Research Article

Collision Avoidance Strategy for Multivehicle Conflict on Common Rail

Bo Fang, Cheng Qiu, Ming Feng, Wei Liang, and Ximing Zhang

1 School of Mechanical Engineering, University of Science and Technology Beijing, Beijing 100083, China
2 China Academy of Machinery Science and Technology Co., Ltd., Beijing 100044, China
3 ST Microelectronics Inc., Beijing 100088, China

Correspondence should be addressed to Bo Fang; fangbo@mtd.com.cn

Received 4 March 2022; Revised 12 April 2022; Accepted 15 April 2022; Published 3 May 2022

1. Introduction

In the logistics transportation system of the intermediate warehouse of the steel plant, for the sudden situation that the common rail multivehicles are easy to collide and lock up when they are running with each other, the conflict detection and collision avoidance of the multivehicle dynamic path can effectively ensure the operation safety of the whole multivehicle common rail transportation system. At present, the related research in this field mainly focuses on preventing multitarget collision or blockage through spatio-temporal segmented partition topology or priority strategy. For example, Nenad [1] uses time division to generate multiobjective conflict-free paths. Fanti and Turchiano [2] and others solve the conflict and collision problem of system operation by distinguishing the workspace of multiple vehicles. Yeh [3] uses the banker algorithm to solve the conflict and blockage problem of multiobjective operation system. Wang et al. [4] and others used the idea of limiting the scope of task assignment to solve the driving conflict and solved the task sequence through the memetic algorithm to deal with the multivehicle operation problem with space-time constraints. Jia et al. [5] and others proposed the control strategy of AGV multivehicle path conflict avoidance based on an improved A* algorithm.

The above research on multiobjective collision avoidance mostly focuses on the application of three-dimensional storage and transportation, crane lifting, and AGV mobile robot, focusing on each trolley serving the corresponding operation area. However, there is less research and application of multivehicle cooperative collision avoidance of rail parent-child nested multivehicle transportation system in the same operation area. Therefore, on the basis of previous similar studies, find a dynamic avoidance strategy that is
more in line with the multivehicle common rail conflict of the system at the same time and space and effectively and actively prevent the collision or blockage of this kind of conveying system. Based on the improved deadlock avoidance banker algorithm, this paper attempts to detect the conflict and collision of the multivehicle transportation system, realize the dynamic path planning, and achieve the active prevention of the operation conflict of the whole intermediate warehouse transportation system without affecting the flexible and changeable combination operation of multiple vehicles. At the same time, in the avoidance link, the combination of dynamic priority transmission and time window offset is used to eliminate all kinds of sudden collision conflicts in multivehicle common rail operation, and its effectiveness is proved by example analysis and literature method comparison.

In this paper, an enhanced banker algorithm is used to perceive active encounter in the common rail multivehicle operation path, formulate the time window adjustment policy of dynamic priority transmission to eliminate multivehicle common rail collision conflict, and institute the system conflict detection and impact avoidance model, which effectively reduces the possibility of system conflict and collision without affecting the diversity of multivehicle procedures.

The paper arrangements are as follows:

Section 2 describes the multiobjective running conflict dynamic detection based on an improved banker algorithm. Section 3 evaluates the time window collision avoidance strategy based on dynamic priority transmission. Section 4 discusses and verifies the contents of the operating process of conflict avoidance strategy by the solving example.

2. Multiobjective Running Conflict Dynamic Detection Based on Improved Banker Algorithm

Banker’s algorithm is the most famous deadlock-avoiding algorithm, originally named for ensuring the safety of bank lending. It evolved from a deadlock-avoiding algorithm designed by Diesterra for T.H.E. The basic principle is based on the allocation strategy of the bank lending system. The operating system is regarded as a banker. The resources managed by the operating system are equivalent to the funds managed by the banker. The operating system allocates resources according to the rules formulated by the banker. When the process requests resources, first check whether the sum of the number of resources occupied and the resources applied for exceeds the maximum demand for resources, and refuse to allocate resources if it exceeds; if it does not, then check if the system’s existing resources meet the maximum required resources; if they do, allocate resources based on the current number of applications or postpone the allocation.

According to its principle, this paper evaluates and detects the conflict security of dispatched routes using statically planned mother-car routes as a system process. Since the operation scenario of this transportation system is a dynamic process and divided into different stages, in order to improve the response processing speed of the deadlock avoidance algorithm, it satisfies the flexible and cooperative operation of multichild and multiparent vehicles. This paper tries to simplify the traditional two-dimensional matrix process security retrieval into binary encoding process security retrieval and divides the system operation process checking into storage and retrieval jobs for conflict detection in stages according to the operation characteristics of the intermediate library transportation system. This improved banker algorithm reduces the amount of data retrieval from both process data structure and partition process checking, reduces unnecessary process security checks, thus reducing the calculation cycle of banker conflict detection algorithm as a whole, and improves the response speed.

