if we let AGVs travel in this rule at an arbitrary speed, it is very likely to have collisions on the mesh layout. However, as we will show shortly, if we control the time when each AGV reaches every junction node(we can do so by controlling the AGV's speed), AGVs can run on the mesh layout with the freedom of conflicts.

We let ΔT_{+x} denote the time required for an AGV to travel through one edge of the mesh along the $+\vec{x}$ direction. Let ΔT_{+x} (ΔT_{+y} , ΔT_{-y}) be defined similarly. We assume that AGVs travel at the speed v_{+x}, v_{-x}, v_{+y}, v_{-y} in these four cases respectively.

According to the preceding routing, we have the following conclusions.

Claim 3.1: *According to our routing algorithm, there is no conflict between any two AGVs belonging to the same set A_x (or A_y).*

[Proof]:
According to the definition of collision, and the assumption that each AGV has a distinct origin, it is quite clear that there is no conflict in the AGVs belonging to A_x (or A_y). □

Claim 3.2: *Based on the routing algorithm, any AGV will not run into conflict with other AGVs on the mesh layout, if the following relation is satisfied.*

$$(1)\ \frac{lcm(\Delta T_1, \Delta T_2)}{max(\Delta T_1, \Delta T_2)} \geq N \qquad (3\text{-}1)$$

where ΔT_1 and ΔT_2 are any two permutations from $\{\Delta T_{+x}, \Delta T_{-x}, \Delta T_{+y}, \Delta T_{-y}\}$.

$$(2)\begin{cases} gcd(\Delta T_{+y}, \Delta T_{+x}) \neq \Delta T_{-y} & (3\text{-}2) \\ gcd(\Delta T_{+y}, \Delta T_{-x}) \neq \Delta T_{-y} & (3\text{-}3) \\ gcd(\Delta T_{-y}, \Delta T_{+x}) \neq \Delta T_{+y} & (3\text{-}4) \\ gcd(\Delta T_{-y}, \Delta T_{-x}) \neq \Delta T_{+y} & (3\text{-}5) \\ gcd(\Delta T_{\pm y}, \Delta T_{\pm x}) \geq N & (3\text{-}6) \end{cases}$$

here gcd the Greatest Common Divisor, and lcm the Least Common Multiple.(cf. Definition A.2 and Definition A.3).

[Proof]:
Firstly let us recall some definitions and theorems of number theory[8] which will be used in the discussions later.

Definition A.1 (Divisibility): *If a and b are integers, we say that a divides b if there is an integer c such that b=ac. If a divides b, we also say that a is a divisor or factor of b.*
Write $a|b$ if a divide b; otherwise, write $a \nmid b$ if a does not divide b.

Definition A.2 (The Greatest Common Divisor): *The greatest common divisor of two integers a and b, not both zero, is the largest positive integer that divides both a and b; it is denoted by gcd(a,b).*

Definition A.3 (The Least Common Multiple): *The least common multiple (gcd) of two integers a and b, is the least positive integer divisible by both a and b; it is denoted by lcm(a,b).*

Definition A.4 (Linear combination): *If a and b are integers, then a linear combination of a and b is a sum of the form ma+nb, where both m and n are integers.*

Definition A.5 (Linear diophantine equation): *A linear diophantine equation in two variables x and y is a diophantine equation of the form ax+by=c, where a, b and c are integers.*

Theorem A.1: *The greatest common divisor of the integers a and b, that are not both zero, is the least positive integer that is a linear combination of a and b.*
Theorem A.2: *Let a and b be positive integers. Then* $lcm(a,b) = \frac{ab}{gcd(a,b)}$.
Theorem A.3: *Let a and b be positive integers and d=gcd(a,b). The equation ax+by=c has no integer solutions if $d \nmid c$. If d|c, then there are infinitely many integer solutions. Moreover, if $x = x_0$, $y = y_0$ is a particular solution of the equation, then all solutions are given by*

$$\begin{cases} x = x_0 + (b/d)n \\ y = y_0 - (a/d)n \end{cases}$$

Now continue with the proof of the claim 3.1. All cases of possible conflicts are shown in Figure 3. We omitted a few similar cases, which are symmetrical to some of the following cases).

