I.J. Education and Management Engineering **2012, 9, 22-28**
Published Online September 2012 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2012.09.04

Available online at http://www.mecs-press.net/ijeme

Research and Implementation on the Flexible Feasibility of AGV Routing

Cheng-duan Wang

*Computer and Communication EngineeringWeifang University Weifang261061, China***Abstract**

In order to solve the evaluation problem of automated guided vehicle system (AGVS) flexibility, a new feasibility evaluation approach is proposed according to the present research situation. This method can deal with the problem of AGVS rescheduling caused by provisional change of production tasks. The effectiveness of this method is also proved in this paper. Simulation result shows that the method can evaluate the reschedule feasibility under provisional situations. And the prospect of this method is also discussed in this paper. It is proved that the method is efficient and feasible.

Index Terms: automated guided vehicle system(AGVS);feasibility evaluation; conflict-free problem; flexibility

1. RESEARCH STATUS

C.W.Kim proposed an approach according to the conflict-free shortest time AGV routing in 1991 [1], which could select one shortest path from the current location to the destination. But this method was not good at dealing with the provisional situation. Samia proposed an improved method in 2001 [2], but that method still had the time delay. Therefore Samia proposed one robust algorithm in 2002 [3], that algorithm allows some vehicles to be ahead of time, and reduces the time delay. These methods do not make the flexibility to enhance as far as possible. This paper will aim at this kind of question and make the preliminary discussion.

2. BASIC CONCEPT

2.1. Time Window

AGV need take a long time to perform a specific action. This time set is called the time window of this particular action.

2.2. Buffer

The buffer is used to connect two different roads.

Corresponding author:
E-mail address: wangchengduan@163.com

2.3. Reserved Time Window

AGV need some time to stay in a buffer, which is called the reserved time window.

2.4. Free Time Window

AGV need take some time to start from a buffer through the specified path to another buffer, which is called the free time window.

2.5. Channel

The channel is referred to the road which connects two different buffers.

2.6. Flexible Feasibility

The flexible feasibility is referred to some AGV rescheduling under provisional situations.

3. EVALUATION METHOD

3. 1 Criterion of Conflict [4]

We represent a time window by a set of seven elements, that is, *TW* = {*n*, *t*, *bn*, *fn*, *bt*, *ft*, *bs*}, Where we refer to *n* as the AGV number, and *t* as the type of resource, and *bn* as the start node of resource, and *fn* as the end node of resource, and *bt* as the start time, and *ft* as the end time, and *bs* as the total number of AGV.

We implement this representation by using a two-dimensional array *TW*, that is, the abscissa axis represents the number of AGV and vertical axis represents the resource used by AGV.

Criteria are as follows:

① Suppose that *TW* [*i*] [*m*]. *bn* = *TW* [*t*] [*m*]. *fn*, *TW* [*i*] [*m*]. *fn* = *TW* [*t*] [*m*]. *bn* and *t* = 1. If *TW* [*i*] [*m*]. *bt* ≤*TW* [*t*] [*m*]. *bt* ≤*TW* [*i*] [*m*]. *ft*, AGV #*i* and AGV #*t* will use the same resources, which results in a conflict. Such a conflict comes from two different directions and it is shown in Fig. 1.

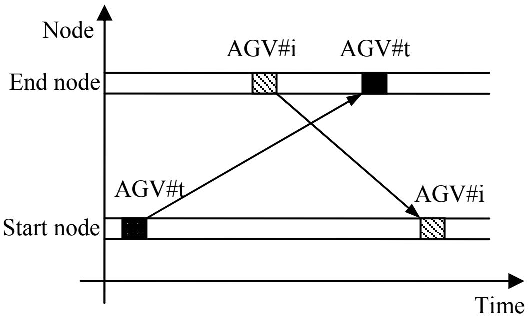


Figure 1. conflict from the different directions

②Suppose that *TW* [*i*] [*m*]. *bn* = *TW* [*t*][*m*]. *bn*, *TW*[*i*][*m*]. *fn* = *TW*[*t*][*m*]. *fn* and *t* = 1. If *TW*[*t*][*m*]. *bt* ≤ *TW*[*i*][*m*]. *bt* ≤ *TW*[*i*][*m*]. *ft*, AGV #*i* and AGV #*t* will use the same resource, which results in a conflict. Such a conflict comes from the same direction and it is shown in Fig. 2.