The data structure in the banker’s algorithm is divided into resource available matrix (Available), maximum demand matrix (Max), allocation matrix (Allocation), need demand matrix (Need), and request matrix (Request). This paper draws on and improves the conflict detection database, which only refers to the available resource matrix (Available) and request matrix (Request). Therefore, define the dynamic path conflict algorithm data structure of the system so that set \( R \) is all the path resources of the system, that is, the request matrix in the banker algorithm, the route resources of \( R_i \) for the static planning of the \( i \)-th parent car, and the route resources occupied by \( Ni \) for the static planning of the \( i \)-th parent car, and set \( U \) is the system available path resources, that is, the available resource matrix in the banker algorithm. For node resources that are connected between route segments, make them belong to the resource of the previous route segment, which contains all the conflicts involved in the operation of system resources. Instead of using a matrix, the above system resources are represented using binary encoding. The first binary encoding represents the route segment, the last binary encoding indicates the route node, and the resource of the route segment occupied by the requested process (parent car) is represented by one, otherwise by zero. For example, for the system available resource \( Uk \) in the \( k \)-th job scenario, assuming it contains 16 route segments (with nodes) resources, it is represented as

\[
U_k = (00 00 01 10 00 10 00 11 00 00 01 00 00 00 10 00).
\]  

Segments 4, 6, 7, and 15 of UK resources are occupied, and nodes 3, 8, and 11 are occupied. In addition to the above optimization of the data structure and data search of the checked resources, the improved banker algorithm also sets the dynamic vector \( W[i] \) for the temporary check of intermediate resources to facilitate the dynamic creation and deletion of the check of resource occupancy by the request process. In addition, it simplifies the conditions under which process resources cannot be preempted and requested to be kept under the principle of resource data conflict checking. Its specific algorithm process and constraints are not detailed, and the main improvements in the algorithm process are as follows:
At the beginning of the algorithm checking resource conflict security, make \( W[i] = \text{set} \{U\} \) and set process resource checking state vector \( D[i] \). Execute the process when \( R_i \leq N_i \) is satisfied; otherwise, it needs more resources than it needs, terminating the exit.

When checking data in the resource collection as requested, \( R \{} \leq W[i] \), make \( D[i] = \text{false} \). If \( R_i \leq U_i \), execute the process; otherwise, a resource conflict occurs, enter the process, and wait.

After the process has successfully obtained the resources, modify the vectors \( W[i] \) and \( D[i] \) to make
\[
W[i] + = \text{Allocation}[i];
D[i] = \text{true}.
\]

The system will allocate resources to the requesting process and modify Available and Allocation to
\[
\text{Available}[i] - = \text{Request}[i];
\text{Allocation}[i] + = \text{Request}[i].
\]

When the checking state \( D[i] = \text{true} \) of all resource processes indicates that the system is in a secure state, the system has not completed all security checks, or there is an unsafe state.

According to the results of the static route planning for each parent-child multivehicle, the route resource \( R_i \) and the currently occupied route resource \( N_i \) of each vehicle process are generated. By comparing the corresponding route segments between the same task scenario and the parent vehicle process resource collection \( [R] \) at the same time, the conflict of collision and blockage among multiple vehicles during the system operation can be found.

### 3. Time Window Collision Avoidance Strategy Based on Dynamic Priority Transmission

As an example, illustrated in the local sketch of the multivehicle transportation network of the intermediate warehouse in Figure 1, there are several paths to choose from the starting point \( \star \) to the ending point. The shortest path is \( L_4^{R_4} - R_4^{L_4} - R_4^{L_3} \), and its corresponding path resource \( R_i = (L_4^{R_4} - L_4^{R_2} - L_4^{R_3} - R_4^{L_3} - R_4^{L_2}) \); if the corresponding path resource is occupied in the current time period, \( N_i = (00 00, 100) \). Comparing whether the same route resource is occupied by more than two processes in the same job period, it is detected that there is a conflict between the route resources \( L_4^{R_2} \) and \( R_4^{L_3} \), and a collision avoidance decision is required.

According to the results of the static route planning for each parent-child multivehicle, the route resource \( R_i \) and the currently occupied route resource \( N_i \) of each vehicle process are generated. By comparing the corresponding route segments between the same task scenario and the parent vehicle process resource collection \( [R] \) at the same time, the conflict of collision and blockage among multiple vehicles during the system operation can be found.

As an example, illustrated in the local sketch of the multivehicle transportation network of the intermediate warehouse in Figure 1, there are several paths to choose from the starting point \( \star \) to the ending point. The shortest path is \( L_4^{R_4} - R_4^{L_4} - R_4^{L_3} \), and its corresponding path resource \( R_i = (L_4^{R_4} - L_4^{R_2} - L_4^{R_3} - R_4^{L_3} - R_4^{L_2}) \); if the corresponding path resource is occupied in the current time period, \( N_i = (00 00, 100) \). Comparing whether the same route resource is occupied by more than two processes in the same job period, it is detected that there is a conflict between the route resources \( L_4^{R_2} \) and \( R_4^{L_3} \), and a collision avoidance decision is required.

As an example, illustrated in the local sketch of the multivehicle transportation network of the intermediate warehouse in Figure 1, there are several paths to choose from the starting point \( \star \) to the ending point. The shortest path is \( L_4^{R_4} - R_4^{L_4} - R_4^{L_3} \), and its corresponding path resource \( R_i = (L_4^{R_4} - L_4^{R_2} - L_4^{R_3} - R_4^{L_3} - R_4^{L_2}) \); if the corresponding path resource is occupied in the current time period, \( N_i = (00 00, 100) \). Comparing whether the same route resource is occupied by more than two processes in the same job period, it is detected that there is a conflict between the route resources \( L_4^{R_2} \) and \( R_4^{L_3} \), and a collision avoidance decision is required.