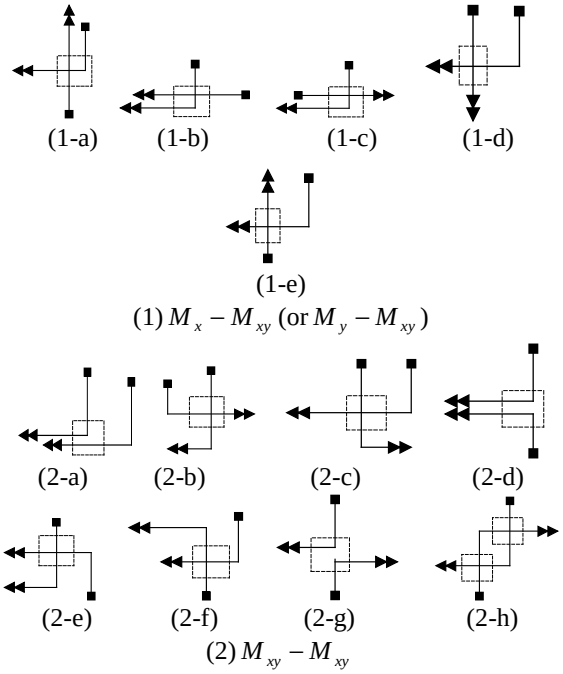

Figure 3. All cases of possible conflicts

Assume the initial states of AGV1 and AGV2 respectively as follows.
AGV1: $((x_1, y_1), t_1 = 0, \vec{v}_1)$;
AGV2: $((x_2, y_2), t_2 = 0, \vec{v}_2)$.

When $(x_1, y_1) = (x_2, y_2) = (x', y')$, the states of AGV1 and AGV2 are respectively,
AGV1: $((x_1', y_1'), t_1', \vec{v}_1')$;
AGV2: $((x_2', y_2'), t_2', \vec{v}_2')$.

Now let us prove that $t_1' \neq t_2'$ in all cases of potential conflicts.

Case (1) This case covers (1-a), (1-b), (1-c), (2-d), (2-g). We have the following relations.

$$\begin{cases} t_1' = t_1 + i\Delta T = i\Delta T, \\ t_2' = t_2 + j\Delta T = j\Delta T, \end{cases}$$

where $0 \le i, j \le N-1$, and ΔT_1 and ΔT_2 are any two permutations from $\{\Delta T_{+x}, \Delta T_{-x}, \Delta T_{+y}, \Delta T_{-y}\}$.

According to the definition of lcm, we have

$$\begin{cases} i_{min} = \frac{lcm(\Delta T_1, \Delta T_2)}{\Delta T_1} \\ j_{min} = \frac{lcm(\Delta T_1, \Delta T_2)}{\Delta T_2} \end{cases}$$

According to Inequality (3-1), we know $i_{min}, j_{min} \ge N$, which conflict with the condition such that $0 \le i, j \le N-1$. Therefore in all these cases, $t_1' \neq t_2'$ for any *i, j, where* $0 \le i, j \le N-1$.

Case (2) This case covers (1-e), (2-a), (2-b), (2-c), (2-e), (2-f), (2-h). We have the following relations.

$$\begin{cases} t_1' = i\Delta T_{+y} + j\Delta T_{+x}, \\ t_2' = k\Delta T_{-y}, \end{cases}$$

or

$$\begin{cases} t_1' = i\Delta T_{+y} + j\Delta T_{-x}, \\ t_2' = k\Delta T_{-y}, \end{cases}$$

or

$$\begin{cases} t_1' = i\Delta T_{-y} + j\Delta T_{+x}, \\ t_2' = k\Delta T_{+y}, \end{cases}$$

or

$$\begin{cases} t_1' = i\Delta T_{-y} + j\Delta T_{-x}, \\ t_2' = k\Delta T_{+y}, \end{cases}$$

where $0 \le i, j, k \le N-1$.

