Comparing Deadlock Detection and Avoidance Policies in Automated Storage and Retrieval Systems

Mariagrazia Dotoli
Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari
Bari, Italy
dotoli@deemail.poliba.it

Maria Pia Fanti
Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari
Bari, Italy
fanti@deemail.poliba.it

Giorgio Iacobellis
Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari
Bari, Italy
giorgio.iacobellis@poste.it

Abstract - This paper focuses on real time control of Automated Storage and Retrieval Systems (AS/RSs), that is in charge of making decisions on resource allocation and scheduling choices. In these systems deadlock conditions may occur when parts require a set of resources in a circular wait situation. We model a generic multi-product AS/RSs serviced by rail guided vehicles as a discrete event system. Next, we compare two different real time deadlock solution strategies for AS/RSs: a deadlock avoidance strategy previously proposed by one of the authors and a deadlock detection/recovery strategy, proposed in the related literature. The considered control strategies are compared by means of a simulation of a case study.

Keywords: Automated storage and retrieval systems, real time control, deadlock detection, deadlock avoidance, discrete event simulation.

1 Introduction

Automated Storage and Retrieval Systems (AS/RSs) are widely used for material handling in warehouses [6, 7]. A typical AS/RS comprises several aisles with storage racks on either side, each served by an automated stacker crane, operating storage and retrieval of the parts. Cranes move in three directions: along the aisle to perform transfers, sideways between the aisle and the racks, and vertically to reach the Storage/Retrieval (S/R) location. Each aisle is also served by a storage and by a retrieval conveyor. Moreover, the AS/RS may include Rail Guided Vehicles (RGVs), transporting parts [5]. Finally, several input (storage) and output (retrieval) buffer stations, where the RGVs load or deposit pallets, are located in the system. The benefits of AS/RSs include low labor cost, low inventory cost, enhanced space exploitation, improved material tracking and high system throughput. However, advantageous operation of an AS/RS closely depends on the efficiency of the adopted control policies.

This paper focuses on real time control at the operational level of AS/RSs serviced by RGVs. The related literature usually deals with classical operational problems, such as analyzing the storage and retrieval sequencing policies [6, 7]. However, few contributions address resource allocations problems such as conflicts and deadlocks. Deadlock is a circular wait situation [3], where a set of jobs waits for never-to-be released resources. Hence, a deadlock occurrence can stop the operation of (a part of) the system and heavily affect the AS/RS performance. In this paper we carry out a performance-based comparison of two operational control strategies of AS/RSs. The first control policy, previously proposed by one of the authors [2, 3], rules the events relevant in the system behavior and guarantees an efficient system performance using a flexible deadlock avoidance policy. In addition, we employ a deadlock recovery strategy, proposed in the related literature [5], that detects deadlock occurrences and recovers the system behavior utilizing buffers to store deadlocked jobs. A discrete event simulation of a case study shows the efficiency of the proposed avoidance technique in comparison with the deadlock detection resolution strategy.

