An AGV-Routing Algorithm in the Mesh Topology with Random Partial Permutation

Zeng Jianyang, Hsu Wen-Jing and Vee Voon Yee
Centre for Advanced Information Systems, School of Computer Engineering
Nanyang Technological University, Singapore 639798
{pg03858494, hsu, ASVYVEE}@ntu.edu.sg

Abstract

In this paper, we model a realistic AGV system by a multi robots system with mesh layout. Based on certain reasonable assumptions, we propose an improved routing algorithm, and prove that it has a good time performance with high probability.

1. Introduction

Automated Guided Vehicles (or AGVs for short) have become an important option in material handling [1-7, 9-11, 6]. 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 is 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. 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.

Actually an AGV system is also a multi robot system. There has been research done on the routing strategy in the multi robot system[12-15], but these solutions assume a small number of robots on the mesh layout—no more than $O(\sqrt{N})$ or $O(n)$ for an $N = n \times n$ mesh layout. However, since there are n^2 nodes in the mesh layout, it should be able to accommodate more AGVs/ robots. In [15], $O(\frac{n^2}{\log n})$ number of robots is considered, and a good routing algorithm is presented to finish all tasks in $O(n)$ steps with high probability.

In this paper, we improve the routing algorithm of [15] and we show that using our routing algorithm, the permutation tasks can be finished in $O(n)$ steps with higher probability than that in [15].

The remainder of the paper is organized as follows. Section 2 describes the routing model. Section 3 gives the routing algorithm. In Section 4, we analyze the time performance of the routing algorithm. Finally, Section 5 discusses possibilities of relaxing certain constraints and points out directions of future study.

2. Routing model

In our AGV system, 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. 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 upper-left side, there is a vehicle park where all AGVs are stationed initially and to which they will return upon completion of all tasks.

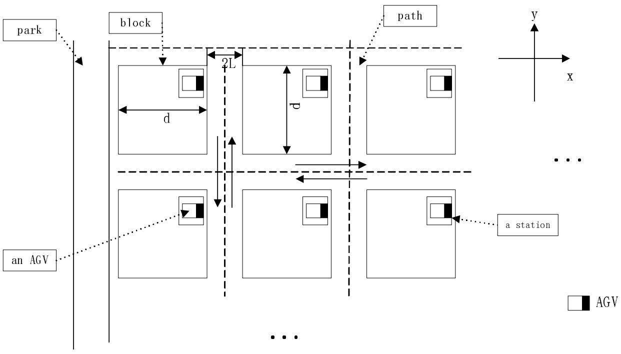


Figure 1. Realistic mesh layout

Although there are some important details for AGV routing, such as the size of the junction, the radius of turns, 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, as shown in Figure 2, there are junctions of pathways. A junction and the associated neighboring station are collectively regarded as a node. Each node is assigned with the coordinates as its address or ID, where x and y represent respectively the row and column IDs. This mesh layout is modeled by a graph. The 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.

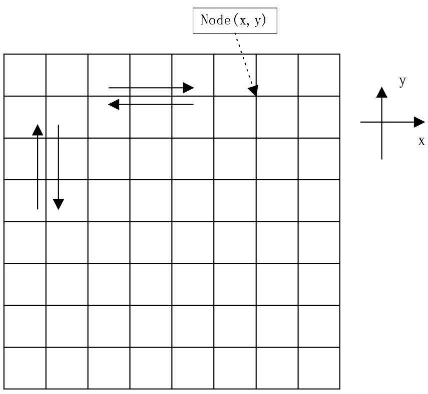


Figure 2. Simplified mesh routing model

We organize 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 will focus only on the second phase when the loaded AGVs move on the mesh layout. In the following, a *step* of an AGV means that it moves from one node to one of its neighboring nodes.

In the mesh topology, we assume that the number of AGVs, m, is bounded by $O(\frac{n^2}{\log n})$. Thus, in the following, we suppose that $m = c_m \frac{n^2}{\log n}$.

The movement pattern is a 1-1 partial permutation, which is defined as follows.

$$\sum_{n\times n} = \{\sigma \mid \sigma : Z_n \times Z_n \to Z_n \times Z_n, \sigma is 1-1, |\sigma| = m\},$$

where $m = O(\frac{n^2}{\log n})$.

At the same time, we assume that the communication mechanism among all AGVs allows each AGV to detect the AGVs which are one unit distance around it. As shown in Figure 3, the AGV in center can detect the AGVs in the "dot" points.

