Abstract: In this paper, we study the design and control of automated guided vehicle (AGV) systems, with the focus on the quayside container transport in an automated container terminal. We first set up an event-driven model for an AGV system in the zone control framework. Then a number of layouts of the road network (guide-path) are carefully designed for the workspace of the AGVs in a container terminal. Based on our zone control model, a traffic control strategy is proposed, which decouples the motion conflict resolution among the AGVs from the routing problem. A routing algorithm is also constructed with the goal of minimizing the vehicle travel distances and the transportation time. The efficiency of the integrated design and control is demonstrated by computer simulations in terms of a set of defined measures of system performance. Lastly, we point out several possibilities towards improving our current results.

1. INTRODUCTION

Automated guided vehicles normally mean mobile robots (or unmanned vehicles) used in transporting objects. They were traditionally employed in manufacturing systems, but have recently extended their popularity to many other industrial applications, such as goods transportation in warehouses and container transshipment at container terminals. See Vis [2006] for an comprehensive survey of the research on the design of AGV systems.

The booming of international trade spurs the development of automated container terminals (ACTs), equipped with automated container transshipment systems (consisting of automated cranes and automated guided vehicles etc.), that can meet rapidly increasing demands for higher operational efficiency, lower costs, and smaller variability than what traditional terminals can achieve. There are four main issues in building an AGV system in an ACT. The first is the design of the guide-path (we call it road network in this paper) that specifies possible paths on which the vehicles can travel [Steenken et al., 2004]. The second is the dispatching problem which is about where and when vehicles should go for the container loading or discharging tasks [Briskorn et al., 2006, Kim and Bae, 1999, Bish et al., 2005, Nguyen and Kim, 2009]. The third is the vehicle routing aimed at finding good paths for vehicles dispatched for certain tasks [Steenke et al., 1993, Duinkerken et al., 2006, Stahlbock and Voß, 2008a]. The last is the conflict resolutions among the fleet of vehicles and between the vehicles and other container handling equipments [Evers and Koppers, 1996, Lehmann et al., 2006, Kim et al., 2006]. Although all these matters are interconnected as far as the system performance is concerned, it is very difficult to take them into a comprehensive consideration. As a meaningful step towards a complete solution, in this paper, we present a package of designs and controls that address the first, third and fourth issues simultaneously for the quayside container transportation.

Our contributions can be summarized as follows. We first establish a novel zone control model underlying our design and control of an AGV system. This model elaborates the structure of the road network (RN) and defines the vehicles’ behavior when traveling on the road network. Then based on this model, a number of RN layouts are designed according to the specifications of a real ACT. With these settings, a traffic control strategy is proposed for the warranty of no inter-vehicle collisions and deadlocks. The traffic control enables the vehicles to freely choose/change their routes and has small time complexity. Concerned about the system performance, a heuristic vehicle routing algorithm is developed which is oriented at minimizing the travel distance of the vehicles as well as the timespan for the container transportation. Incorporating with a task dispatching scheme of random nature, the performance of the AGV system with the set of proposed designs and controls is investigated by computer simulations. Finally, some discussions and open problems for future work are given.

2. ZONE CONTROL MODEL

The workspace of the AGVs is divided into non-overlapping zones, and the entrance of a zone is strictly controlled. This strategy eases the avoidance of inter-vehicle collisions by demanding that each zone can be occupied by at most one vehicle. Another reason that makes it the favorite of most applications is that it is simple to install and easy to expand. Our zone control model described below contains two components: the structure of the road network and the discrete-event based behavior of a vehicle when it follows a route consisting of a sequence of zones.
2.1 Building blocks of the road network

The road network is composed of lanes, crossings and depots.

Lane and depot  A lane is a finite sequence of zones. Physically, a lane is a road segment on which a vehicle can move in the direction specified by the order of zones of the lane. Particularly, we call the first zone and the last zone of a lane the starting zone (SZ) and ending zone (EZ) of the lane respectively.

Every depot is modeled as a zone that can accommodate any number of vehicles. Each depot is affiliated with at least one in-lane and one out-lane. An in-lane (resp. out-lane) of a depot is a lane with the direction that allows a vehicle moving inwards (resp. outwards from) the depot. We emphasize that a depot is not a zone of any lane.

We use $C$ to denote the set of all zones in the system respectively.