2 System description and case study

2.1 System description

We consider a large scale AS/RS (see Figure 1) that includes bidirectional (S/R) conveyors, storage conveyors, retrieval conveyors, stacker cranes, a Rail-Guided Vehicle System (RGVS), storage stations and retrieval stations [6]. We call \( J = \{ j_k : k=1,\ldots,N_J \} \) the set of all the parts to be stored or retrieved in the system and \( V = \{ v_h : h=1,\ldots,N_V \} \) and \( G = \{ g_i : i=1,\ldots,N_G \} \) the sets of RGVs and stacker cranes available in the system, respectively. In the sequel we denote RGVs and cranes as system vehicles. The set \( U = J \cup V \cup G \) denotes the complete set of the system users requiring resources. The RGVS rail is divided into disjoint zones, each representing a location adjacent to a storage or a retrieval conveyor, to the input or output stations, or else to a section of the RGVS rail-track (see Figure 1). Each zone of the RGVS is a resource that vehicles can acquire...
and is denoted by \( r_i \) for \( i=1,\ldots,N_Z \), where \( N_Z \) is the number of zones in the RGVS. In addition, we generically call resources the storage and retrieval stations as well as the unidirectional and bi-directional conveyors. In the following we indicate with \( r_i \) for \( i=N_Z+1,\ldots,N_Z+N_R \) such material handling resources, that are \( N_R \) in number. Moreover, all the aisle positions, where each crane can move, are resources that the corresponding stacker crane can acquire. These are denoted by \( r_i \) for \( i=N_Z+N_R+1,\ldots,N_Z+N_R+N_A \), where \( N_A \) is the overall number of crane positions in the aisles, including the vertical, horizontal and sideways shifts, as well as the home positions (i.e. the aisle positions where cranes are located when idle). Finally, let \( N_L \) be the total number of storage locations available in the racks at the sides of the aisles. We indicate with \( r_i \) for \( i=N_Z+N_R+N_A+1,\ldots,N_Z+N_R+N_A+N_L \) these resources that parts can acquire and with \( r_0 \) a fictitious resource modeling the output of the system. Hence, \( R = \{ r_i \mid i=0,\ldots,N_Z+N_R+N_A+N_L \} \) denotes the system resource set.

As each RGVS zone, aisle zone, transportation resource, or rack location can accommodate only one truck, crane or job, each resource in \( R \setminus \{ r_0 \} \) has unit capacity, while \( r_0 \) is always available and exhibits infinite capacity.

2.2 Case study

Consider the case study in Figure 1, inspired from an example reported in [5]. The figure shows the schematic plan layout of a multi-product AS/RS serviced by several RGVs. The RVGS fleet comprises \( N_V \) vehicles, traveling counterclockwise along the rail. The RGVS includes \( N_Z=23 \) zones (denoted by \( r_1,\ldots,r_{23} \)). In particular, \( N_R=16 \) zones in the RGVS are located next to a material handling resource each, while the remaining zones are sections of the rail track. More precisely, the AS/RS includes two storage (\( r_{24} \) and \( r_{27} \)) and two retrieval (\( r_{25} \) and \( r_{26} \)) stations, five storage conveyors (\( r_{30}, r_{32}, r_{34}, r_{36}, r_{38} \)), five retrieval conveyors (\( r_{29}, r_{31}, r_{33}, r_{35}, r_{37} \)) and two bidirectional conveyors (\( r_{28} \) and \( r_{39} \)), for a total of \( N_R=16 \) material handling resources. Moreover, \( N_G=7 \) narrow aisles are located in the system. Each aisle includes a rail with 35 positions, along which the corresponding crane travels, beginning from its home position (\( r_{40},\ldots,r_{46} \)). The aisles comprise two racks each with 665 locations partitioned in 19 flats, for a total of 14 racks and \( N_L=9310 \) locations. Moreover, each crane can move sideways and vertically to serve the racks, for a total of 1365 positions in each of the seven corridors. Hence, \( N_A=9555 \) crane positions are singled out in the AS/RS. Finally, the fictitious resource \( r_0 \) represents the system output. We remark that for the sake of simplicity in Figure 1 only some locations are shown and most racks and aisle positions labels are neglected.

3 The system model

A path \( p_0 \), i.e. a sequence of resources, is assigned to each job entering the system for a retrieval or storage operation. The dynamics of an AS/RS is described by the flow of jobs in process and by the movements of the vehicles traveling in the system. In particular, we can model such a system as a timed discrete event dynamical system [2] whose state at a time \( t \), denoted by \( q(t) \), contains information about the set \( J_q \) of jobs currently in process, the operation \( p(j) \) pertaining to each job \( j \in J_q \) and the residual path of each job \( r_p(j) \), i.e., the residual sequence of resources that each job \( j \in J_q \) has still to visit before ending its path.