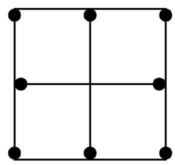
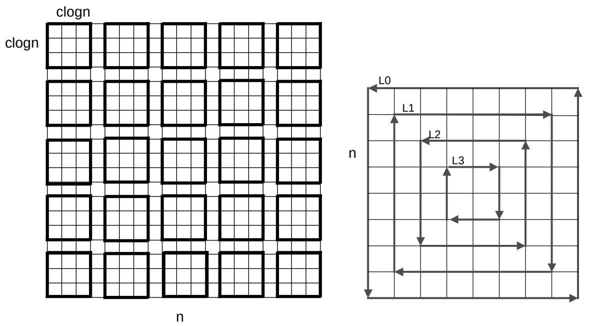


Figure 3. Communication level

As in [15], the mesh layout for routing is partitioned into imaginary squares. Each square consists of $c \log n \times c \log n$ nodes of the grids, as shown in Figure 4 (a). There are $\frac{n}{c \log n}$ rows of squares and $\frac{n}{c \log n}$

columns of squares. Each square is marked by the coordinates *(i,j)* as its address or ID, where *i* and *j* represent respectively the row and column. At the same time, we assume all AGVs in each square can only travel in the pre-specified cycle direction shown in Figure 4(b). The directions of any two neighboring cycles are different. The cycles are represented by L_0, L_1, L_2,…, L_k, where L_k represents the boundary, and L_1 represents the next internal cycle, …, L_k represents the innermost cycle in the square.



(a).The partition of mesh layout (b). Imaginary cycles in each square in imaginary squares

Figure 4. Pre-specialization of the mesh layout.

At the same time, we follow the formal definition of good partial permutations defined by [15].

Definition A.1: *For a permutation* $\sigma : Z_n \times Z_n \rightarrow Z_n \times Z_n$, $\sigma \in \sum_{n\times n}$, *if at most* $C_g \log n$ *AGVs are originated from (or destined to) every square, we call* σ *a* ***good partial permutation****, where* $3c-1 \geq C_g \geq max\{12c^2 c_m, 6\}$.

Since in $n \times n$ mesh layout, a random $\sigma \in \sum_{n\times n}$ is a good partial permutation with high probability $1-\frac{1}{n^3} \approx 1$, for large n, it is reasonable for us to assume that in our routing system, the permutation is a good partial permutation. Our routing algorithm is based on this assumption. In Section 5, we will show how to deal with the routing problem if this assumption is relaxed.

Based on the pre-specified squares in the mesh layout and the good partial permutation, we formally define the following notations.

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

Definition (Origin square job set): *An origin square job set* $S^P_{(i,j)}$ *denoting a job set in which each job is originated from the square (i,j), i.e.*

$$S^P_{(i,j)} = \{ J((PX,PY),(DX,DY)) \,|\, (PX,PY) \in square(i,j) \}$$

Note that, by our assumption, $|S^P_{(i,j)}| \leq C_g \log n$.

Definition (Destination square job set): *A destination square job set* $S^D_{(i,j)}$ *denoting a job set in which each job is destined to the square (i,j), i.e.*

$$S^D_{(i,j)} = \{ J((PX,PY),(DX,DY)) \,|\, (DX,DY) \in square(i,j) \}$$

Note that, by our assumption, $|S^D_{(i,j)}| \leq C_g \log n$.

Definition (Square cycle): *A square cycle L denoting a cycle an AGV's job set in which each job is destined to the square (i,j), i.e.*

$$S^D_{(i,j)} = \{ J((PX,PY),(DX,DY)) \,|\, (DX,DY) \in square(i,j) \}$$

Note that, by our assumption, $|S^D_{(i,j)}| \leq C_g \log n$.

Definition (AGV's status): *An AGV's status denoting the position of the AGV in the mesh layout is defined by* $A((S_x, S_y), L, (x,y))$*, where* (S_x, S_y) *is the AGV's square ID, and* (x,y) *is the AGV's position ID within the square* (S_x, S_y). *L is the AGV's cycle position in the square.*

Definition (Priority)[15]: *The priority is that AGVs which continue circling on the same square boundary are preferred over AGVs that try to go into a neighboring square boundary".*

For example, in the right side of Figure 5, if the AGV on node 7 wants to go to the boundary of the square on its right hand side, then AGV on node 5 has higher priority.

3. Routing algorithm

[15] proposed Square Algorithm, as illustrated in Figure 5, which consists of three phases: In the first phase, every robot tries to move from its origin to the boundary of the square, using the imaginary internal Hamiltonian cycle in the square containing its origin. In the second phase, using only nodes on the boundary of the square that contains its destination. In the third phase, every robot moves from the square's boundary to the destination inside the square, also walking through the Hamiltonian cycle.