These four relations are similar to each other, so we can focus on the first one.

Firstly we take a look at the following equation.

$$x\Delta T_{+y} + y\Delta T_{+x} = k\Delta T_{-y} \qquad (3\text{-}7)$$

where x and y are integers.

From Eq. (3-6) and Eq. (3-2), we have

$gcd(\Delta T_{+y}, \Delta T_{+x}) \nmid k\Delta T_{-y}$

According to Theorem A.3, we know that Eq. (3-7) has no integer solutions, so for any $i, j \in [0, N]$, $t_1' \neq t_2'$.

Case (3) This case covers (1-d). We have the following relation.

$$\begin{cases} t_1' = i\Delta T_{+y} + j\Delta T_{+x}, \\ t_2' = k\Delta T_{+y}, \end{cases}$$

In order to prove that $t_1' \neq t_2'$, we need to show that

$$i\Delta T_{+y} + j\Delta T_{+x} \neq k\Delta T_{+y},$$

namely, $j\Delta T_{+x} \neq |k - j|\Delta T_{+y}$.

We know that $0 \le |k - j| \le N-1$, then this situation can be converted to the one in case (1) that we have proved.

Therefore, we conclude that in all cases, there is no any conflict using our routing algorithm and the time limit.

4. Time requirement

Claim 4.1: *The time requirement T_r for all AGVs to transport all jobs is upper-bounded by*

$$2(N-1)max\{\Delta T_{+x}, \Delta T_{-x}, \Delta T_{+y}, \Delta T_{-y}\}$$

[Proof]: Since all jobs are carried out in parallel, the time requirement for a job set *M* is determined by the most time-consuming job in the set. Formally, for any given job set, we have

$$T_r = max\{T((PX_1, PY_1), (DX_1, DY_1)), T((PX_2, PY_2), (DX_2, DY_2)), \ldots, T((PX_k, PY_k), (DX_k, DY_k))\},$$

where $T((PX_i, PY_i), (DX_i, DY_i))$ is the time requirement for AGVi to complete its job.

Assume that there exists job $((1,1),(N,N))$ which uses the most time and use $max\{\Delta T_{+x}, \Delta T_{-x}, \Delta T_{+y}, \Delta T_{-y}\}$ time to go through one edge on the mesh layout, then we obtain the following relation.

$$T((1,1),(N,N)) = 2(N-1)max\{\Delta T_{+x}, \Delta T_{-x}, \Delta T_{+y}, \Delta T_{-y}\}$$

Thus, the time requirement for a job set is upper-bounded by

$$T_r \le T((1,1),(N,N)) = 2(N-1)max\{\Delta T_{+x}, \Delta T_{-x}, \Delta T_{+y}, \Delta T_{-y}\}$$ □

Although our routing algorithm guarantees collision-freedom under some special criteria, the control system needs to know the time point when each AGV goes through every junction node. So it is necessary for us to consider the relation between different time points when each AGV goes through every junction node.

Definition (Time difference): The time difference is the minus of two time points when two AGVs reach one special junction node. The minimum time difference is the minimum time difference for all AGVs at every junction node on the mesh layout.

Claim 4.2: T*he minimum time difference on the mesh layout is lower-bounded by.*

$$min\{gcd(\Delta T_{+x}, \Delta T_{-x}), gcd(\Delta T_{+x}, \Delta T_{+y}), gcd(\Delta T_{+x}, \Delta T_{-y}), gcd(\Delta T_{-x}, \Delta T_{+y}), gcd(\Delta T_{-x}, \Delta T_{-y}), gcd(\Delta T_{+y}, \Delta T_{-y})\}$$

namely,

$$\min_{\substack{x,y \in S \\ x \neq y}} \{gcd(x, y)\}$$

where $S = \{\Delta T_{+x}, \Delta T_{+y}, \Delta T_{-x}, \Delta T_{-y}\}$.