Obviously, the first resource in \( r_p(j) \) is the resource currently held by \( j \). Moreover, if a storage task is assigned to part \( j \) (say \( p(j) \)), \( j \) is loaded by the input station with the assigned route, i.e. a path starting from the storage station, following a number of RGVS zones up to the one next to the storage conveyor servicing the aisle where the selected rack is located. The task terminates with crane selection and indication of the destination position in the rack.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1}
\caption{Plan layout of a multi-product AS/RS serviced by RGVs (\( N_V=3 \))}
\end{figure}
Moreover, the controller books an idle RGV, say \( v \), to load the part. When \( v \) loads the part, the path \( p(v)=rp(j) \) is assigned to the vehicle that carries the job and arrives to the storage conveyor. Hence, the part is mechanically loaded from the RGV onto the storage conveyor, and it is picked up by the narrow aisle crane \( g \in G \) that has been requested by the supervisor to transport the part to the assigned rack location. At this point the storage path \( p(g)=rp(j) \) is assigned to crane \( g \). At storage completion the stacker crane returns to the corresponding narrow aisle home position (single cycle crane operation).

Similarly, if job \( j \) is to be retrieved, the path \( p(j) \) is assigned to it and the appropriate narrow aisle crane \( g \in G \) is activated by the supervisor to travel to the rack location. Contemporaneously, an RGV \( v \in V \) is booked and the path \( p(v) \) is assigned so that it can reach the job to load. Then, the crane, following the path \( p(g)=rp(j) \) loads the part and carries it to the retrieval conveyor. After loading the part on the booked truck, the RGV assumes the same path of the job, i.e., \( p(v)=rp(j) \), to transport the part from the narrow aisle retrieval conveyor to the output station.

Hence, the system state \( q \) collects data concerning the users in set \( U \), the operation and the residual operation (storage or retrieval) associated with each job, the truck and crane carrying a job, the paths of a booked vehicle. The set of all the possible AS/RS states is indicated by \( Q \).

### 3.1 The system events

The following four types of events can occur:

- **type 1 event**: a job \( j \) enters the system for a storage or retrieval operation. The event is identified by the pair \( \sigma_1=(j,p(j)) \), where \( p(j) \) is the retrieval or storage operation assigned to part \( j \);
- **type 2 event**: a vehicle \( v \) or a crane \( g \) or a job \( j \) moves from a resource to another. The event is identified by the triple \( \sigma_2=(a,b,j) \), where \( a \in V \cup G \cup \{\emptyset\} \) and \( b \in Q \cup \{\emptyset\} \). Symbol \( a \in V \cup G \) represents a vehicle carrying the job \( j \in Q \) and it holds \( a=\emptyset \) if the job is not carried. Moreover, the symbol \( b \in Q \) represents the job carried by the vehicle \( a \in V \cup G \) and \( b=\emptyset \) means that the considered vehicle is idle or booked.
- **type 3 event**: a job \( j \) books a crane \( g \) or a vehicle \( v \) for a storage or retrieval operation. The event is identified by the triple \( \sigma_3=(g,j,p(g)) \) or \( \sigma_3=(v,j,p(v)) \), where \( p(g) \) and \( p(v) \) are the paths assigned to crane \( g \) or vehicle \( v \), respectively, to reach job \( j \) position;
- **type 4 event**: a job \( j \) leaves the system or is stored in a rack. This event is identified by the pair \( \sigma_4=(j,r_m) \), where \( r_m \in \{r_0,r_1,\ldots,r_N\} \) with \( i=N_f+N_m+N_l+1 \).

### 4 Review on the deadlock characterization

Deadlock is a circular wait situation [3], where a set of jobs waits for never-to-be released resources. This paper addresses deadlock in AS/RSs using the approach developed by one of the authors in [3] for automated manufacturing systems and subsequently modified in [2] for Automated Guided Vehicle Systems (AGVSs).