As an example, illustrated in the local sketch of the multivehicle transportation network of the intermediate warehouse in Figure 1, there are several paths to choose from the starting point \( \star \) to the ending point. The shortest path is \( L_4^{R_4} - R_4^{L_4} - R_4^{L_3} \), and its corresponding path resource \( R_i = (L_4^{R_4} - L_4^{R_2} - L_4^{R_3} - R_4^{L_3} - R_4^{L_2}) \); if the corresponding path resource is occupied in the current time period, \( N_i = (00 00, 100) \). Comparing whether the same route resource is occupied by more than two processes in the same job period, it is detected that there is a conflict between the route resources \( L_4^{R_2} \) and \( R_4^{L_3} \), and a collision avoidance decision is required.

As an example, illustrated in the local sketch of the multivehicle transportation network of the intermediate warehouse in Figure 1, there are several paths to choose from the starting point \( \star \) to the ending point. The shortest path is \( L_4^{R_4} - R_4^{L_4} - R_4^{L_3} \), and its corresponding path resource \( R_i = (L_4^{R_4} - L_4^{R_2} - L_4^{R_3} - R_4^{L_3} - R_4^{L_2}) \); if the corresponding path resource is occupied in the current time period, \( N_i = (00 00, 100) \). Comparing whether the same route resource is occupied by more than two processes in the same job period, it is detected that there is a conflict between the route resources \( L_4^{R_2} \) and \( R_4^{L_3} \), and a collision avoidance decision is required.

As an example, illustrated in the local sketch of the multivehicle transportation network of the intermediate warehouse in Figure 1, there are several paths to choose from the starting point \( \star \) to the ending point. The shortest path is \( L_4^{R_4} - R_4^{L_4} - R_4^{L_3} \), and its corresponding path resource \( R_i = (L_4^{R_4} - L_4^{R_2} - L_4^{R_3} - R_4^{L_3} - R_4^{L_2}) \); if the corresponding path resource is occupied in the current time period, \( N_i = (00 00, 100) \). Comparing whether the same route resource is occupied by more than two processes in the same job period, it is detected that there is a conflict between the route resources \( L_4^{R_2} \) and \( R_4^{L_3} \), and a collision avoidance decision is required.

As an example, illustrated in the local sketch of the multivehicle transportation network of the intermediate warehouse in Figure 1, there are several paths to choose from the starting point \( \star \) to the ending point. The shortest path is \( L_4^{R_4} - R_4^{L_4} - R_4^{L_3} \), and its corresponding path resource \( R_i = (L_4^{R_4} - L_4^{R_2} - L_4^{R_3} - R_4^{L_3} - R_4^{L_2}) \); if the corresponding path resource is occupied in the current time period, \( N_i = (00 00, 100) \). Comparing whether the same route resource is occupied by more than two processes in the same job period, it is detected that there is a conflict between the route resources \( L_4^{R_2} \) and \( R_4^{L_3} \), and a collision avoidance decision is required.

As an example, illustrated in the local sketch of the multivehicle transportation network of the intermediate warehouse in Figure 1, there are several paths to choose from the starting point \( \star \) to the ending point. The shortest path is \( L_4^{R_4} - R_4^{L_4} - R_4^{L_3} \), and its corresponding path resource \( R_i = (L_4^{R_4} - L_4^{R_2} - L_4^{R_3} - R_4^{L_3} - R_4^{L_2}) \); if the corresponding path resource is occupied in the current time period, \( N_i = (00 00, 100) \). Comparing whether the same route resource is occupied by more than two processes in the same job period, it is detected that there is a conflict between the route resources \( L_4^{R_2} \) and \( R_4^{L_3} \), and a collision avoidance decision is required.
system in the intermediate depot can be divided into the following.

3.1. Node Collision Conflict. Node collision will occur when there is more than one trolley running at the same time at the nodes connected by different path segments as shown in Figure 2.

3.2. Collision in Opposite Directions. When multiple trolleys run in opposite directions on the same path segment, collision in opposite directions may occur as shown in Figure 3.

3.3. Collision in the Same Direction. There is more than one trolley running in the same direction on the same path segment; if the speed of the front vehicle relative to the rear vehicle slows down or stops, it is possible to have a collision in the same direction at a certain time as shown in Figure 4.

Formulating corresponding collision avoidance strategies for the above different collision types to effectively prevent the conflict locking of system operation is an important link of dynamic path planning of common rail multivehicle transportation system. At present, when choosing the multivehicle path collision avoidance strategy, most studies choose the collision avoidance strategy of avoiding the high priority vehicle path with low priority. The priority strategy is widely used in solving the conflict problem of multi AGV path planning [8, 9]. For the steel plant parent-child nested common rail multivehicle system when performing the access task, the collision conflict mainly focuses on the characteristics of multivehicle common rail or multivehicle handover [10]. The time window collision avoidance strategy with priority transmission is adopted to eliminate the collision blockage of multivehicle operation in the process of system operation, which can better solve the collision conflict of this kind of multivehicle common rail transportation system. Firstly, it is necessary to clarify the priority of the operation task of the subparent vehicle of this multivehicle system. According to the operation stage of the operation task, each operation stage is given different priority [11]. Generally, the operation task of saving/taking volume can be divided into five stages:

<table>
<thead>
<tr>
<th>Phase</th>
<th>Status</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>No-load waiting</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>No-load operation</td>
<td>2</td>
</tr>
<tr>
<td>c</td>
<td>Load waiting</td>
<td>3</td>
</tr>
<tr>
<td>d1</td>
<td>Load caching</td>
<td>4</td>
</tr>
<tr>
<td>d2</td>
<td>Load taking out</td>
<td>5</td>
</tr>
<tr>
<td>e</td>
<td>Loading/unloading</td>
<td>6</td>
</tr>
</tbody>
</table>

a: no-load waiting; b: no-load operation; c: load waiting; d: load operation; e: loading/unloading. Among them, load operation is divided into d1 (load caching); d2 (load taking out). The static priority of operation tasks of child and parent vehicles is summarized in Table 1.