Crossing  A crossing is physically a junction area connecting multiple lanes. Specifically, each crossing is attached with a set of in-lanes and a set of out-lanes. An in-lane (resp. out-lane) of a crossing allows a vehicle to move towards (resp. away from) the crossing. The EZ of any in-lane of a crossing is named an at-crossing zone of the crossing. The zones that are not at-crossing zones are called off-crossing zones. Note that any depot is an off-crossing zone as it is not on any lane. Let $R$ and $A$ be the sets of all crossings and the set of all at-crossing zones respectively.

Neighboring zone  The concept of neighboring zone characterizes immediate zone-to-zone connections. The set of neighboring zones of each depot includes the SZs of all its out-lanes. The neighboring zone of a non-EZ zone $c$ on a lane is the adjacent zone of $c$ on the lane with respect to the direction of the lane. The neighboring zone of the EZ of any in-lane of a depot is the depot. The neighboring zones of the EZ of an in-lane of a crossing (i.e., an at-crossing zone) are the SZs of all the out-lanes of the crossing. We use $T_c$ to denote the set of neighboring zones of zone $c$. (One can see from Assumption 1 that neighboring zones are defined for any zone in $C$.)

We impose the following assumption on the layout of the road network.

Assumption 1. (a) Each lane is an in-lane (out-lane) of either a unique crossing or a unique depot; (b) each lane has at least two zones; (c) any in-lane of a crossing has at least one neighboring lane.

2.2 Vehicle routes, states and events

We consider an AGV system with $N$ vehicles, which is denoted by the set $V = \{v_1, v_2, \ldots, v_N\}$. A finite sequence of zones $a_1^t, a_2^t, \ldots, a_n^t$, $c_i \in \mathbb{N}$, is called a route of $v_i \in V$ if $a_{i+1}^t \in T_{a_i^t}$ for any $p \in Z_{a_i^t}^{c_i}$ (For any $m, n \in \mathbb{N}$, $Z_m^n$ is defined to be $(m, m+1, \ldots, n)$ if $n \geq m$ and $\emptyset$ if $n < m$). There are only two types of vehicle states: being in $a_p^t$, for any $p \in Z_{a_p^t}^{c_p}$, or moving from $a_p^t$ to $a_{p+1}^t$, for any $p \in Z_{a_p^t}^{c_p}$.

The event-driven behavior of the vehicle $v_i$ can be sketched as follows. At a certain time when $v_i$ is in $a_{p_i}^t$, $p \in Z_{a_{p_i}}^{c_{p_i}}$, it triggers an “intend to leave” event. For the sake of collision and deadlock avoidance, the vehicle is required to check some traffic rules (see Section 4) to make a “go ahead” or “stop” decision. If the vehicle will be permitted to go ahead, it triggers a “leave” event and changes its state to be moving from $a_{p_i}^t$ to $a_{p_i+1}^t$; otherwise it has to stop but will continually trigger the “intend to leave” event for leaving the zone $a_{p_i}^t$. In the former case, after the vehicle has completely entered $a_{p_i+1}^t$, it triggers an “arrive at” event, and changes its state to be in $a_{p_i+1}^t$. The process above keeps iterating and terminates if $v_i$ is in $a_n^t$.

We say that a vehicle arrives at, intends to leave, or leaves some zone at time $t$ if the vehicle triggers the corresponding event at $t$. Note that a vehicle switches its state only when it arrives at or leaves a zone, and must be in the zone when it intends to leave a zone.

3. LAYOUT DESIGN OF THE ROAD NETWORK

3.1 Quayside Container Transport at an ACT

In this work, we consider a container handling scenario in which the quay cranes (QCs) and yard stackers (YSs) are in charge of the container collection operations in the quay area (QA) and yard area (YA); and a team of AGVs are used for shuttling the containers between the two areas. The operation is illustrated in Figure 1. Specifically, in discharging a vessel, the QCs take the containers off the vessel and put them in the associated container buffers. These containers will be later on picked up by some of the AGVs, and transported across the transportation area (TA) to the container buffers of some specified YSs in the yard area. There the containers will be put in container stacks by the YSs. The other way around, in loading a vessel, the containers are collected from certain yard stacks by the YSs and transported to designated QCs by the AGVs. In this case study, the workspace of the AGVs is considered as the combined area of the QA, TA and YA.