A crucial point to analyze deadlock conditions consists in properly describing the interactions between users and resources. To this aim, a direct graph (digraph) \( D_4(q)=[N, E_4(q)] \), called residual path digraph, synthetically represents the possible resource allocations considering the residual operations of the jobs in the system and the residual paths of the booked vehicles. In particular, the vertex set \( N \) represents the resource set of the system. Hence, the same symbols indicate vertices and system resources, i.e., \( N=R \). Moreover, the edge set \( E_4(q) \) includes edge \( e_{m,n} \) if and only if (iff) \( r_m \) immediately follows \( r_i \) in some residual path \( rp(j), rp(v), rp(g) \) with \( j \in J_q, v \in V \) and \( g \in G \), respectively.

The peculiar information contained in state \( q \) can be exhibited by means of a second digraph \( D_3(q)=[N, E_3(q)] \), named transition digraph. While the vertex set still coincides with the resource set, the edge set \( E_3(q) \) is contained in \( E_4(q) \). Hence, \( D_3(q) \) is a sub-digraph of \( D_4(q) \). In particular, edge \( e_{m,n} \) is in \( E_3(q) \) iff in state \( q \) a job \( j \in J_q \) or a booked vehicle of \( V \cup G \) holds \( r_i \) and requires \( r_m \) as next resource. Since vehicles carrying jobs follow the path assigned to the carried part, such vehicles are taken into account to build the digraphs considering the job in progress. As shown in [3], in a system characterized by resources with unit capacities, deadlock conditions are related to cycles of the transition digraph, as stated by the following proposition.

**Proposition 1.** State \( q \) is a deadlock for the system iff there exists at least one cycle in \( D_4(q) \).

We remark that 3-type and 4-type events are always enabled because they can not determine deadlocks or collisions. Hence, the control actions to avoid deadlock may consist of inhibiting or enabling the occurrence of some 1-type or 2-type events, depending on the current state \( q \). Starting from **Proposition 1**, a simple deadlock avoidance policy can be defined, based on a look-ahead procedure of only one step. In particular, when a type 1 or type 2 event driving the system from state \( q \) to \( q' \) has to occur, the policy builds the new transition digraph \( D_3(q') \) and inhibits the event iff such a digraph contains a cycle. As shown in [3], this control algorithm might lead the system to a restricted deadlock state. Such a situation can occur only if the cycles of \( D_4(q) \) enjoy a particular
property that can be exhibited using a further digraph, $D^2_\mathcal{S}(q)=[\mathcal{N}^2_{\mathcal{S}}(q),E^2_{\mathcal{S}}(q)]$, named residual second level digraph obtained from $D_\mathcal{S}(q)$ as follows. Denoting by $\{\gamma_1, \gamma_2, \ldots, \gamma_m\}$ the complete set of the cycles of $D_\mathcal{S}(q)$, we associate a vertex $\gamma_i\in\mathcal{N}^2_{\mathcal{S}}(q)$ to each cycle $\gamma_i$ of $D_\mathcal{S}(q)$. Moreover, an edge $e_i(k)$ is in $E^2_{\mathcal{S}}(q)$ iff the following two conditions hold: a) $\gamma_h$ and $\gamma_k$ have only one vertex in common (say $r_m$); b) there exists a path assigned to a job $j\in\mathcal{J}$ or a booked vehicle of $G\cup V$ requiring resources $r_1$, $r_m$ and $r_p$ in strict order of succession and $e_{\text{emp}}$ is an edge of cycle $\gamma_h$ while $e_{\text{emp}}$ is an edge of cycle $\gamma_k$.