[Proof]:

To get the minimum time difference, we can find the minimum of the following equation:

$$i\Delta T_1 + j\Delta T_2$$

where *i*, *j* are integers, and ΔT_1 , ΔT_2 are any two numbers from $\{\Delta T_{+x}, \Delta T_{-x}, \Delta T_{+y}, \Delta T_{-y}\}$.

According to Theorem A.1, the least positive integer of $i\Delta T_1 + j\Delta T_2$ is $gcd(\Delta T_1, \Delta T_2)$.

Thus for any ΔT_1 , ΔT_2 from $\{\Delta T_{+x}, \Delta T_{-x}, \Delta T_{+y}, \Delta T_{-y}\}$, the minimum of the time difference is

$$min\{gcd(\Delta T_{+x}, \Delta T_{-x}), gcd(\Delta T_{+x}, \Delta T_{+y}), gcd(\Delta T_{+x}, \Delta T_{-y}),$$

$gcd(\Delta T_{-x},\Delta T_{+y}),gcd(\Delta T_{-x},\Delta T_{-y}),gcd(\Delta T_{+y},\Delta T_{-y})\}$

namely

$$\min_{\substack{x,y\in S\\ x\neq y}}\{gcd(x,y)\}$$

where $S=\{\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}\}$ □

5. The method to construct $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$

We introduce the following method to construct $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$, which satisfy the criteria of conflict-free routing.

We let $\Delta T_{+y}=P_1^{a+b}P_5^g$, $\Delta T_{-y}=P_2^{c+d}P_5^g$, $\Delta T_{-x}=P_1^aP_2^cP_3^e$ and $\Delta T_{+x}=P_1^aP_2^cP_4^f$

Where P_1, P_2, P_3 and P_4 are primes;

$a,b\geq log_{P_1}N$;

$c,d\geq log_{P_2}N$;

$e\geq log_{P_3}N$;

$f\geq log_{P_4}N$;

$g\geq log_{P_5}N$.

Claim 5.2: *The values of $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$ constructed by this method satisfy the criteria of conflict-free routing.*

[Proof]:

According to Theorem A.2, we have

$$\frac{lcm(\Delta T_{+y},\Delta T_{-y})}{\Delta T_{+y}}=\frac{\Delta T_{+y}\cdot\Delta T_{-y}}{gcd(\Delta T_{+y},\Delta T_{-y})\cdot\Delta T_{+y}}$$

$$=\frac{\Delta T_{-y}}{gcd(\Delta T_{+y},\Delta T_{-y})}=\frac{P_2^{c+d}P_5^g}{P_5^g}$$

$$=P_2^{c+d}\geq N^2\geq N$$

Similarly we can prove that for *any two permutations ΔT_1 and ΔT_2 from $\{\Delta T_{+x}$, ΔT_{-x} , ΔT_{+y} , $\Delta T_{-y}\}$, the following relations are satisfied.*

$$\frac{lcm(\Delta T_1,\Delta T_2)}{\Delta T_1}\geq N\ .$$

According to the construction method, we have

$$gcd(\Delta T_{+y},\Delta T_{+x})=gcd(P_1^{a+b}P_5^g,P_1^aP_2^cP_4^f)=P_1^a\geq N$$

Because $P_1^a\neq P_2^{c+d}P_5^g$, we have

$$gcd(\Delta T_{+y},\Delta T_{+x})\neq\Delta T_{-y}\ .$$

Similarly we can prove that the inequalities (3-3)-(3-6) are satisfied.

Therefore the values of $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$ constructed by this method satisfy all the criteria of conflict-free routing. □

The differences in $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$ have implications on the routing control. The smaller value of this difference generally means the more accurate timing when vehicles arrive at the junctions.