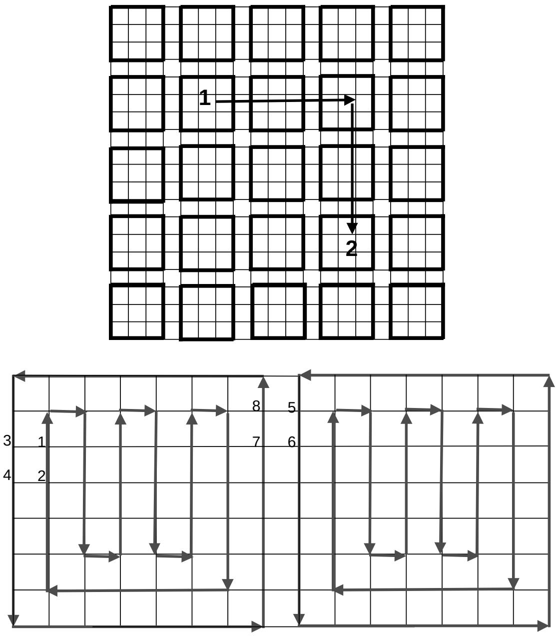


Figure 5. The square algorithm in [15]

Our AGV routing algorithm also consists of three phases. The difference is that in the first and last phases, we use square cycles instead of Hamiltonian cycles. In the middle phase, we have more than one path to go, not only one specified way as in the square algorithm in [15].

Based on the same assumption of good partial permutation, our routing algorithm is given as follows.

Suppose that an AGV's status is $A((S_x, S_y), L, (x, y))$, and its job is $J((PX,PY),(DX,DY))$. Square $(S_x^{'}, S_y^{'})$ is the neighboring square of (S_x, S_y), then we know that $S_x^{'} = S_x \pm 1$ or $S_y^{'} = S_y \pm 1$.

The algorithm, divided into three phases, is given as follows.

Phase 1. Move the AGVs from their origins to the square's boundary.

Repeat for $c^2 \log^2 n$ steps (in Section 4, we will prove that in the worst case, after $c^2 \log^2 n$ steps, all AGVs will finish their first phase).

If the AGV is on the boundary L_0
Then advance on the boundary in clockwise direction
Else if the AGV is on the cycle L_i and there is no AGV with higher priority on the cycle L_{i-1},
Then it moves into the L_{i-1}
Else it advances on the cycle L_i

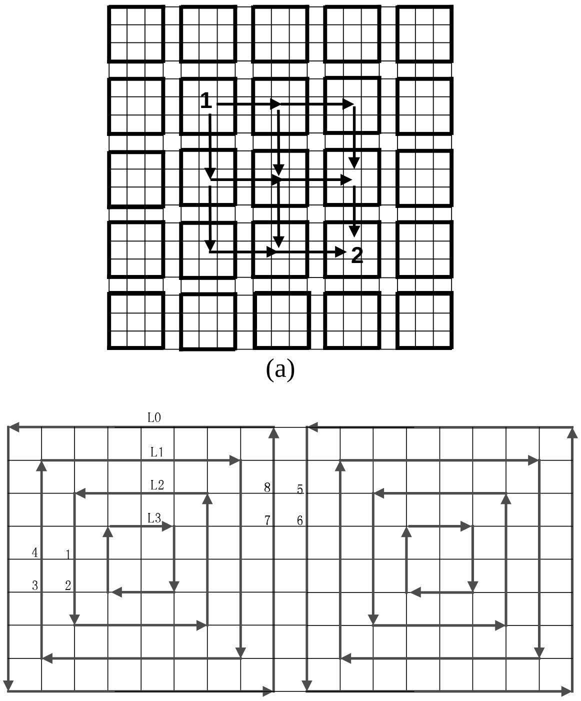
Phase 2. Move the AGVs from their origin square boundaries to their destination square boundaries.

Repeat at each step
If $(S_x, S_y) = (DX, DY)$
Then it starts last phase
Else If $S_x^{'} = S_x + sgn(DX - PX)$ and there is no AGV with higher priority in $(S_x^{'}, S_y^{'})$ for AGV *A* (where *sgn(DX-PX)=1*, if *DX>PX*; otherwise, *sgn(DX-PX)=0*)
Then it moves into the square $(S_x^{'}, S_y^{'})$
Else it advances on the boundary of square (S_x, S_y)

Phase 3. Move the AGVs from the boundary of the destination's square to their destination.

Repeat for $c^2 \log^2 n$ steps (in Section 4, we will prove that in worst case, after $c^2 \log^2 n$ steps, all AGVs will finish their last phase).