Now, let $\gamma^2$ be a cycle in $D^2_\mathcal{S}(q)$ and let $\Gamma^2(q)$ be the subset of second level cycles enjoying the following property: $\gamma^2\in\Gamma^2(q)$ iff the cycles associated with the vertices of $\gamma^2$ are all disjoined but for one vertex, common to all of them. Moreover, let the capacity of a cycle $\gamma$ (denoted by $C(\gamma)$) be defined as the number of resources involved in such a cycle. Analogously, let us define the capacity of a second level cycle $C(\gamma)$ as the number of distinct resources involved in all the cycles corresponding to the vertices of $\gamma^2$. Finally, let $C^2_{\mathcal{S}}(q)$ be the minimum capacity of the second level cycles from $\Gamma^2(q)$ ($C^2_{\mathcal{S}}=\infty$ if $\Gamma^2(q)$ is empty). The following proposition is proven in [3], then modified for AGVs in [2] and $n_v(q)$ indicates the number of operations assigned to jobs in state $q$.

**Proposition 2.** State $q$ is a restricted deadlock for the system only if $\Gamma^2(q)$ is not empty and $n_v(q)>(C^2_\mathcal{S}(q)-1)$.

5. **The deadlock avoidance policy**

The Deadlock Avoidance Policy (DAP) for the AS/RS is defined specifying two control functions called resource validation and storage/retrieval validation. More precisely, we introduce the control functions $f_1: \Sigma_1Q\rightarrow\{0,1\}$, where $f_1(\sigma_1,q)=0$ ($f_1(\sigma_1,q)=1$) means that for the AS/RS in the state $q$ event $\sigma_1\in\Sigma_1$ is inhibited (enabled).

5.1 **Resource validation**

The resource validation function enables or inhibits 2-type events to avoid collisions and immediate deadlock. Denoting by $q'$ the state that the AS/RS will reach from $q$ as a consequence of the resource acquisition, the resource validation controller consists of a one-step look-ahead procedure testing whether $q'$ is a deadlock or not. The following function $f_1$ carries out the resource validation:

\[
f_1(\sigma_2,q)=\begin{cases} 1 & \text{if the second zone of } r_p \text{ is not empty} \\ 0 & \text{others}
\end{cases}
\]

The following proposition is proven in [3], then modified for AGVs in [2] and $n_v(q)$ indicates the number of operations assigned to jobs in state $q$.

**5.2 Storage/retrieval validation**

The storage/retrieval function enables or inhibits 1-type events in order to avoid immediate deadlocks and restricted deadlocks as long as a new path is assigned to a job. Let $p$ be the storage or retrieval operation that the scheduler proposes for the job $j\in J$ and let $q$ be the current system state. Moreover, let $q'$ indicate the state the system will reach if the event $\sigma_2=(j,p(j))$ occurs. The following function $f_2$ defines the storage/retrieval validation:

\[
f_2(\sigma_1,q)=\begin{cases} 1 & \text{if } D_1(q') \text{ contains no cycle and } n_v(q)<(C^2_\mathcal{S}(q)-2) \\ 0 & \text{others}
\end{cases}
\]

To state the deadlock avoidance policy, we consider RGVs and cranes as users, while the resources are zones, retrieval and storage stations, conveyors. On the other hand, trucks and cranes can be viewed as resources that jobs can acquire. However, deadlock does not involve trucks and cranes as resources because the utilized booking policy guarantees that a retrieval and storage operation begins only if there are an idle truck and an idle crane. However, some particular deadlock situations involving unidirectional conveyors and booked trucks can occur. The following example describes such a condition.

**Example.** Suppose that a retrieved job $j_1$ is on a unidirectional retrieval conveyor $r_1$ waiting for vehicle $v_1$ previously booked. The crane $g$ is occupied by another retrieved job $j_2$ to be unloaded on the busy conveyor $r_1$ and subsequently loaded by a booked vehicle $v_2$. Hence, $j_2$ is blocked. At the same time, suppose that job $j_3$ is blocked on the aisle storage conveyor $r_2$ because it has to acquire $g$ to reach its destination in the rack. Finally, vehicle $v_3$, that is ahead the other vehicles on the rail track, is waiting for the busy conveyor $r_1$, which has now unloaded a piece $j_4$. Hence, the waiting truck $v_1$ holds the unit capacity RGVs zone adjacent to the storage conveyor, so that the booked vehicles $v_1$ and $v_2$ cannot proceed to their destination.