In Table 1, the higher the priority value of the task, the more important the current time of the task, the higher the operation level, and the less the active avoidance. The highest priority of operation in the operation stage is that, in the process of steel coil loading/unloading, for safety reasons, it cannot be affected by other operating trolleys, and other operations can be carried out only after the successful completion of this stage. For another example, the operation stages d1 and d2 are also the load operation stages. Considering the principle of first-in last-out for the steel coil intermediate depot, the operation priority of d2 is higher than that of d1, which ensures the consistency and smoothness of the logistics direction of the steel coil intermediate depot.

However, in the actual operation process of the system, there is a dynamic transformation process in each task stage, such as no-load waiting stage a to no-load operation stage b, load waiting stage c to d1 or d2, and no-load operation stage b to coil loading stage. There are priorities of different tasks in the same stage. To ensure the timeliness and authenticity of the priority collision avoidance strategy, it is necessary to establish a dynamic priority function in the task stage based on the static priority category in Table 1 to clarify the priority relationship of multiple vehicles with different tasks in the same stage in time and avoid collision conflict. The approach is based on the RT (remaining completion time) of various tasks in the same stage. The smaller the value, the greater the dynamic value of task priority, and vice versa. For example, for task \( i \) from stage \( k \) to stage \( k + 1 \), the \( k + 1 \) completion time \( RT_k = T_k^b - T_k^0 \), and the \( k + 1 \) completion time \( RT_{k+1} = RT_k + 1 - T_{k+1} \). It is assumed that the task stage in time \( t \) is from stage \( k \) to stage \( k + 1 \). Then, the comprehensive priority of multiple vehicles undertaking task \( i \) in \( t \) time can be expressed as

\[
P_i^t = P_i^t + \frac{(RT_k + 1 - RT_k)}{RT_k + 1 - RT_k} \times \text{Rank} \left( \frac{\text{t}}{T+i} \right). \tag{4}
\]

Formula (4) is the comprehensive priority of the task stage obtained after the static priority is superimposed with the dynamic priority. \( P_i^t \) indicates the static priority of multivehicle task \( i \); \((RT_k + 1 - RT_k)/(RT_k + 1 - RT_k)\) indicates whether the priority change is positive increase or negative decrease according to the minimum priority.
principle of remaining time RT when switching from stage \( b \) to stage \( \hat{d} \); \((\text{Rank}(i, t))/(S_i + 1))\) means that the rank sorting function is used to sort the multiple vehicles of task \( I \) in time \( t \) from the largest to the smallest according to the remaining completion time. The value returned by the function is the sequence number of vehicles in the sequence, and \( S_i \) is the number of multiple vehicles running task \( i \) in time \( t \). Therefore, \( \text{Rank}(i, t) \leq S_i \), then \((\text{Rank}(i, t))/(S_i + 1)) < 1\), and its dynamic priority change is \( \Delta P < 1 \). In this way, the change of dynamic priority will not exceed the scope of static priority and follow the static priority allocation criteria of subparent vehicle operation tasks in Table 1.

It is also required to import the security mechanism of multivehicle priority level by level transmission after creating the comprehensive priority function of task phase. That is, when there is a conflict between adjacent subparent workshops running on the same track, it is necessary to coordinate the operation status of adjacent subparent vehicles, maintain the safe distance of multiworkshop operation, and realize passive safety assurance [12, 13]. The principle is to transfer the status of the highest priority subparent vehicle running on the same track to the low-level subparent vehicle at both ends of the track and judge whether there is operation conflict for multiple vehicles running on the same track from both positive and negative directions, that is, if there is a low priority multivehicle operation within the safe distance of high priority multivehicles. Then, the priority and operation status of the low priority multivehicle shall be temporarily adjusted step by step until being consistent with the highest priority multivehicle. [14, 15].

The specific priority transfer process is exemplified, as shown in Figure 5. The priority of common rail multivehicle is transferred level by level. The operation direction defines the logistics direction of the main channel from left to right and the buffer channel from top to bottom as forward operation (\( \longrightarrow \)) and vice versa as reverse operation (\( \longleftrightarrow \)) and stop state (\( \longleftrightarrow \)). It is specified that the running direction of the parent vehicle with the highest priority is the active direction, and the opposite running direction is the passive direction. It is assumed that the adjacent distance between common rail subparent trains is above the safe distance, the task priorities of subparent trains \((i, i + 1, i + 2, i + 3)\) are \((p_6, p_2, p_3, p_4)\), respectively, and the operation direction in a certain period is \((\longrightarrow, \longleftrightarrow, \longleftrightarrow)\). If the priority of vehicle \( i \) is greater than \( i + 1 \), the priority and operation status of \( i + 1 \) will be adjusted to be consistent with that of vehicle \( i \), and the priority will be temporarily changed to \( p_6 \). Similarly, the priority of \( i + 2 \) and \( i + 3 \) is lower than \( p_6 \). After the priority is transferred from high to low, it is changed to \( p_6 \), and the operation state is changed to passive, which is consistent with that of \( i \). The current task priority of the final subparent vehicle \((i, i + 1, i + 2, i + 3)\) is \( p_6 \), the status is forward operation (\( \longrightarrow \)), only \( i \) is active operation, and other subparent vehicles are in temporary passive operation. If the common rail subparent vehicle is \((i + 1, i + 2, i + 3)\) and the highest priority is \( p_4 \) of \( i + 3 \), then through the reverse transmission of priority, the last multivehicle state becomes reverse operation (\( \longleftrightarrow \)), the temporary priority is \( p_4 \), \( i + 2 \) and \( i + 3 \) are active operation state, and \( i + 1 \) is passive operation state.