If the AGV reaches its destination (DX, DY)
Then it enters the buffer and leaves the mesh
Else if the AGV is on the cycle L_i and there is no AGV with higher priority on the cycle L_{i+1},
Then it jumps to the L_{i+1}
Else it advances on the cycle L_i



(a)

(b)

Figure 6. Our routing algorithm.

4. Analysis of time complexity

We analyze the time performance of each phase in our routing algorithm.

Claim 4.1: *In the worst case, the first phase in our routing algorithm takes* $O(\log^3 n)$ *steps for all AGVs to complete their permutation operations.*

[Proof]: Since the directions of two neighboring cycles are different, the AGVs on one cycle can only "disturb" each AGV on another cycle once (" disturb" means an AGV--*A* blocks another one to come to the same cycle because of *A* has higher priority). Because $4c\log n > c_g \log n$, for the second cycle L_1, which is close to the square boundary and the size of which is $(c\log n - 2)\times(c\log n - 2)$, in the worst case, an AGV on it will take $4(c\log n - 2)$ steps to reach the boundary. Similarly, for the AGV on the next cycle L_2, it will take $4(c\log n - 4)$ steps to reach the cycle L_1, then it would take another $4(c\log n - 2)$ steps, in the worst case. We can analyze the similar cases for the other cycles. Therefore, we get the running time of the first phase in the worst case.

$$\begin{aligned} T_1 &\le 4(c\log n - 2) + [4(c\log n - 2) + 4(c\log n - 4)] \\ &+ \ldots + [4(c\log n - 2) + 4(c\log n - 4) + 4\times 2] \\ &\le c^3 \log^3 n \end{aligned}$$

□

Claim 4.2: *In the worst case, the third phase in our routing algorithm takes* $O(\log^3 n)$ *steps.*

[Proof]: The proof is very similar to that of Claim 4.1 and is therefore omitted.

□

Claim 4.3: *In the second phase of our routing algorithm, with high probability* $(1 - \frac{1}{n^{\frac{c_1}{4}-3}})^{\frac{2n}{c\log n}}$, *all AGVs will reach their destination square's boundary in* $O(n)$ *steps.*

[Proof]: The proof uses an argument similar to that of [15]. The following version of Chernoff bound [8] is used in our proof.

Chernoff Bound[8] Let $p_1, p_2, \ldots, p_n \in R$ with $0 \le p_i \le 1$, for $i = 1,2,\ldots,n$. Let $p = \frac{p_1 + p_2 + \ldots + p_n}{n}$ and $m = np$, and let $X_1, X_2, \ldots, X_n$ be independent Bernoulli random variables with $Prob[X_i] \le p_i$, for $i = 1,2,\ldots,n$, $S = X_1 + X_2 + \ldots + X_n$. Then for $r \ge 6m$, $Prob[S \ge r] \le 2^{-r}$.

We also need need the following lemma.

Lemma 4.3.1 *After the first phase, with probability*

$$(1-\frac{1}{n^{\frac{c_1}{4}-3}})^{\frac{2n}{c\log n}},$$

1. *during each of the first* $\frac{2n}{c\log n}$ *rounds, every AGV moves to the next square in its path,*
   and
2. *during these rounds, at each step, every square has no more than* $2c_1\log n$ *AGVs,*

where $c \geq 5c_1+1$ *and* $c_1 \geq 120c_g$.

[Proof of Lemma 4.3.1]:
Firstly, let's introduce the definitions of certain events also defined in [15].

E_{orig} = *{at most* C_g*logn AGVs are outbound in every square},*

E_{dest} = *{at most* C_g*logn AGVs are inbound to arrive at every square},*
and

$E_0 = E_{orig} \cap E_{dest}$,

where $C_g \geq max\{6, 12c^2c_m\}$.

For $\forall i > 0$

A_i = *{at round* i *all outbound AGVs move to the next square in their path },*

B_i = *{at end of round i there are at most* $c_1 logn$ *AGVs in every square },*

and $E_i = \{A_i \cap B_i\}$, *where* $c \geq 5c_1+1$ and $c_1 \geq 120c_g$.

Since we assume the good partial permutation, $Prob[E_0]=1$.

In order to prove the lemma, we introduce the following claims.

Claim 4.3.1: *For every* $1 \leq t \leq \frac{2n}{c\log n}$, *if* $\bigcap_{i=0}^{t-1} E_i$ *occurs, then* A_t *also occurs.*

[Proof of Claim 4.3.1]:
The proof is the same as that of [15].

□

Claim 4.3.2: *For every* $1 \leq t \leq \frac{2n}{c\log n}$,

$$Prob[B_t \mid \textstyle\bigcap_{i=0}^{t-1} E_i] \geq 1-\frac{1}{n^{\frac{c_1}{4}-3}}.$$

[Proof of Claim 4.3.2]:
See Appendix A.