Claim 5.2: *The minimum time difference of $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$ constructed by this method $min\{P_1^a,P_2^c,P_5^g\}$,which is lower-bounded by $\Omega(N)$.*

[Proof]:

According to Claim 4.2, the minimum time difference on the mesh layout is lower-bounded by

$min\{gcd(\Delta T_{+x},\Delta T_{-x}),gcd(\Delta T_{+x},\Delta T_{+y}),gcd(\Delta T_{+x},\Delta T_{-y}),$
$gcd(\Delta T_{-x},\Delta T_{+y}),gcd(\Delta T_{-x},\Delta T_{-y}),gcd(\Delta T_{+y},\Delta T_{-y})\}$

Substituting into the values of $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$, we have

$min\{gcd(\Delta T_{+x},\Delta T_{-x}),gcd(\Delta T_{+x},\Delta T_{+y}),gcd(\Delta T_{+x},\Delta T_{-y}),$
$gcd(\Delta T_{-x},\Delta T_{+y}),gcd(\Delta T_{-x},\Delta T_{-y}),gcd(\Delta T_{+y},\Delta T_{-y})\}$
$=min\{P_1^a\cdot P_2^c,P_1^a,P_2^c,P_1^a,P_2^c,P_5^g\}=min\{P_1^a,P_2^c,P_5^g\}$

Therefore, the minimum time difference of $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$ constructed by this method is lower-bounded by $min\{P_1^a,P_2^c,P_5^g\}$. Because $P_1^a,P_2^c,P_5^g\geq N$, the minimum time difference is lower-bounded by $\Omega(N)$. □

From claim 4.1, the longest value of $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$ generally means a higher task completion time(for the given choice of time unit). The following result bounds this value.

Claim 5.3: *The values of $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$ are bounded by $\Theta(N^3)$.*

[Proof]:

According to the construction method, we have

$$\Delta T_{+y}=P_1^{a+b}P_5^g\geq N^2\cdot N=N^3$$

Similarly, we can prove that $\Delta T_{-y}\geq N^3$, $\Delta T_{+x}\geq N^3$, $\Delta T_{-x}\geq N^3$.

If we keep the values of $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$ as small as possible, we can make them be bounded by $\Theta(N^3)$. Therefore, the values of $\Delta T_{+x},\Delta T_{+y},\Delta T_{-x},\Delta T_{-y}$ is bounded by $\Theta(N^3)$. □

We give a simple example to illustrate the construction. Let N=7, we can choose $\Delta T_{+y}=2^{3+3}\cdot 13$,

$\Delta T_{-y} = 3^{2+2} \cdot 13$, $\Delta T_{-x} = 2^3 \cdot 3^2 \cdot 7$, $\Delta T_{+x} = 2^3 \cdot 3^2 \cdot 11$.

The minimum time difference of this case is $2^3 = 8$, and the ratio between the maximum and the minimum of $\Delta T_{+x}, \Delta T_{+y}, \Delta T_{-x}, \Delta T_{-y}$ is about 2.

6. Discussions and conclusions

We have presented a discrete time model for AGV routing on the mesh topology. In this model, each AGV is assumed to reach every junction node at discrete points in time. Based on this model, we proposed a routing algorithm which allows AGVs to travel at different multiples of the unit time along different directions, which guarantees the freedom of conflicts. The timing control requirement was analyzed, and the method to construct the multiples of the unit time was also introduced.

With our routing algorithm, all the AGVs can move directly towards their destinations without conflicts. Therefore, the overall routing performance is ensured. Moreover, since each AGV makes at most one turn during the entire routing process, the speed of each AGVs is changed no more than once. Therefore, the energy requirement by our routing algorithm is also relatively low. In our routing model, each AGV on the mesh topology is assumed to be one point. However, there are some details that we must consider in actual implementations, such as the size of junction, the length of the AGV, etc. These considerations have a requirement of the minimum time difference, which can be adjusted by the control system. According to Claim 5.2, the minimum time difference is decided by $min\{ P_1^a, P_2^c, P_5^g \}$. Thus we can choose the value of $\{ P_1^a, P_2^c, P_5^g \}$ to increase the minimum time difference. Therefore, the discrete time model and the routing algorithm can be used in real mesh-like layout. Similarly, the task completion time can be controlled by choice of the units of time, the distance between intersections and/or the speeds of AGVs. For instance, by Claim 5.3, the maximum value of $\Delta T_{+x}, \Delta T_{+y}, \Delta T_{-x}, \Delta T_{-y}$ is bounded by $\Theta(N^3)$. As N increases, the time requirement to finish the jobs seems to increase quickly. However, we can choose a small unit of time to keep the actual time requirement low, as long as the minimum time difference for avoiding conflicts is satisfied.