Fig. 1. Quayside container transport at an automated container terminal

Now suppose that a road network defined as in Section 2 is built over the workspace, with the buffer of each QC or YS modeled as one or more zone(s) (See also Subsection 3.2). For simplicity, these zones are said to be the zones of the QC or YS. Then based on the working scenario described above, each AGV in operation can be seen as
being assigned a sequence of tasks, where a task of a vehicle means an ordered pair of zones, say \((c_1, c_2)\). We call the former zone \(c_1\) and the latter \(c_2\) the source zone and the destination zone of the task. In addition, a route of a vehicle is said to be a route for the task \((c_1, c_2)\) if it starts with \(c_1\) and ends up with \(c_2\).

3.2 Layout Design of the Road Network

In this case study, we consider a typical large container terminal with a TA of the dimension 2000m×40m; and 20 QCs and 66 YSs distributed on the sea side and yard side respectively (see Figure 5). In addition, without any loss of generality, we assume that there is only one depot, which is located outside the workspace (but connected to the TA by its in-lanes and out-lanes), and hence not a subject of our layout design. For this reason, in the remainder of this subsection, by zones we mean non-depot zones.

Geometrical properties of the zones First we fix the shape of the zones to be rectangular because the rectangular shape of the vehicles. We call the dimension of the edge of a zone along which a vehicle moves the length of the zone; the dimension of the other edge the width. The geometric and kinematic specifications of the AGVs considered in this paper are listed in the table below.

<table>
<thead>
<tr>
<th>Geometric and kinematic characteristics of AGVs</th>
<th>Length (m)</th>
<th>Width (m)</th>
<th>Max speed (m/s)</th>
<th>Max acc. (m/s²)</th>
<th>Max dec. (m/s²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10m</td>
<td>4m</td>
<td>7m/s</td>
<td>0.5m/s²</td>
<td>2.5m/s²</td>
<td></td>
</tr>
</tbody>
</table>

Based on the above specifications of AGVs, we put the length of a zone 22m (slightly larger than the sum of the length of a vehicle and the distance a vehicle takes to make a full stop from the maximum speed) and the width 7m. The basic rationale behind this choice is that, for safety reasons, the body of a vehicle must be completely inside a zone when the state of the vehicle is in the zone. See Li et al. [2011] for more details.

Layout design for the quay area and yard area For simplicity, the road network has the same layout at each QC or YS, which is illustrated in Figure 2. The buffer of a QC or YS is modeled by the SZs on three different lanes in between two crossings. The 2 in-lanes of one of the two crossings (left one in Figure 2) allow vehicles to drive off the TA and towards the QC or YS; while the 2 out-lanes of the other crossing lead the vehicles back to the TA.

![Fig. 2. Road network at a QC or YS](image)

Layout design for the transportation area In view of the geometric characteristics of the TA, we focus on the layouts with straight horizontal (parallel to the long edge of the TA) adjacent roads. A road is indeed composed of multiple lanes, joined by crossings, on the same vertical level. See Figure 3 for an example of such layouts.

For simplicity, each crossing is chosen to be rectangular. In order to avoid collision between a vehicle passing some crossing and another vehicle waiting at the same crossing, each crossing should be able to accommodate the whole body of a vehicle. Considering the turning radius of the vehicles, the horizontal dimension of the crossings is set as 15m. The vertical dimension of a crossing depends on the layout design and its location in the TA.

As the vertical dimension of the TA is considered here as 40m, it allows at most 6 parallel roads. Based on this restriction, four types of layouts for the TA will be discussed:

- (T1) 4 parallel roads and small crossings.
- (T2) 4 parallel roads and big crossings.
- (T3) 6 parallel roads and small crossings.
- (T4) 6 parallel roads and big crossings.

For want of space, only examples of the layouts of Types T1 and T2 are shown here in Figures 3 and 4. (We refer the reader to Li et al. [2011] for illustrations of other types of layouts.)

