Conflict-free dynamic route multi-agv using dijkstra Floyd-warshall hybrid algorithm with time windows

Solichudin, Aris Triwiyatno, Munawar A. Riyadi
Department of Electrical Engineering, Faculty of Engineering, Diponegoro University, Indonesia

Article Info

Article history:

Received Sep 9, 2019
Revised Dec 29, 2019
Accepted Jan 11, 2020

Keywords:

Conflict-free route
Dynamic programming
Hybrid algorithm
Multi-AGV
Time windows

ABSTRACT

Autonomous Guided Vehicle is a mobile robot that can move autonomously on a route or lane in an indoor or outdoor environment while performing a series of tasks. Determination of the shortest route on an autonomous guided vehicle is one of the optimization problems in handling conflict-free routes that have an influence on the distribution of goods in the manufacturing industry's warehouse. Pickup and delivery processes in the distribution on AGV goods such as scheduling, shipping, and determining the route of vehicle with short mileage characteristics, is very possible to do simulations with three AGV units. There is a windows time limit on workstations that limits shipping. The problem of determining the route in this study is considered necessary as a multi-vehicle route problem with a time window. This study aims to describe the combination of algorithms written based on dynamic programming to overcome the problem of conflict-free AGV routes using time windows. The combined approach of the Dijkstra and Floyd-Warshall algorithm results in the optimization of the closest distance in overcoming conflict-free routes.

Corresponding Author:

Solichudin,
Department of Electrical Engineering,
Diponegoro University,
Prof. Soedarto, SH. Street, Tembalang, Semarang, 1269, Indonesia.
Email: sudin3007@gmail.com

1. INTRODUCTION

Vehicle transportation systems based on AGVs have been developed for decades. They are technically advanced and complex. There are many tactical and operational issues which have to be addressed. Decisions related to the design stage have a very large impact on future system performance [1]. This area includes issues such as:

- guide-path design
- estimating the number and location of parking, pick-up and delivery points,
- estimating the number of required vehicles

The second area of problems relating to system management includes the following:

- positioning of idle vehicles,
- vehicle dispatching,
- vehicle routing,
- vehicle scheduling,
- collision and deadlock avoidance

One such interpretation involves the following problem: if a delivery decision is made, the route and schedule must be planned for another AGV, that the purpose of AGV scheduling is to send an AGV set and the route mission is to find a suitable route [2]. But on another interpretation, determining the vehicle route must be in accordance with the order of stations that must be visited by this vehicle. Scheduling also determines the vehicle's time when it has to take and send loads [3]. Sometimes dispatching, routing and scheduling are handled separately. However, this problem is closely related and must be considered simultaneously. Consideration of the problem carried out depends on the form of methods and algorithms applied. The parameters used in this study are as follows.

– Vehicle dispatching

This problem is considered from two points of view. First, an available load is assigned to an idle vehicle. Second, an idle AGV is assigned to a waiting load. Dispatching methods are based on therules by which the idle AGVs are selected and assigned to execute the transportation tasks, e.g. nearest or shortest vehicle travel time rule [4].

– Vehicle routing and scheduling

After dispatching, a certain route should be planned and assigned to the AGV. Once the routing decision is made, the appropri ate schedule must be set. With scheduling, all arrival and departure times of the AGV at each significant point on the route are determined. This is necessary to ensure collision and deadlock free routing [5].

– Dynamic path topology

Optimal route search generally takes a long time. This is because the algorithm is based on global information and calculates all routes for each AGV. The dynamic method of finding routes is based on real-time information about traffic. For example, in the Sun and Wank research algorithm the method is called Dynamic Time-windows for Smart Indoor (DTWS) [6].

Collision-free routes occur in many applications ranging from robot movement to scheduling of several cranes in logistics to container transportation by automated guided vehicles (AGV) [7]. Typically, AGV is not equipped with a smart local collision avoidance system and only relies on central control. This is usually done with a static approach, but in this study we developed a dynamic approach. The first step is to calculate the route in the underlying graph and then use additional methods during route execution to avoid a collision. These rules are usually heuristic [8] which can lead to unexpected arrival times and even deadlocks. To overcome this we developed using classical rules.

Previous research on routing algorithms can be classified into two classes, namely general path topology and special path topology [9]. This study proposes the development of a general path topology with a grid model using the dijkstra algorithm. The general path here means that the path network is a regular graph. Broadbent et al. [10] was the first researcher to provide the concept of AGV routing in a short time free of conflict and propose routing procedures based on the shortest path algorithm. While Hamzeei et al. [11] introduced an algorithm to route AGV vehicles in two-way networks based on annealing algorithm methods, then Nishi and Tanaka [12] discussed dynamic routes to overcome conflict-free dispatching and AGV routes.