We assumed that when an AGV reaches its destination, it enters the buffer and leaves the mesh grid. This assumption can be also relaxed. Usually, when an AGV enters the buffer of the P/D station, it takes some time for the vehicle to completely leave the mesh grid. The situation is similar when an AGV goes through the junction. However, as long as the time required for an AGV to enter the buffer of the station is less than the minimum time difference, there are still no conflicts during the AGV routing.

We assume that each AGV has distinct origin and also a distinct (but different) destination, namely, the pattern of AGV movement is permutation. This assumption can also be relaxed such that each AGV has multiple origins and multiple destinations. As long as we control the time points for AGVs to set out from the stations and let them enter the buffers at proper time point, we can still guarantee the freedom of conflicts.

There are many open issues for future research. Firstly, how to deal with the failure of AGVs? In our algorithm (as well as others), a single blockage will cause the failure of the whole system. Therefore, it is essential to consider fault-tolerant strategies. Secondly, if we allow each AGV to make more than one turn before it reaches its destination, we need more complicated scheme to avoid conflicts. Thirdly, we need to devise a method to decide the number of AGVs for the given jobs and to deal with idle AGVs.

Acknowledgement

We acknowledge the Maritime and Port Authority, A*STAR and Nanyang Technological University, all of Singapore, for their support of the research project.

References

[1] Evers, J. J. M. and S. A. J. Koppers. Automatic guided vehicle traffic control at a container terminal. *Transportation Research Part A*, 30(1):21-34,1996.

[2] HSU, W.-J. and HUANG, S.-Y., 1994, Route planning of automated guided vehicles. Proceedings *of Intelligent Vehicles*, Paris, pp.479-485.

[3] Huang, S.-Y. and W.-J. Hsu. Routing automated guided vehicles on mesh like topologies. In *Proceedings of International Conference on Automation, Robotics and Computer Vision*, 1994.

[4] Qiu, L. and W.-J. Hsu, A bi-directional path layout for conflict-free routing of AGVs. *International Journal of Production Research*, 39(10): 2177-2195, 2001.

[5] Qiu, Ling, Wen-Jing Hsu, Shell-Ying Huang, and Han Wang, "Scheduling and Routing Algorithms for AGVs: a Survey". *International Journal of Production Research*, Vol. 40, No. 3, pp. 745-760, 2002.

[6] Qiu, Ling and Wen-Jing Hsu, "Routing AGVs on a Mesh-like Path Topology". In *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (IVS*

2000), pp. 392-397, Dearborn, Michigan, USA, Oct. 3-5, 2000.

[7] Qiu, Ling and Wen-Jing Hsu, "Algorithms for Routing AGVs on a Mesh Topology". In *Proceedings of the 2000 European Conference on Parallel Computing (Euro-par 2000)*, pp. 595-599, Technical University of Munich, Munich, Germany, Aug. 29-Sep. 1, 2000.

[8] Thomsa Koshy, Elementary Number Theory with Applications. *Harcourt/Academic Press*, 2002.

[9] Ye, R., W.-J. Hsu, and V.-Y. Vee. Distributed routing and simulation of automated guided vehicles. In *Proceedings of TENCON 2000*, volume II, pages 315-320, Kuala Lumpur, Malaysia, September 24-27, 2000.

[10] Ye, R., V.-Y. Vee, W.-J. Hsu and S.N. Shah. Parallel simulation of AGVs in container port operations. In *Proceedings of 4th International Conference/Exhibition on High Performance Computing in Asia-pacific Region (HPC-ASIA 2000)*, volume I, pages 1058-1063, Beijing, China, May 14-17, 2000.

[11] Yu, X. and S.-Y. Huang, A Centralized Routing Algorithm for AGVS in Container Ports. In *Proceedings of the 4th International Conference on Computer Integrated Manufacturing*, Singapore, pages 589-600, 1997.