Obviously, by limiting the number of jobs that may acquire resources corresponding to unidirectional conveyors the described example can not occur. Hence, to solve the deadlock situation described in the previous example, the function $f_2$ is modified as follows:

\[
f_2(\sigma_1,q)=\begin{cases} 1 & \text{if the following conditions are verified: } D_1(q') \text{ contains no cycle, } n_v(q)<(C^2_\mathcal{S}(q)-2), \text{ for each } g\in G \text{ serviced by two unidirectional conveyor, there exist at most two jobs } j_1, j_2 \in \mathcal{J} \text{ so that } g \text{ is in } r_p(j_1) \text{ and } r_p(j_2). \\ 0 & \text{others}
\end{cases}
\]
6 Simulation results

6.1 Description of the simulation experiments

This section presents a simulation study, performed for the case study described in section 2, employing in turn the DAP defined in the previous section and the deadlock detection policy (DDP) proposed in [5]. More precisely, deadlock detection/recovery algorithms allow deadlock to occur and detect and recover for deadlock using appropriate buffers to store deadlocked jobs.

We assume incoming operations to have been sorted by an appropriately designed scheduler. In particular, the scheduler activity is modeled by randomly generated S/R requests obtained via an exponential distribution of mean 150. Moreover, the following scheduling options are considered [7]. The incoming requests are sequenced according to the first come first served criterion. Parts requiring the conveyors, or input/output stations, RGVs and cranes are serviced according to the first come first served criterion. The single command crane operation is considered, i.e. cranes wait for an assignment in their home positions, where they return after executing a task.

The selected run time is T=28800 time units, corresponding to 8 hours if we associate one time unit to one second, for a set of 30 replications of each test. The RGVs fleet size is varied from NV=3 to NV=6 vehicles, and the vehicles speed is changed from v=0.25 m s\(^{-1}\) to v=1 m s\(^{-1}\), in order to test the DAP and DDP Control Policies (CPs) under traffic volume changes. With reference to Figure 1, the length of each RGVs zone is reported in Table 1 and the system characteristics are reported in Table 2 [5]. The AS/RS is implemented in the Arena environment [4], in which it is possible to integrate modeling with the execution of C++ computation routines (e.g., developing the DDA and the DDP). Hence, by simulating the case study controlled in turn by the selected deadlock control policies, we investigate, for equivalent settings, which policy performs better guaranteeing that the S/R operations are performed and neither conflicts nor deadlocks affect the system performance.

6.2 Simulation results

The main index employed to compare the effectiveness of the two CPs is the system throughput, i.e., the number of completed S/R tasks in the run time under the selected CP. Such an index assesses the CPs efficiency in having the system execute S/R incoming requests while minimizing blockings. Clearly, only completed missions are considered in the throughput computation. The measure of performance of vehicles activity is the average percentage time an RGV spends in each of its states, i.e.: (i) traveling booked, (ii) traveling loaded, (iii) idle.

Figures 2, 4 and 6 depict the AS/RS throughput, obtained under the considered CPs using different RGVs fleet sizes, when the trucks velocity equals respectively 0.25 m s\(^{-1}\) (Figure 2), 0.5 m s\(^{-1}\) (Figure 4) and 1 m s\(^{-1}\) (Figure 6). The figures show that the system throughput increases when the trucks speed is augmented, since the average operation time decreases correspondingly. It can be inferred from the figures that under both CPs the transportation system is inefficient for a modest RGVs fleet size, since the number of vehicles is small compared both with the AS/RS size and the large number of tasks to be completed in the assigned run time. Under both real time control strategies, the homogeneous increase of the allocated missions is an obvious consequence of the growth of the available RGVs number. However, the throughput value reaches a maximum for a given fleet size, since traffic becomes congested and vehicles are often blocked in order to avoid deadlock and collisions. In particular, comparing the system throughput obtained under the DDP and under the DAP shows that, although for a low trucks speed the deadlock detection and deadlock avoidance CPs tend to perform equally (see Figure 2), an increase in the vehicles speed corresponds to a noticeable improvement of the DAP throughput with respect to the DDP (see Figures 4 and 6). The DAP allows a larger number of completed tasks than the DDP.