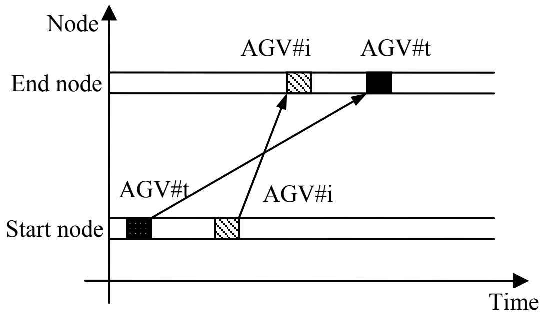
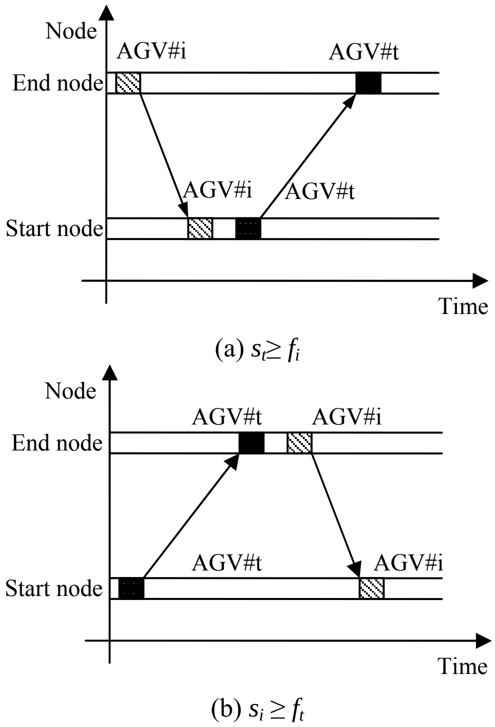


Figure 2. conflict from the same direction

③Suppose that $bs = 1$ in ① and ②. If $bs \neq 1$, the case is similar. If the number of elements in column m for array TW is less than TW [] [m]. bs, there is no conflict. Otherwise, there is some conflict.

3. 2 Compatible Activities [5]

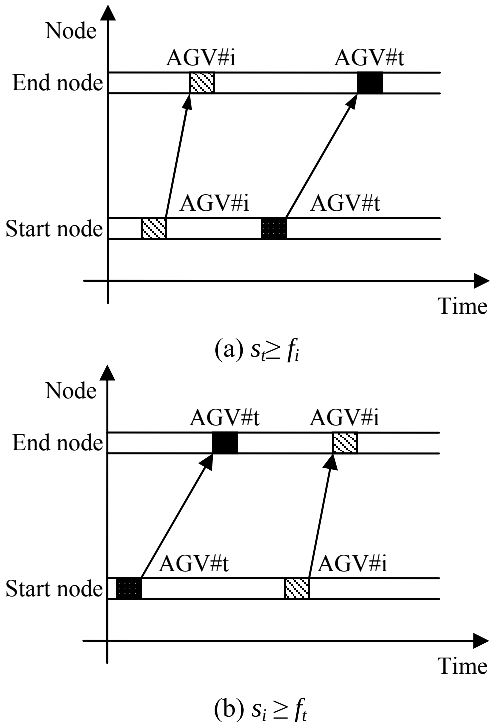
Let $E_m = \{1, 2, ..., n\}$ be a set with n activities, where all the activities want to use the same channel m. And each activity has one start-time and one end-time, respectively written s_i and f_i, such that $s_i < f_i (1 \leq i \leq n)$. If you choose the activity i, then AGV #i may use channel m in the time window $[s_i, f_i)$. If $[s_i, f_i)$ and $[s_t, f_t)$ are not joint ($1 \leq i,t \leq n$), then AGV #i and AGV #t can use channel m. In this case, activities i and t are compatible activities. That is, if $s_i \geq f_t$ or $s_t \geq f_i$, AGV #i and AGV #t can pass through channel m, as shown in Fig. 3 and Fig. 4.



(a) $s_t \geq f_i$

(b) $s_i \geq f_t$

Figure 3. compatible activities of the different direction



(a) $s_t \geq f_i$

(b) $s_i \geq f_t$

Figure 4. compatible activities of the same direction

Judge all channels one by one which will be used by the AGV *#i*. If the AGV *#i* can use such every channel, then the original plan is feasible. Otherwise, which is rescheduled.*the Basic Thought of Method*

The initial task will be recombined using greedy technology. After being combined, the task has characteristics to maximize the system transportation capacity.

(1)When AGV is at work, at regular intervals, it will report its current position, the number of a channel which will be used by AGV, the free time window (start-time, end-time) and other information, to the host computer.

(2)After receiving data, PC will convert the start time and end time into integer units and store them by arrays *s* and *f*, in which the elements are sorted by end-time in ascending order.

(3)Use the greedy algorithm for arrangements. PC will allow the AGV to pass through that path, which is near the path. If an AGV can not pass through that path, PC will arrange AGV to pass through other path or wait in buffer.

(4) According to the proposal, we will give an arrangement for transportation tasks.