Priority transfer mechanism is the last barrier of passive collision avoidance strategy and only intervenes when there is operational conflict after system dispatch planning. However, if priority adjustment involves more adjacent cars, priority transfer step by step will have a chain effect on multivehicle walking and reduce operating efficiency. Therefore, the above priority transfer strategy only adjusts the temporary priority when there is a running state conflict between the adjacent parent subvehicles. For more common rail conflicts of parent subvehicles, the introduction of collision avoidance control with multivehicle driving time window adjustment is an effective means to deal with passive conflict resolution of multivehicle operation, which can
better ensure efficient and safe operation for the multivehicle system on common rail [16–18].

Time window adjustment collision avoidance strategy is to adjust the running time of the route of multivehicle conflict segment to achieve the method of eliminating collision conflict. It calculates the running time window by calculating the period from driving into one or more route segments to driving out [19]. Considering the consistency of system operating parameters, ignoring the acceleration and deceleration process of multivehicles, and making the average running speed equal, the safe passing time of the subcar \(i\) from \(R_i\) to \(R_j\) on the main channel \(L_i\) can be expressed as

\[
TS_i = \left(\frac{DL_{R_iR_j}}{v}\right) + Tc. \tag{5}
\]

In formula (5), \(Tc\) represents the safe interval time, \(Tc = v/tsc\), \(tsc\) is the safety time factor, and its value range is \(0 < tsc < 1\), representing that the faster the multivar runs on the common rail, the longer the interval time between each other, so as to ensure that there is enough buffer parking time to avoid collisions between multiple vehicles. The \(V\) convention is the average running speed of the subcarrier in the channel. \(DL_{R_iR_j}\) represents the straight-line distance of the car through the route segment \(R_i\) to \(R_j\) on the main channel \(L_i\). For each parent subcar, only one segment path can be occupied in the same period of time, and conversely, the same segment path can only be occupied by one car. Therefore, the parent car \(i\) occupies the segment \(L_{R_iR_j}\) of the route for a certain period of time, and the time window \(TW_i\) can be expressed as

\[
TW_i = time_{\text{out}}^{in} - time_{\text{in}}^{in}. \tag{6}
\]

In formula (6), \(time_{\text{out}}^{in}\) is the moment when the parent car \(i\) leaves the route segment \(L_{R_iR_j}\), \(time_{\text{in}}^{in}\) is the moment when the parent car \(i\) enters the route segment \(L_{R_iR_j}\) and \(TW_i \geq TS_i\), because \(TS_i\) is the shortest safe time for the car to pass through the segment of the route, and \(TW_i\) is the time for the actual car to run on the segment of the route. During this period, the time window that the car passes through may become longer due to unexpected circumstances that slow down the passage time of the car.

Because different cars on the common rail perform different tasks and have different priorities, the time window distribution of multiple cars on a certain route segment is expressed in the form of multidimensional vectors. The first parameter in the vector corresponds to the car with the highest priority, and the last parameter corresponds to the car with the lowest priority. The dimension of all vectors is equal to the number of all the cars that pass through a segment path. Then, the path segment \(L_{R_iR_j}\)’s time window vector can be expressed as

\[
[TW_i] = \left[ Pi; time_{\text{in}}^{in}(i); time_{\text{out}}^{in}(i) \right]. \tag{7}
\]

In formula (7), \(Pi\) is ranked from highest to lowest priority. Based on the ideal path of each parent car obtained from the system static route planning, the safe passage time \(TS_i\) and the original time window vector \([TW_i]\) of each car through the same segment of the route can be obtained by formulas (5) and (7), respectively.

For the adjustment principle of overlapping time windows in case of conflict, the time windows of lower priority cars need to be moved back or the time windows of higher priority cars need to be moved forward. That is, adjust the time window entry time \(ntime_{\text{in}}^{in}(L_{R_iR_j})\) according to the higher priority of formula (8) or adjust the time window entry time \(ntime_{\text{out}}^{in}(L_{R_iR_j})\) with lower priority of formula (9) to stagger the entry time of the conflict path segment.

\[
ntime_{\text{in}}^{in}(L_{R_iR_j}) = time_{\text{in}}^{in}(L_{R_iR_j}) - TS_i, \tag{8}
\]

\[
notime_{\text{out}}^{in}(L_{R_iR_j}) = time_{\text{in}}^{in}(L_{R_iR_j}) + TS_i; \tag{9}
\]

According to the classification for collision conflicts of multivehicle system operation in the intermediate depot, the time window adjustment principles are as follows:

1. For node collision conflicts, since the overlap area of the collision is a certain junction node, the time window of the lower priority car is moved backward, and the time window entry time of the previous route segment before the node is adjusted backward. The amount of backward shift is set to the total safe transit time of the other higher priority car through the route segment of this conflict node.