□

Based on Claim 4.3.1 and Claim 4.3.2, we conclude that

$$Prob[E_t \mid \textstyle\bigcap_{i=0}^{t-1} E_i] = Prob[A_t \cap B_t \mid \bigcap_{i=0}^{t-1} E_i]$$
$$= Prob[B_t \mid \textstyle\bigcap_{i=0}^{t-1} E_i] \geq 1-\frac{1}{n^{\frac{c_1}{4}-3}}$$

Therefore, $\forall 1 \leq t \leq \frac{2n}{c\log n}$ we have

$$Prob[\bigcap_{i=0}^{t} E_i] \geq Prob[E_t \mid \bigcap_{i=0}^{t-1} E_i] \times Prob[E_{t-1} \mid \bigcap_{i=0}^{t-2} E_i]$$
$$\times \ldots \times Prob[E_1 \mid E_0] \times Prob[E_0]$$
$$\geq (1-\frac{1}{n^{\frac{c_1}{4}-3}})^t (\because Prob[E_0]=1)$$

Substituting $t=\frac{2n}{c\log n}$ into the inequality, we have

$$Prob[\bigcap_{i=0}^{\frac{2n}{c\log n}} E_i] \geq (1-\frac{1}{n^{\frac{c_1}{4}-3}})^{\frac{2n}{c\log n}}.$$

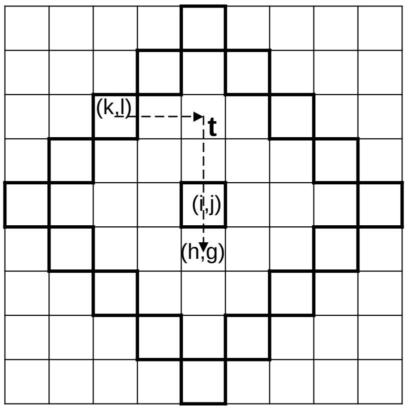
Thus, we get the proof of Lemma 4.3.1.

□

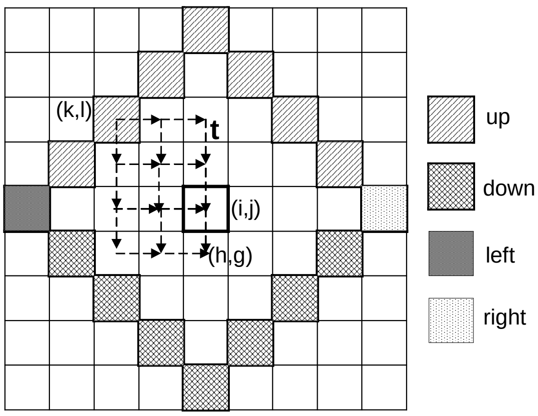
According to Lemma 4.3.1, at each one of the first $\frac{2n}{c\log n}$ rounds, all AGVs move to the next square during their paths. Each path contain at most $\frac{2n}{c\log n}$ squares, and each round needs $4c\log n$ steps, so we know that with probability $(1-\frac{1}{n^{\frac{c_1}{4}-3}})^{\frac{2n}{c\log n}}$, the second phase takes $\frac{2n}{c\log n}\times 4c\log n = O(n)$ steps.

Therefore, we get the proof of Claim 4.3

□



(a) In the square algorithm



(b) In our routing algorithm

Figure 7. The squares of $D_t^{(i,j)}$. The path marked by dashed line is the one for the job J((k,l),(h,g)). For a square $(i,j)\in D_t^{(i,j)}$, $|k-i|+|l-j|=t$.

Claim 4.4: *In our routing algorithm, with high probability, all AGVs will reach their destinations in* $O(n)$ *steps.*

[Proof]: Based on Claim 4.1, Claim 4.2 and Claim 4.3, and since $O(\log^3 n) = O(n)$, we can easily get the proof.

□

5. Discussions and conclusions

In this paper, we have analyzed a realistic AGV system with a mesh layout, and considered the case where the number of AGVs is bounded by $O(\frac{n^2}{\log n})$. Based on some pre-specified path of the mesh layout and the good partial permutation, we present an improved routing algorithm, and prove that with high probability, it can be done in $O(n)$ steps.

Our algorithm is an improvement over the results in [15]. In the second phase of the routing algorithm in [15], each robot can only travel in one special path to reach its destination. In our routing algorithm, every AGV has more paths to choose from than in the square algorithm, when it tries to move towards its destination. Intuitively, because we allow AGVs to move into any square that decreases the square distance to their destinations, it should have more chances to avoid potential conflicts, so it is easier to reach its destination. From the probability analysis, it has also been confirmed.