A small crossing is defined as a crossing which connects two adjacent parallel roads (See Figure 3). A big crossing is a crossing which connects more than two parallel roads (see Figure 4). One advantage of small crossings is that only a small number of AGVs need to wait at the crossing while some other vehicle is passing the crossing. The main disadvantage of using small crossings is that an AGV has to pass a relatively large number of crossings for a QC-YS or YS-QC task. On the other hand, the use of big crossings might increase the waiting time of the AGVs at the crossings due to possibly more vehicles competing for the crossings; but in general, compared with using small crossings, it reduces significantly the distances of the transportation tasks by providing direct channels linking the QA and the YA. The latter point is illustrated in Figures 3 and 4 by two routes (in dashed lines) that serve the same purpose for a vehicle to cross the TA.

In Figures 3 and 4, each crossing with an upward (resp. downward) arrow, called a TA-YS (resp. TA-QC) crossing is linked with the in-lanes and out-lanes of two adjacent YSs (resp. one QC), i.e., it allows a vehicle to move to two YSs (one QC) from the TA and vice versa by passing it.

![Fig. 3. An example of layout type T1](image)
Fig. 4. An example of layout type T2

In the layouts consisting of 4 parallel roads, all lanes in the 2 upper roads have the same direction which is opposite to the common direction of the lanes in the 2 lower roads. Analogously, in the case of 6-road layouts, the direction of all lanes in the 3 upper roads is the opposite to that of the lanes in the 3 lower roads. These choices aim at making the AGVs brake as little as possible when traveling across the TA. See Li et al. [2011] for more details.

4. TRAFFIC CONTROL AND ROUTING ALGORITHM

4.1 Traffic Control

The goal of the traffic control is to prevent inter-vehicle collisions and deadlocks. This is an important issue that has attracted intensive research effort [Evers and Koppers, 1996, Rajeeva et al., 2003, Steenken et al., 2004, Vis, 2006, Kim et al., 2006]. See also Fanti et al. [1997], Yeh and Yeh [1998], Reveliotis [2000], Wu and Zhou [2005] for related studies.

In Li et al. [2010], we have proposed a set of traffic rules, required to be checked with when any vehicle intends to leave any zone, to realize the avoidance of deadlocks and inter-vehicle collisions. Applying the traffic control to this case study, it is ensured that (1) no inter-vehicle collision can happen (2) each vehicle can complete any finite number of tasks, with any route for each task, if the route of the last task ends up with a depot (meaning that the vehicle terminates its operation by parking in the depot). We refer the reader to Li et al. [2011] for the details of the traffic control.

An appealing advantage of the proposed traffic control is that it offers a large extent of freedom in routing the AGVs. In fact, the route for each task of a vehicle can even be established online rather than necessarily fixed beforehand. Specifically, before a vehicle arrives at some zone, it can freely choose its next zone for the route as long as it will reach its current destination in finite zones. Another merit of the traffic control is that it is time-efficient: the time complexity is linearly dependent on the number of vehicles in the system.

4.2 Routing algorithm

As mentioned in the previous subsection, before a vehicle arrives at some zone c it has the freedom to choose the next zone for its route out of the neighboring zones of c. But from Assumption 1 we know that if c is an off-crossing zone, then c has only one neighboring zone. Therefore, the only case of interest is when c is an at-crossing zone. Roughly speaking, the routing algorithm introduced here ensures that the vehicle always chooses a zone such that it will be closer to the destination of its current task at the resulting next crossing. As a direct consequence, this guarantees the vehicle reach the destination in finite zones. Beyond this prime property, the algorithm also intends to “smooth” the motion of the vehicles as much as possible.

Now suppose that \( c_d \) is the destination zone of the current task of some vehicle, which may be some zone of a QC, a YSs or the depot. Then before arriving at some at-crossing zone c of crossing \( r \in \mathcal{R} \), the vehicle needs to select a zone out of the zones in \( \mathcal{Y}_c = \mathcal{S}_r := \{ c^e_i : l_i \in \mathcal{O}_r \} \), where \( \mathcal{O}_r \) is the set of out-lanes of crossing \( r \), and \( c^e_i \) denotes the EZ of the lane \( l_i \). For any zone \( c^e \), we use \( d_m(c^e, c_d) \) and \( x_m(c^e, c_d) \) to represent the length of the shortest route(s) between zone \( c^e \) and \( c_d \) and the smallest number of crossings a vehicle has to pass when running from \( c^e \) to \( c_d \).