2. For opposite collision conflicts, the conflict overlap area may be one or more route segments; in order to minimize the chain effect on the multivehicle operation of the system, the same route segment can only be occupied by one car at the same time. For cars occupying overlapping area for less than 1/2 of the total conflict time, the cars occupy overlapping area route segments first in order of occupying overlapping route segments from smallest to largest, while the other cars extend the time window until there are no cars occupying the overlapping area in the previous route segment node of the conflict route segment. For cars occupying overlapping areas that are longer than 1/2 of the total conflict duration, the total safe transit time of other cars passing through the route segment of this conflict node is set as the time window for cars with lower priority.

3. For collision in the same direction, there is no possibility of collision in the same direction because all the parent cars run at the same speed in the corridor. There is only the possibility that the front car may collide in the same direction by loading/unloading the subcar at the node or by passive concession waiting. As a result, regardless of the priority of the rear car in a same-direction conflict, a time window delay on the route segment before the conflict point is necessary, which is the safe transit time for the automobile in front of the conflict via the conflict path segment.
It is important to note that, by adjusting the time window forward, backward, or delayed collision avoidance strategies, the following optimal route time window vectors will be adjusted. Therefore, it is necessary to continuously detect and update the time windows of the corresponding parent car static route planning to avoid new running collision conflicts.

4. Operating Process of Conflict Avoidance Strategy

The contents of the above dynamic collision avoidance strategies for the multivehicle system are sorted out, and the process of dynamic collision avoidance is summarized as shown in Figure 6.

The detailed operation flow is described as follows:

**Step 1.** According to the static route planning given by the intermediate warehouse logistics transportation system, the improved banker algorithm is used to initialize the data of the route resource \( R_i \) and the currently occupied route resource \( N_i \) for each car process, forming the binary encoding process security retrieval.

**Step 2.** According to the running characteristics of the intermediate depot transportation system, divide the system operation process checking into two stages: storage and retrieval. If deadlock conflicts are identified, the conflict data is restored, and the conflict type is determined by comparing binary encoding data of resource availability matrix and request matrix. Detect banker exits until no conflicts are found.

**Step 3.** Based on the detected objects, if the number of objects in conflict is less than 3, the dynamic priority \( P_{ti} \) is transferred step by step according to formula (1) to eliminate the conflicts involving fewer objects.

**Step 4.** For cases where the number of conflicting objects is more than or equal to 3, the time window \( T_{Wi} \) of the route line is generated according to formula (6), and the data initialization of the time window vector \([T_{Wi}]\) of the route segment planning of the conflicting objects is completed according to formula (7).

**Step 5.** Adjust the offset and delay of the conflict object’s parent car time window according to formulas (8) and (9) according to different conflict types until the overlap of all route segments is eliminated and the time window adjustment procedure is quit.

4.1. Example Solving Analysis and Verification. Various simultaneous instances of collision conflicts are simulated in order to more naturally and variedly portray the collision conflicts that may arise when multiparent vehicles are running on the same track.

This will allow us to better test the upgraded banker algorithm’s accuracy and sensitivity in dealing with multivehicle conflict detection and verify the efficacy of a time window collision avoidance strategy based on dynamic priority transfer in resolving complicated multivehicle conflicts. The local multivehicle operation state diagram in \( T \) period is created using Siemens industrial control visual dynamic simulation programme Wincc, based on the distinct priorities and conflict kinds of multivehicle operation of the system as shown in Figure 7. In the \( T \) period, multivehicle operation range consists of 22 cache channels, 3 main channels, and limited coil buffer areas 1 and 2. There are parent subcars MC1, MC2, and MC3 running on main channel L1, MC4 and MC5 running on main channel L2, and MC6, MC7, and MC8 running on main channel L3. They perform different tasks and generate different priorities and path plans with specific parameters such as Table 2.

According to the operation state and parameters of the multicar system for steel coil logistics transportation in the intermediate warehouse in this local area in \( T \) period, the task path of the car MC1 to MC8 is described as follows:
MC1: during the $T$ period, the steel coil is transported along the main channel $L1$ in the same direction as the production logistics direction. In Figure 7, it passes through the path segment from left to right: $L1^R_{R6} - L1^R_{R7} - L1^R_{R8} \ldots L1^R_{R16} - L1^R_{R17} - L1^R_{R18}$. The dynamic priority of this stage is $P4$. The preset distance of path planning is $60$ m (the length between buffer rows is $5$ m, i.e., the shortest path segment distance of the main channel).

MC2: load the steel coil at node $L1^R_{R8}$ first in the $T$ period, and then pass through the path segment from left to right in the direction of production logistics along the main channel $L1$: $L1^R_{R8} - L1^R_{R9} - L1^R_{R10} \ldots L1^R_{R19} - L1^R_{R20} - L1^R_{R21} - L1^R_{R22}$. The dynamic priority of this stage is $P6 \rightarrow P4$. The preset distance of path planning is $90$ m (take-up position → outlet).

MC3: no-load from the current stop position is reversed to the production logistics direction along the main channel $L1$ in the $T$ period. In Figure 7, it passes through the path segment from right to left: $L1^R_{R18} - L1^R_{R17} - L1^R_{R16} \ldots L1^R_{R3} - L1^R_{R2} - L1^R_{R1}$. The dynamic priority of this stage is $P2$. The preset distance of path planning is $140$ m (current location → taking location).

MC4: the steel coil is transported along the main channel $L2$ in the same direction as the production logistics direction in the $T$ period. In Figure 7, it passes through the path segment from left to right: $L2^R_{R8} - L2^R_{R9} - L2^R_{R10} \ldots L2^R_{R19} - L2^R_{R20} - L2^R_{R22}$. The dynamic priority of this stage is $P5$. The preset distance of path planning is $95$ m.