Research related to the AGV vehicle conflict-free route which was scrutinized by Rahnama et al [13], Bahari [14], Antackly et.al [8], Malopolski [1], and Kumar et.al [15], when operated only 1 or 2 AGVs that can accept assignments so that when another AGV is operated, the previous AGV must wait for an order or command. Furthermore, when accepting the next assignment, each AGV was only able to avoid front collisions, and deadlocks between intersections as in Jae Pil's research [16] and M. Strzelecki [17] only avoidance of static barriers, but not discuss livelock handling [18], so that the speed of AGV decreases when approaching obstacles as research Arutselvan [19] and Y. Wan et.al [20] resulted in the emergence of a hysteresis effect because accuracy decreases when AGV speed increases.

The problem of AGV movement that appears in the literature review above is that researchers pay less attention to the distance function, the problem of handling uncertainty between AGV, finding the shortest route, the conflict-free route, and optimal scheduling, and parking efficiency. Various kinds of deficiencies which explained in related research, so in this paper, the researcher proposes an optimization solution to handle conflict free routes with a dynamic route approach using a hybrid algorithm dijkstra floyd-warshall which includes scheduling parameters, shortest route search, and conflict free route.

2. RESEARCH METHOD

The research method to overcome obstacle free routes in multi-AGV used is dynamic programming using the floyd-warshall algorithm with a grid topology route model. Dynamic routes use dynamic programming with time windows based on problems in determining vehicle routes, with solutions to solve problems that appear in the form of Conflict-free Vehicle Route Problems with Time Window (CVRPTW). Dynamic routing model is proposed as a solution to overcome the problem of conflict-free routes in finding the shortest path to the destination (workstation) in the AGV pick-up/drop-off distribution process, so that each AGV operation runs smoothly according to the specified time. The multi-AGV autonomous navigation system consists of three parts, namely the input part (with parameters: delivery, conflict free routes, and scheduling), the process part (data processing) and the output part (distribution process: pickup and delivery) on the workstation. The AGV route topology system based on the parameters above is shown in Figure 1.

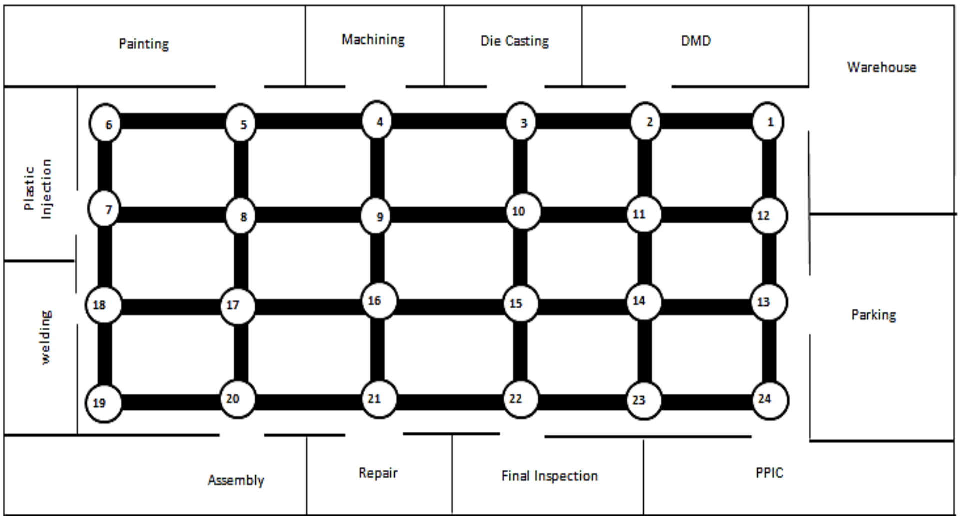

Figure 1. Grid topology route

Problems associated with the design of flow paths such as displacement and buffers, fleet size estimates, and take-up and drop-off stations have been agreed by Vimar and Selva [15], Klein and Kim [21], Liu et al. [5], Umar et al. [22], and Correa et al. [23] Three types of bi-directional network route models are widely used on AGV routes, such as single lane, double lane, and mixed models [24]. We use a two-way single path design using a grid topology model with directional graphs.