3. 4 Algorithm Implementation

```
#define TRUE 1

#define FALSE 0

int Pass_t(timeinterval T[ ChannelNum ] [ AGVNum ], int t ,int tpath [ ], int n){

int i, j, k, m, b[ ChannelNum ];

for(k=0 ; k<n ; k++){

i = tpath[k];
b[i] = FALSE;
m = 0;
T[i][m].tag = TRUE;
for (j=1 ; j<AGVNum ; j++)
if (T[i][j].starttime >= T[i][m].finishtime) {
T[i][j].tag = TRUE;
m=j;
}
else
T[i][j].tag = FALSE;
for ( j=0 ; j<AGVNum ; j++)
if (T[i][j].agvnumber = t && T[i][j].tag=TRUE) {
b[i]=TRUE;
break;
}
}
for(i=0 ; i< ChannelNum ; i++)
if(b[i]==FALSE)
return  FALSE ;
return TRUE;
}
```

4. EXAMPLE

One production system is shown in Fig. 5. Here we represent the buffer as a square and represent a channel as a dotted line.

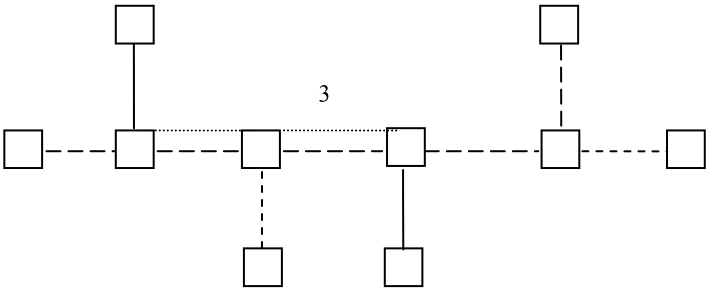


Figure 5. work station graph

Select channel 3 in Fig. 1 as an example to illustrate the implementation process of this method, the initial tasks is shown in Table 1.

TABLE I. INITIAL TASKS

AGVNum	1	2	3	5	6	7	8
s_i	3	1	8	3	2	5	12
f_i	8	4	11	5	13	7	14

Sort the tasks by f_i in ascending order, as shown in Table 2.

TABLE II. RESCHEDULED TASKS

AGVNum	**2**	5	**7**	1	**3**	6	**8**
s_i	**1**	3	**5**	3	**8**	2	**12**
f_i	**4**	5	**7**	8	**11**	13	**14**

As shown in Table 2, there are four AGVs that will be allowed to use the channel 3, their numbers are 2, 7, 3 and 8, respectively. For those AGVs not be allowed to use channel 3, PC will arrange them to wait or select other channel to pass through.

5. ANALYSIS OF METHOD

In different cases, running time is shown in Table 3. The time complexity of algorithm is O (n). This method can ensure the good real-time.

TABLE III. RUNNING TIME IN THE SITUATION

ChannelNum	running time(unit s)
4	0.310
5	0.316
7	0.626

If the design of the initial task is inconsequent for a section of the path, then after calculating there is only one AGV that will be allowed to pass through, which indicates that the initial task is reasonable. But the system still needs to compare n times that took a lot of calculation time. In this case, we should modify the initial transportation scheme. The defect of scheme is cannot avoid waiting, if on the basis of this method, we may develop a method that can further reduce waiting time and maximize the system's transportation capacity, which is more perfect.

The method not only can be applied to the real-time task planning, also can be applied to the initial task planning.

6. CONCLUSION

For large-scale AGVS arranging transport task is very effective by using this method. Simulation experiment data show that the online decision-making performance of this method is very excellent.

In an emergency, this method still can guarantee the maximization of system transportation capacity.

REFERENCES

[1] C.W.Kim, J.M.A.Tanchoco. Conflict-free shortest time bi-directional AGV routing[J].International Journal of Production Reasearch.1991，29(12):2377-2391.

[2] Samia Maza, Pierre Castagna. Robust conflict-free AGV routing in bi-directional network[C]. Proceedings of 8th IEEE International Conference on Emerging Technologies and Factory Automation，2001(2):761 – 764.

[3] Samia Maza，Pierre Castagna. Robust conflict-free routing of bi-directional automated guided vehicles systems[J],2002 IEEE International Conference on Man and Cybernetics,2002 (10):6～12.

[4] Qiu L,Hsu W j,Huang S Y.Scheduling and Routing Algorithms for AGVS:a Survey[J].International Journal of Production Research,2002,40(3):745-760.

[5] Yoo J W,Sim E S,Cao C.An Algorithm for Deadlock Avoidance in an AGV System[J].International Journal of Advanced Manufacturing Technology,2005,26(5):659-668.