MC5: in the $T$ period, no-load from the current stop position is reversed to the production logistics direction along the main channel $L2$. In Figure 7, it passes through the path segment from right to left: $L2^R_{R14} - L2^R_{R13} - L2^R_{R12} \ldots L2^R_{R6} - L2^R_{R5} - L2^R_{R4}$. and then stores loading steel coils at node $L2^R_{R3}$ and runs in the logistics direction to the outlet of the intermediate warehouse. The dynamic priority of this stage is $P2 \rightarrow P6 \rightarrow P5$. The preset distance of path planning is $>165$ m (current position → taking position → exit).

MC6: in the $T$ period, the steel coil is transported along main channel $L3$ in the same direction as the production logistics direction. In Figure 7, it passes through the path segment from left to right: $L3^R_{R2} - L3^R_{R3} - L3^R_{R4} \ldots L1^R_{R20} - L1^R_{R21} - L1^R_{R22}$. The dynamic priority of this stage is $P5$. The preset distance of path planning is $>100$ m (the whole process passes through the main channel $L3$ to the outlet of the intermediate warehouse).

MC7: load the steel coil at node $L3^R_{R9}$ in time period $T$, and then pass through the path segment from left to right along the same direction of main channel $L3$ in the direction of production logistics in Figure 7: $L1^R_{R10} - L1^R_{R11} - L1^R_{R12} \ldots L1^R_{R19} - L1^R_{R20} - L1^R_{R21}$. The dynamic priority of this stage is $P6 \rightarrow P5$. The preset distance of path planning is $90$ m (taking position → exit).

MC8: in the $T$ period, unload the steel coil at node $L3^R_{R16}$ first and then pass through the path segment from right to left along the $L3$ main channel opposite to the production logistics direction in Figure 7: $L1^R_{R16} - L1^R_{R15} - L1^R_{R14} \ldots L1^R_{R6} - L1^R_{R5} - L1^R_{R4}$. The dynamic priority of this stage is $P6 \rightarrow P2$. The preset distance of path planning is $>80$ m (storing position → outlet).

The improved banker algorithm is used to dynamically detect the conflict of the multivehicle running paths of the above example, generating the $R_{mc1}$ of the route resource of the car processes MC1–MC8 in period $T$ and the binary encoding of route resource $N_{L1}$ occupied the main channels L1, L2, and L3 and checking their security, respectively. The results are as follows.

$T$ period system operational status diagram multivehicle process resources running on main channel L1 are

\[
R_{mc1} = (0 0 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 00 00 00),
\]

\[
R_{mc2} = (0 0 00 00 00 00 00 00 00 01 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11),
\]

\[
\]
The resource $U_{L1}$ of the main channel $L1$ of the system operation state map at $t$ time contains 22 path sections (including nodes) resources, which are

$$N_{L1} = (00\ 11\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 11\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 11\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00).$$ (11)

Check security for set $\{U_{L1}\}$ of route segment resource Matrix Available $L1$. After comparing whether the same path resource was occupied by more than two processes within the same operation period, the result of the collision path section is

$$\{L1_{R13}; L1_{R14}; L1_{R15}; L1_{R16}; L1_{R17}\}.$$ (12)

$T$ period system operational status diagram multivehicle process resources running on main channel $L2$ are

$$R_{MC6} = (00\ 00\ 00\ 00\ 00\ 00\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 00\ 00\ 00\ 00\ 00\ 00).$$ (13)

The resource $U_{L2}$ of the main channel $L2$ of the system operation state map at $t$ time contains 22 path sections (including nodes) resources, which are

$$N_{L2} = (00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00).$$ (14)

Check security for set $\{U_{L2}\}$ of route segment resource Matrix Available $L2$. After comparing whether the same path resource was occupied by more than two processes within the same operation period, the result of the collision path section is

$$\{L2_{R10}; L2_{R11}; L2_{R12}; L2_{R13}; L2_{R14}\}.$$ (15)

$T$ period system operational status diagram multivehicle process resources running on main channel $L3$ are

$$R_{MC6} = (11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 11\ 00\ 00\ 00\ 00\ 00\ 00).$$ (16)

The resource $U_{L3}$ of the main channel $L3$ of the system operation state map at $t$ time contains 22 path sections (including nodes) resources, which are

$$N_{L3} = (00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00\ 00).$$ (17)

Check security for set $\{U_{L3}\}$ of route segment resource Matrix Available $L3$. After comparing whether the same path resource was occupied by more than two processes within the same operation period, the result of the collision path section is

$$\{L3_{R10}; L3_{R11}; L3_{R12}; L3_{R13}; L3_{R14}; L3_{R15}\}.$$ (18)

In order to verify the accuracy of the path segment conflict detected by the above-improved banker algorithm, establish the time window arrangement of the task process in the corresponding period, as shown in Figures 8–10. Define
the span length of different multivehicle, which represents the time length occupying the corresponding path segment and node, and the real-time window length, and the starting position of the time window of the vehicle corresponds to the starting time of its operation, the abscissa is time, and the ordinate is the area of different path sections. The comparison and analysis are carried out, respectively, and the avoidance strategies of time window adjustment and priority transmission are given.