The FMS system grid topology route network simulation in Figure 1 uses 24 points/nodes all of which are calculated to find the closest distance to 12 nodes as workstations calculated in the P/D process of distribution of goods in the manufacturing industry warehouse, assuming that node 1 (warehouse) is original node, node 2 (DMD), node 3 (Die casting), node 4 (Machining), node 5 (Painting), node 7 (Plastic Injection), node 18 (Welding), node 20 (Assembly), node 21 (Repair), node 22 (Final Inspection), node 24 (PPIC), and node 15 (Parking). All nodes are calculated using the dijkstra and floyd-warshall algorithm in finding the shortest route with dynamic programming. The route is created using a bidirectional grid topology model to overcome the obstacle free route.

The type of collision-free routes can be classified into four types [25] with three suggested solutions, namely (a): Schedule each AGV; (b): Select the desired route; and (c) AGV goes according to its purpose. Each collision classification is followed by one or two cases. Based on this case, we chose the right strategy for the different collision classifications is shown in Figure 2. Regarding the planning and scheduling of the proposed routes for each AGV, it is expected to improve the efficiency of the Autonomous Vehicle System.

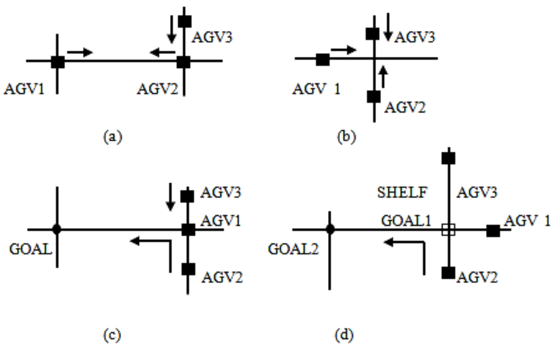

Figure 2. Dynamic route collision type (a) head-on collision, (b) cross-collision, (c) node-occupancy collision, (d) shelf-occupancy collision

The purpose of scheduling is to send a series of AGV tasks in achieving goals in pick-up/drop-off work (or P/D for short) under certain constraints such as deadlines, priorities, route conflicts etc. Objectives are usually related to processing time or resource utilization, such as minimizing the amount of AGV involved while maintaining system throughput, or minimizing the total travel time of all vehicles [26]. The problem given can be stated as follows. There are three AGV vehicles with Q capacity and will distribute good from the origin node to the warehouse which is represented by a network $G = (V, A)$, $V = \{0,1,...,n, n + 1\}$. This representation is the set of nodes or A as the set of arcs, where $A = \{i, j) \mid i, j \in V\}$. For $i \in V$ is *time windows* $[a_i\ b_i]$ and service time S_i, for example $V` = V - \{n=1\}$. For arc $(i, j) \in A$ has travel time t_{ij} and cost (distance) c_{ij}. The origin node is denoted by 0 or n + 1 according to the position indicating the origin node or destination node (end) with $S_0 = S_{n+1} = 0$, $q_0 = q_{n+1} = 0$. E and L indicate the earliest departure time from the origin node and the last arrival time when reaching the designated workstation. The purpose of this problem is to minimize the mileage in serving the process of pick-up and drop-off distribution of AGV goods in the manufacturing industry warehouse using grid topology by meeting the time window constraints.

Before resolve a conflict-free route, we introducethe following equation notation:

G = the directed graph
V = initial node
V' = next node
A = side / bow
Q = AGV load capacity
C_{ij} = the distance of origin node to destination node
D = the distance between two adjacent station
X_{ij} = decision variable
t_{ij} = travel time
N = number of vertices
k = intermediate node
$S_0 = q_0$ = origin node
$[a_i\ b_i]$ = time window
S_i = service time

Problem formulation given:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} Cij\ Xij \tag{1}$$