### TABLE 1

<table>
<thead>
<tr>
<th>RGVS zone</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Length [m]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3.07</td>
<td>2.22</td>
</tr>
<tr>
<td>2</td>
<td>1.30</td>
<td>1.30</td>
</tr>
<tr>
<td>3</td>
<td>1.30</td>
<td>1.30</td>
</tr>
<tr>
<td>4</td>
<td>1.30</td>
<td>1.30</td>
</tr>
<tr>
<td>5</td>
<td>1.30</td>
<td>1.30</td>
</tr>
<tr>
<td>6</td>
<td>1.30</td>
<td>1.30</td>
</tr>
<tr>
<td>7</td>
<td>1.30</td>
<td>1.30</td>
</tr>
<tr>
<td>8</td>
<td>1.30</td>
<td>1.30</td>
</tr>
</tbody>
</table>

### TABLE 2

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter-arrival rate for storage operations</td>
<td>Expo(150)</td>
</tr>
<tr>
<td>Inter-arrival rate for retrieval operations</td>
<td>Expo(150)</td>
</tr>
<tr>
<td>Velocity of cranes (horizontal, vertical)</td>
<td>(1.67, 1) m s(^{-1})</td>
</tr>
<tr>
<td>Number of RGVs</td>
<td>3, 4, 5, 6</td>
</tr>
<tr>
<td>Speed of RGVs</td>
<td>1, 0.25, 0.5, 1 m s(^{-1})</td>
</tr>
<tr>
<td>Speed of conveyors</td>
<td>0.21 m s(^{-1})</td>
</tr>
<tr>
<td>Conveyors length</td>
<td>4.25 m</td>
</tr>
<tr>
<td>Conveyors width</td>
<td>1.3 m</td>
</tr>
<tr>
<td>Distance between non adjacent conveyors</td>
<td>1.7 m</td>
</tr>
<tr>
<td>Total number of racks locations (racks x rows x columns)</td>
<td>(14 x 19 x 35) = 9310</td>
</tr>
</tbody>
</table>

The selected RGVs dispatching rule is the nearest vehicle policy. More precisely, once a vehicle has completed a task, it remains idle holding the last resource (i.e., the RGVs zone) that it acquired. When an operation is requested, a test is performed to check whether at least a truck in the RGVs fleet is inactive, and in case more than one vehicle is idle then if a storage (retrieval) is demanded the idle truck nearest to the corresponding storage station (retrieval conveyor) is booked.

The selected run time is T=28800 time units,
The average RGVs activity under both CPs is depicted in Figures 3, 5 and 7 with varying trucks speed. The reported results are very similar, showing that the vehicles activity is not particularly affected by the RGVS fleet size, the RGVS velocity and the control policies.

7 Conclusions

This paper focuses on real time control of Automated Storage and Retrieval Systems (AS/RSs). AS/RSs are modeled as discrete event systems where users compete to acquire resources so that deadlock situations can occur. Revisiting a Deadlock Avoidance Policy (DAP) proposed by one of the authors for automated guided vehicle systems, a deadlock avoidance strategy is specified. Moreover, the DAP is compared with a deadlock detection/resolution technique suggested in the related literature. A discrete event simulation for a case study shows the effectiveness of the DAP compared to the deadlock detection technique. Future research has to explore integration of the proposed deadlock avoidance strategy with scheduling policies.

References