We assume that the AGVs have good partial permutation. However, when this assumption is not satisfied, we can use a big Hamiltonian cycle in the whole mesh layout, then in the worst case, the permutations which are not good partial ones can be finished in $O(n^2)$ steps.

Our routing algorithm relies on the minimal local communication mechanism. However, the communication level can be extended. Then there should exist a more efficient routing algorithm for finishing the permutation operation.

We have assumed that the permutations are 1-1, and each AGV is only assigned to one job. These assumptions can also be relaxed. When an AGV just finishes dropping off a box (or container, etc.) and picks up a new one, we can regard it a new AGV originating at that time moment (suppose that the assumption of good partial permutation is still satisfied. Therefore, removing this assumption would not add much difficulty to our analysis.

In this paper, we only consider the time performance in the routing algorithm. But in our mesh routing algorithm, all AGVs should make many turns before they reach their destinations, thus, they consume more energy than some other greedy routing algorithms [5]. Therefore, it is important for us to consider the energy efficiency in the routing algorithm.

There are still many open issues for future research. Firstly, how to extend the simplified routing model in which each block is not a square, but instead, a rectangle. Secondly, we assumed that the buffer of each node can only accommodate one AGV, and there is no queue in the routing model. How to determine the size of the buffer and the queue, if the assumption is relaxed? Thirdly, in our study, we did not consider the case when some AGVs break down, or when the communication system is broken. These failures could lead to a serious problem of the whole system. Therefore, it is essential to consider fault-tolerant algorithms.

Acknowledgment

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, L., W. –J. 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, L., W. –J. 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, L., W. –J. 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] T. Hagerup and C. Rub, A guided tour of Chernoff bounds. Information Processing Letters 33, 305-308, 1989-90.

[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.

[15] 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.

[12] Y. Moses and M. Tennenholtz, On computational aspects of artificial social systems. In Proceedings of DAI-92, 1992.

[13] Y. Shoham and M. Tennenholtz, On traffic laws for mobile robots. In First Conference on AI Planning Systems, 1992.
[14] Y. Shoham and M. Tennenholtz, On social laws for artificial agent societiew: Off-line design. Artificial Intelligence, vol. 73, 1995.
[15] Preminger, S. Complexity analysis of movement in multi robot system. Master's thesis, Department of Applied Mathematics, the Weizmann Institute of Science, Rehovot, Israel, 1995.
[16] Zeng, J., Hsu W. –J. and Qiu L.. An Energy-Efficient Algorithm For Conflict-Free AGV Routing On A Linear Path Layout. In Proceedings of the International Computer Symposium (ICS 2002), Hualian, Taiwan, Dec. 18-21, 2002.

Appendix A: Proof of Claim 4.3.2

[Proof]:

From Claim 4.3.1, if $\bigcap_{i=0}^{t-1} E_i$, then A_t occurs, namely, all outbound AGVs which are on the boundary of the square at the beginning of round *t*, will leave the square during the round. So we only need to consider the AGVs entering the square at the *t*-th round. For this purpose, we consider the following events.

$B_t^{(i,j)} = \{$*at most* $c_1 logn$ *AGVs are arriving into square (i,j) at round t* $\}$,

then we know that $B_t = \bigcap B_t^{(i,j)}$.

$D_t^{(i,j)} = \{$ *the set of all squares that are at a distance of t squares from (i,j)*$\}$,

The squares of $D_t^{(i,j)}$ are shown in Figure 7 (b).

We use the following Bernoulli variables.

$X_{(k,l),m}^{(i,j)} =$ 1 if the *m-th* AGV originating from (k,l) has (i,j) on its path; 0 otherwise (including the case in which less than m AGVs originate in (k,l)),

where $(k,l) \in D_t^{(i,j)}$, and $m \le c_g \log n$ (by the assumption of good partial permutations)

In order to use the Chernoff bound, each variable must be independent. However, in our formula, $X_{(k,l),m}^{(i,j)}$ is not independent of $X_{(k',l'),m'}^{(i,j)}$, for $(k,l),m \neq (k',l'),m'$. Each of the four sets marked by different patterns in Figure 7 (b) is independent of the others, so we introduce the following independent events according to Figure 7 (b).

$$C_{up} = \{ \textstyle\sum_{(k,l)\in D_t^{(i,j)}, i<k, m} X_{(k,l),m}^{(i,j)} \le \frac{c_1}{4} \log n \}$$

$$C_{down} = \{ \textstyle\sum_{(k,l)\in D_t^{(i,j)}, i>k, m} X_{(k,l),m}^{(i,j)} \le \frac{c_1}{4} \log n \}$$

$$C_{right} = \{ \textstyle\sum_{(k,l)\in D_t^{(i,j)}, l>j, i=k, m} X_{(k,l),m}^{(i,j)} \le \frac{c_1}{4} \log n \}$$

$$C_{leftt} = \{ \textstyle\sum_{(k,l)\in D_t^{(i,j)}, l<j, i=k, m} X_{(k,l),m}^{(i,j)} \le \frac{c_1}{4} \log n \}$$