As shown in Figure 8(a), the distribution diagram of multivehicle time window of channel $L_1$, from the position distribution of MC1, MC2, and MC3 in each time period of $t + \sum T_{Si}$ passing through the path segment $L_1 R_{11}-L_1 R_{12}$ in the main channel $L_1$, it can be seen that the position of MC1 and MC2 in the main channel increases with the time of $t$, while MC3 runs in the opposite direction, so there is a collision conflict in the opposite direction. That is, at time $t + 35$, the opposite collision conflict between MC2 and MC3 occurs in the path segment $L_1 R_{13}$. After MC2 and MC3 do not give way to each other, MC1 will collide with MC2 in the same direction at $t + 40$, and the conflict point is also $L_1 R_{14}$. This is consistent with the starting point of the conflict segment detected by the improved banker algorithm. The time window adjustment and priority transmission are combined by using an avoidance plan: adjust the priority of the MC3 task to be consistent with MC2 until MC2 reaches the exit and restore the original task and priority of MC3 to avoid the collision between MC2 and MC3. Moving the MC1 time

Figure 8: (a) Distribution of multivehicle time window of channel $L_1$. (b) Time window arrangement after multivehicle collision avoidance in channel $L_1$. 
window back for 10 s ensures that MC2 travels in the same direction with its guaranteed safe distance (2 buffer rows) after the steel coil is loaded at node $L_{1R8}$, avoiding the possibility of collision between MC1 and MC2 in the same direction at path segment $L_{1R8} - L_{1R9}$. Finally, as shown in Figure 8(b), the time window arrangement after multivehicle collision avoidance of channel $L_1$ is obtained.

As shown in Figure 9(a), the distribution diagram of multivehicle time window of the $L_2$ channel, from the position distribution of MC4 and MC5 in each period of $t + \sum T_{Si}$, passing through the path segment $L_{2R16} - L_{2R22}$ of main channel $L_2$, it can be seen that the position of MC4 in the main channel increases with the time of $t$, and MC5 runs in the opposite direction, so there is a collision conflict in the opposite direction. That is, at time $t + 20$, the path segment $L_{2R16}^{R11} - L_{2R12}^{R11}$ has the opposite collision conflict between MC4 and MC5, and the median value of the conflict path segment set detected by the improved banker algorithm is just $L_{2R11}^{R11} - L_{2R12}^{R11}$. When the number of conflict objects is less than 3, the avoidance strategy of priority transmission is adopted: the priority of MC4 is P5, and the priority of MC5 is P2. Therefore, the priority of the MC5 job task is adjusted to be consistent with that of MC4 until MC4 completes the job task and restores the original job task and priority of MC5. Finally, as shown in Figure 8(b), the arrangement of the time window after multivehicle collision avoidance in channel $L_2$ is obtained.

As shown in Figure 10(a), the distribution diagram of multivehicle time window of channel $L_3$, from the position distribution of MC6, MC7, and MC8 in each period of $t + \sum T_{Si}$, passing through the route segment $L_{3R1} - L_{3R22}$ of the main channel $L_3$, it can be seen that the position of MC6 and MC7 in the main channel increases with the time of $t$, while MC8 operates in the opposite direction. Therefore, there is a collision conflict between the former and its opposite direction. That is, at time $t + 25$, the path segment $L_{3R12}^{R11} - L_{3R13}^{R12}$ has the opposite collision conflict between MC7 and MC8, which is consistent with the median value of the conflicting path segment set detected by the improved banker algorithm. After MC7 and MC8 do not avoid each
other, MC6 will collide with MC7 in the same direction at $t + 60$, and the conflict point is also $L_3^{R12}_{R13}$. The avoidance method combines time window adjustment with priority transmission: at a safe distance before a collision, MC7’s priority is changed from $P_{6}$ to $P_{5}$, and the objective is to remove the loaded volume; MC8’s priority is changed from $P_{6}$ to $P_{2}$, and the task is no-load operation. Therefore, carry out priority transmission, adjust the priority of MC8 job task to be consistent with that of MC7 until MC7 reaches the exit, restore the original job task and priority of MC8, and move back its task time window for 30 s (6 buffer rows) to let the MC6 with priority $P_{5}$ pass through the exit, so as to avoid colliding with it. Finally, as shown in Figure 10(b), the arrangement of time window after L3 channel multivehicle collision avoidance is obtained.

The multivehicle operating time window arrangement for eliminating dynamic collision conflict after avoidance is produced by the above priority transfer and time window adjustment. It can be seen that the collision avoidance strategy combining priority transfer and time window adjustment can timely and effectively eliminate and avoid the dynamic conflict deadlock problem in the "last

![Figure 10](image-url)

**Figure 10:** (a) Distribution of multivehicle time window of channel L3. (b) Time window arrangement after multivehicle collision avoidance in channel L3.
In this paper, the improved banker algorithm is used to detect the active conflict in the common rail multivehicle operation path, formulate the time window change strategy of dynamic priority transmission to eliminate the multivehicle common rail collision conflict, and establish the system conflict detection and collision avoidance model, which effectively reduces the possibility of conflict and collision of the system without affecting the diversity of multivehicle operation paths. The paper proves that the priority transfer time window adjustment collision avoidance strategy based on improved banker algorithm has comprehensive advantages over the two previous algorithms and effectively improves the operation efficiency and safety of the final link of the multivehicle dispatching planning system for intermediation. The effectiveness of the priority transfer time window adjustment collision avoidance strategy based on the improved banker algorithm is verified through example analysis and comparison with literature methods, providing a reference for the efficient and intelligent operation of this type of common rail multivehicle logistics system.

### Data Availability

The data used to support the findings of this study are included within the article.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### References