Before we state our routing algorithm, define two zone sets:

\[
\Gamma_1 = \{ c^e_i \in \mathcal{S}_r : d_m(c^e_i, c_d) < d_m(c, c_d) \}, \\
\Gamma_2 = \{ c^e \in \Gamma_1 : c^e \text{ is available} \}.
\]

where \( c^e_i \) is the EZ of the lane \( l_i \). (By assumption 1(a), \( c^e_i \) is either an at-crossing zone or the EZ of the in-lane of the depot); and \( c^e \) is “available” means that there is no vehicle in \( c^e \), moving from some zone to \( c^e \), or moving from \( c^e \) to some zone. In addition, let

\[
\Gamma_3 = \begin{cases} 
\{ c^e \in \Gamma_2 : c^e = \min_{c^e} d_m(c^e, c_d) \}, & \text{if } \Gamma_2 \neq \emptyset; \\
\{ c^e \in \Gamma_1 : c^e = \min_{c^e} d_m(c^e, c_d) \}, & \text{otherwise}.
\end{cases}
\]

### Online Vehicle Routing Algorithm

**Step 1:** If \( c_d \) is on some lane \( l_k \in \mathcal{O}_r \), then choose the zone \( c^e_k \) otherwise go to Step 2.

**Step 2:** If \#\( \Gamma_3 \neq 1 \), then choose the single zone in \( \Gamma_3 \); otherwise go to Step 3.

**Step 3:** Let \( \Gamma_4 = \{ c^e \in \Gamma_3 : c^e = \min_{c^e} x_m(c^e, c_d) \} \). If \#\( \Gamma_4 \neq 1 \), then choose the single zone in \( \Gamma_4 \); otherwise go to Step 4.

**Step 4:** If there is a zone \( c^e \) in \( \Gamma_4 \) to reach which the vehicle does not need a turn, then \( c^e \) is selected; otherwise choose randomly a zone from \( \Gamma_4 \).

**Remark 2.** In each layout of the road network designed in Subsection 3.2, the set \( \Gamma_1 \) cannot be empty; and such a \( c^e \) in Step 4, if exists, must be unique.

Note that to implement the routing algorithm above, one needs to maintain for each crossing \( r \) and each possible destination zone \( c_d \) the data sets \( \{ d_m(c^e, c_d) : c \text{ is any at-crossing zone of } r \} \), \( \{ d_m(c^e_i, c_d), x_m(c^e_i, c_d), d_m(c^e, c_d) : \text{for any } l_i \in \mathcal{O}_r \} \). But for each given layout, these data can be generated and stored offline before the system starts running. Therefore, the time complexity of the proposed routing algorithm is very small.
5. SIMULATION STUDIES

In this section, via computer simulations, we investigate the performance of the AGV systems with the designs and controls presented in the preceding content.

5.1 Task dispatching scheme

Here we introduce a simple stochastic task dispatching scheme that is used to drive the simulations. An in-depth discussion on this topic is beyond the scope of the paper. See Steenken et al. [2004] and Stahlbock and Voß [2008b] for recent progress in the research on this subject.

There are three types of tasks for each AGV:

1. traveling from the depot to a zone of a QC or YS (first task only);
2. traveling across the TA, i.e., from a zone of a QC to a zone of a YS or the other way around;
3. traveling from a zone of a QC or YS to the depot (last task only).

For the tasks of the second type, the source and destination stations are not arbitrarily chosen. Instead, according to a typical container transshipment scenario, the 20 QCs and 66 YSs are divided into 5 groups respectively. The groups of the QCs are disjoint, each of which consists of 4 QCs; while the groups of the YSs are overlapping with each group containing 24 YSs. For each vehicle, the source and destination zones for any task of the second type are randomly selected respectively from the zones of the related QC and YS groups (see Figure 5).

In addition, for the tasks of types (1) and (2), after a vehicle arrives at the destination zone of each task, it will stay stationary in the zone for 30 seconds for a container loading or discharging process before being assigned a new task.

![Group of 4 quay cranes related to 24 yard stackers and vise versa](image)

Fig. 5. Task dispatching scheme

5.2 Performance measures

Several performance measures are defined here in different aspects of concern. The first measure is related to the throughput of the terminal, called “average tasks performed per hour” (ATP), which is defined as

\[
ATP = \frac{3600}{ATT}, \tag{1}
\]

where ATT denotes the “average task time” calculated by

\[
ATT = \frac{\sum_{i=1}^{N} \sum_{j=1}^{M_i} T_{ij}}{\sum_{i=1}^{N} M_i}, \tag{2}
\]

with \(T_{ij}\) the time (in second) for vehicle \(v_i\) to complete its \(j\)th task. (Recall that \(N\) is the number of vehicles; and \(M_i\) is the number of tasks for vehicle \(v_i\).)