Now we will calculate $Prob[\overline{C_{up}} \mid \bigcap_{i=0}^{t-1} E_i]$, $Prob[\overline{C_{down}} \mid \bigcap_{i=0}^{t-1} E_i]$, $Prob[\overline{C_{left}} \mid \bigcap_{i=0}^{t-1} E_i]$, and $Prob[\overline{C_{right}} \mid \bigcap_{i=0}^{t-1} E_i]$ respectively, where $\overline{C}$ denotes the complement of *C*.

$$Prob[\overline{C_{up}} \mid \bigcap_{i=0}^{t-1} E_i] = \frac{Prob[\overline{C_{up}} \cap (\bigcap_{i=1}^{t-1} E_i \mid E_0)]}{Prob[\bigcap_{i=1}^{t-1} E_i \mid E_0]}$$

$$\le \frac{Prob[\overline{C_{up}} \mid E_0]}{Prob[\bigcap_{i=1}^{t-1} E_i \mid E_0]} \le \frac{Prob[\overline{C_{up}} \mid E_0]}{1 - 1/n}$$

$$\le 2 \times Prob[\overline{C_{up}} \mid E_0] \le 2 \times Prob[\overline{C_{up}}] (\because Prob[E_0] = 1)$$

According to Figure 7 (b), we know that there are at least $(n^2 - cn\log n)$ (for $n \ge 2$) nodes that can be the destinations of AGVs that originate in (k,l) for the *m*-th AGV(all the nodes minus the nodes of the "up" set).

What interest us are the nodes that can be possible destination nodes. According to Lemma A.1, the largest number of squares in $D_t^{(i,j)}$ is $\frac{5}{2}\times\frac{n}{c\log n}$, and there are at most $c^2\log^2 n$ destination nodes in every square. So there are at most $\frac{5}{2}\times\frac{n}{c\log n}\times c^2\log^2 n=\frac{5}{2}cn\log n$ nodes that can be possible destination nodes. Therefore, we have

$$E[X_{(k,l),m}^{(i,j)}]\le\frac{\frac{5}{2}cn\log n}{n^2-cn\log n}\le\frac{5c\log n}{n}$$

Next, in order to use the Chernoff bound, we argue that $\sum_{(k,l)\in D_t^{(i,j)},i<k,m}X_{(k,l),m}^{(i,j)}$ is stochastically dominated by the sum $\sum_{j=1}^{\frac{c_g n}{c}}Y_j$, where Y_j are independent Bernoulli trials with success probability $\frac{5c\log n}{n}$ (we sum to $\frac{c_g n}{c}$ since there are totally $\frac{n}{c\log n}\times c_g\log n=\frac{c_g n}{c}$ nodes in the "up" set).

Thus we have

$$E[\sum_{(k,l)\in D_t^{(i,j)},i<k,m}X_{(k,l),m}^{(i,j)}]\le E[\sum_{j=1}^{\frac{c_g n}{c}}Y_j]=\frac{c_g n}{c}\frac{5c\log n}{n}=5c_g\log n$$

By Chernoff bound we get for $c_1\ge 120c_g$

$$Prob\{\sum_{(k,l)\in D_t^{(i,j)},i<k,m}X_{(k,l),m}^{(i,j)}\ge\frac{c_1\log n}{4}\}$$
$$\le Prob\{\sum Y_j\ge\frac{c_1\log n}{4}\}\le 2^{-\frac{c_1\log n}{4}}\le\frac{1}{n^{\frac{c_1}{4}}}$$

Therefore, we have

$$Prob[\overline{C_{up}}\mid\bigcap_{i=0}^{t-1}E_i]\le\frac{1}{n^{\frac{c_1}{4}}}.$$

Since the "down" set is symmetrical to the "up" set, we have

$$Prob[\overline{C_{down}}\mid\bigcap_{i=0}^{t-1}E_i]\le\frac{1}{n^{\frac{c_1}{4}}}.$$

Because there are at most $c_g\log n$ AGVs originating in each square, and $c_1>4c_g$, we have

$$Prob\{\sum_{(k,l)\in D_t^{(i,j)},l>j,i=k,m}X_{(k,l),m}^{(i,j)}\ge\frac{c_1\log n}{4}\}$$
$$\le Prob\{\sum_{(k,l)\in D_t^{(i,j)},l>j,i=k,m}X_{(k,l),m}^{(i,j)}\ge c_g\log n\}=0$$

So $Prob[\overline{C_{right}}\mid\bigcap_{i=0}^{t-1}E_i]=0$. Similarly we get $Prob[\overline{C_{left}}\mid\bigcap_{i=0}^{t-1}E_i]=0$.

Now we continue to prove Claim 4.3.2.

Since $B_t^{(i,j)}\supseteq(C_{up}\cap C_{down}\cap C_{right}\cap C_{left})$, we get

$$Prob[B_t^{(i,j)}\mid\bigcap_{i=0}^{t-1}E_i]\ge Prob[C_{up}\cap C_{down}\cap C_{right}\cap C_{left}\mid\bigcap_{i=0}^{t-1}E_i]$$
$$=1-Prob[\overline{C_{up}}\cup\overline{C_{down}}\cup\overline{C_{right}}\cup\overline{C_{left}}\mid\bigcap_{i=0}^{t-1}E_i]$$
$$\ge 1-2\times\frac{1}{n^{\frac{c_1}{4}}}\ge 1-\frac{1}{n^{\frac{c_1}{4}-1}}$$

Thus, we have

$$Prob[B_t\mid\bigcap_{i=0}^{t-1}E_i]=1-Prob[\overline{B_t}\mid\bigcap_{i=0}^{t-1}E_i]$$
$$=1-Prob[\overline{\bigcap_{(i,j)}B_t^{(i,j)}}\mid\bigcap_{i=0}^{t-1}E_i]$$
$$\ge 1-prob[\bigcup_{(i,j)}\overline{B_t^{(i,j)}}\mid\bigcap_{i=0}^{t-1}E_i]$$
$$\ge 1-\sum_{(i,j)}prob[B_t^{(i,j)}\mid\bigcap_{i=0}^{t-1}E_i]\ge 1-\sum_{(i,j)}\frac{1}{n^{\frac{c_1}{4}-1}}$$
$$=1-\frac{\frac{n^2}{c^2\log^2 n}}{n^{\frac{c_1}{4}-1}}\ge 1-\frac{1}{n^{\frac{c_1}{4}-3}}$$

Therefore we complete the proof of Claim 4.3.2.

□

Lemma A.1: *Consider a mesh with $P\times P$ number of squares. For a given squares (i,j), there are at most possible $\frac{5}{2}P$ squares that are the destinations of the AGVs that originate from square which is in $D_t^{(i,j)}$ and have the square (i,j) on their paths.*

[Proof]: When *(i,j)* is the center of the mesh, we have the maximum of the possible destination, where $t = \frac{P}{2}$. For convenience, we set *(i,j)* to be the *(0,0)* point of the coordinates. For a given square *(k,l)* in $D_t^{(i,j)}$, and any square *(h,g)* is a square that has the AGVs that originate from square *(k,l)* and have square *(i,j)* on their paths, as shown in Figure 7 (b), there are totally $S_1 = \frac{(k+l)!}{k!l!}$ square paths from *(k,l)* to *(i,j)*, and totally $S_2 = \frac{(h+g)!}{h!g!}$ square paths from *(i,j)* to *(h,g)*. At the same time, there are totally $S = \frac{(k+l+h+g)!}{(h+k)!(l+g)!}$ *paths from* *(i,j)* to *(h,g)*.

The probability, of which *(h,g)* can be the square originating and having the square *(i,j)* on its path, is given as follows.

$$Pr^{(h,g)} = \frac{S_1 \times S_2}{S} = \frac{(k+h)(k+h-1)...(k+1)\times(l+g)(l+g-1)...(l+1)}{(k+l+h+g)(k+l+h+g-1)...(h+g+1)}$$

where $k+l \le \frac{P}{2}$ and $0 \le h,g \le \frac{P}{2}$.

When *h* or *g* increases, *Pr* decreases. Suppose that there are at most *S* possible squares that satisfies the requirement, we have

$$\#S \le P \sum_{h=g=0}^{P/2} Pr^{(h,g)}$$

For $h = g = x(x > 0)$, we have

$$Pr^{(x,x)} \le \frac{2}{4^x} < \frac{1}{x^3}.$$

Therefore, we have

$$\#S \le P(2 + \sum_{h=g=2}^{P/2} Pr^{(h,g)}) \le P(2 + \sum_{x=2}^{P/2} \frac{1}{x^3})$$
$$\le 2P + P\int_1^{\frac{P}{2}} \frac{1}{x^3} dx \le 2P + P(\frac{1}{2} - \frac{2}{P^2}) \le \frac{5}{2}P$$

□