From an energy cost point of view, another performance measure of interest is the “average travel distance” (ATD) whose definition mimics that of ATT with \(T_{ij}\) replaced by \(D_{ij}\), the travel distance for the \(j\)th task of \(v_i\).

The last two performance measures, namely “average tardiness” (ATR) and “standard deviation of the tardiness” (STDTR), indicate how much are the motions of the AGVs delayed by the traffic. The formula for computing the ATR has the same form as (2) but with \(T_{ij}\) substituted by \(D_{ij}\).

\[
D_{ij} = T_{ij} - \min \text{ time required to complete task } j \text{ of } v_i
\]

Here the second term on the right hand side is the time for \(v_i\) to complete the task via the shortest route while assuming \(v_i\) the only vehicle running in the workspace.

The STDTR is computed by

\[
STDTR = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{M_i} (D_{ij} - ATR)^2}{\sum_{i=1}^{N} M_i}}. \tag{3}
\]

5.3 Simulation results

We run simulations with \(N = 20, 30, \ldots, 120\) and \(M_i = 50\) for any \(i \in \mathbb{Z}_N^+\). In order to obtain adequate precision of analysis, we repeat the simulations for each layout for 20 times, and take the average value for each performance measure. Because of the page limit, we refer the reader to Li et al. [2011] for the numerical results.

We see from the simulation results that the layout with four roads and big crossings remains the best choice for this wide range of the size of the vehicle team. This can be roughly explained as follows: As mentioned in Subsection 3.2, the use of small crossings may have less vehicles waiting at the crossings in the TA (not much less since simultaneous passings of a big crossing by two vehicles are not necessarily conflict with each other). However, in average, the travel distance of the vehicles in this case is considerably larger than that in the case of using big crossings. The use of six roads, compared with four roads, renders more space (lanes, zones) for vehicles’ movement. But this benefit is offset by increased travel distances and number of vehicles competing for crossings.

In addition, we note that using the layout with four roads and big crossings gives moderate ATR and STDTR even for a large team of operational vehicles, which implies that in this case (1) the traffic congestion is not too bad; and (2) one can estimate for each task the arriving time of the vehicle performing the task with an acceptable error.

6. CONCLUSIONS AND DISCUSSIONS

In this paper, we give a solution to the design and control of the AGV system for an automated container terminal. The popular zone control approach is used here to ease the traffic management. Various road network layouts are designed based on the practical dimension of
a container terminal. A traffic control scheme as well as a routing algorithm are developed. Computer simulations demonstrate the efficiency of our results in terms of some defined performance measures.

For an interesting comparison, we also did computer simulations with free ranging AGVs, where the collision avoidance is realized by the so-called potential field based approaches [Whitcomb et al., 1992]. We refer the reader to the conclusion part of Li et al. [2011] for our findings.

There are several possible directions for further research. First, in our traffic control strategy, it is required that each lane of the road network must contain at least two zones. To somehow weaken this restriction may be important for some applications where the workspace of the AGVs is very limited. In addition, one could think of a relaxation of the token-holding requirement in the traffic control scheme so that multiple vehicles can leave different at-crossing zones simultaneously, and hence the performance of the AGV system can be improved. As far as we could tell, using local crossing tokens, instead of a global one required in this work, and adding inter-crossing communications may lead to a successful trial.

The designs of the layout of the road network and the routing algorithm are vital to the performance of the AGV system. The designs presented in this paper are partially heuristic with the help of intuitions and practical experiences. Better results could be obtained by setting up and solving formal optimization problems. Along this line of thinking, the balance between the optimality and real-time applicability of the solutions becomes critical for practical applications.

In this work, the tasks of the AGVs are assigned in a purely random manner. We predict a room for performance improvement in the future if a sophisticated task dispatching strategy could be incorporated, although we have seen only a moderate tardiness for a typical number of operational AGVs with our current strategy. Such a dispatching strategy is desired to work collaboratively with the routing algorithm, and utilize the feedback of the real-time traffic flow information.

REFERENCES