With

$$\sum_{i \in V} Xij = \infty \quad \forall j \in N, \tag{2}$$

$$\sum_{j \in V} Xij = \infty \quad \forall i \in N, \tag{3}$$

$$\sum_{j \in N} Xoj = 1, \tag{4}$$

$$\sum_{j \in N} Xi(n+1) = 1, = 1 \tag{5}$$

$$\sum_{j \in N} Xij - \sum_{j \in N} Xji = 0 \quad \forall j \in N, \tag{6}$$

$$q_i \le u_i \le Q, \qquad u_i - uj + Qx_{ij} \le Q - q_j \tag{7}$$

$$\forall i,\ j \in V \qquad \text{with } q_1 + q_j \le Q \tag{8}$$

$$S_i + t_{ij} - k(1 - x_{ij}) \le Sj \qquad \forall i,\ j \in N, \tag{9}$$

$$a_i \le D_i \le b_i, \qquad \forall i \in V \tag{10}$$

$$X_{ij} \in \{0,1\} \qquad \forall (i,j) \in A \tag{11}$$

In (2) and (3) are stated each workstation is approved and accepted on demand. Equations (4) and (5) represent the arcs that lead to and leave the workstation. Equation (6) expresses the AGV flow at a node. Equation (7) states that the demand for workstations does not exceed the AGV capacity. Equation (8) states that the route is free of obstacles. Then (9) and (10) guarantee the process of taking and sending goods accordance with the time windows.

Three policies were made regarding movement of AGV positions, namely scheduling, sending, and conflict free routes in the simulation. In the first case, when the vehicle receives delivery assignments to the pick up and drop off point, that is done immediately. The second case, the vehicle is directed to a centralized workstation area if there is no further task request. Then in the third case, all AGVs can carry out the process of loading and unloading of goods from the place of origin to the destination without route conflict. Given a route network G = (V, A) which is a directed graph. *A* set V = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} V'= {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12} with a total of 24 nodes. The set of nodes traversed is $S \subseteq V'$, where node i = 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 are the selected nodes to be visited. Furthermore node 1 is the node that shows the beginning and end of the trip node.

In addition, the distance c_{ij} is the t_{ij} travel time between nodes i and j for pair $i, j \in V$ and the time window [a_i, b_i] for each node i. Distance and travel time meet the imprecision of triangles $c_{ij} \leq c_{ik} + c_{kj}$, $t_{ij} \leq t_{ik} + tkj$ for $i, j, k \in V$. In the grid topology used as a simulation, the distance between vertices at all nodes is 40 centimeters assuming a network route of 24 nodes form a square box on each node.

2.1. Hybrid algorithm formula dijkstra-floyd warshall

2.1.1. The dijkstra algorithm for CVRPTW

Given a directional graph G = (V, E), the non-negative transit time function ce (t) from each side with e = (v, w) ∈ E with [ai, bj] is a window of time where ai ≤ t ≤ bj, t ∈ T is service and leaves time for node v, source node s, destination node d and departure time t0 at the source node of the shortest time-dependent path problem with a time window such that FIFO (first-in first-out) [27] with d can be achieved from the dijkstra algorithm. In table the optimal solution is found if G satisfies three conditions:

– For all verteks $v, w \in V$, and $t_v \leq tw \leq, + \ h(v, t_v) \leq \ t_w + h(w, t_w)$ is a FIFO condition.
– For all edges $e = (v, w) \in E$ and $t \in T$, $h(v, t) \leq \ c_e(t) + h(w, t + c_e(t))$ is a square condition.
– For all verteks $v, w \in V$ and $t_v \leq \ t_w \ [a_v, bv] \leq [a_w, b_w]$ is a time windows condition.

2.1.2. Pseudocode The algorithm dijkstra for the time windows

```
function Dijsktra(Graph, source):
for each vertex v in graph:        //initializations
dist[v]:=infinity:                 //unknown distance function from source to v
previous[v]:=undefined;            //previous node in optimal path
end for                            //from source
dist[source]:=0;                   //distance from source to source
Q :=the set of all nodes in Graph; //all nodes in the graph are unoptimized – thus are in Q
While Q is not empty:              //the main loop
u :=vertex in Q with smallest distance in dist[] ;  //start node in first case
remove u from Q
if dist[u] = infinity:
break;                             //all remaining vertices are
end if                             //inaccessible from source
for each neighbor v of u:          //where v has not yet been
//remove from Q
alt :=dist[u] + dist_between(u,v);
if alt<dist[v]:                    //relax (u,v,a)
dist[v] := alt ;
previous[v] := u;
decrease-key v in Q;               //reorder v in the Queue
end if
end for
end while
return distance;
```

2.1.3. Dynamic programming with floyd-warshall algorithm

The next step is to find the shortest distance with dynamic programming using the floyd-warshall algorithm according to these two statements.

di_{jo} = path length between points i and j
$d_{ijk} = \min (d_{ijk-1}, d_{ikk-1} + d_{kjk-1})$

Then in resolving the AGV conflict-free route problem with time windows in a dynamic program using forward calculations by following the steps developed by Dumas [28] as in the description above can be explained as follows.

Step 1. Initialization
State form $F(\{1, j\}, j, t) = c_{ij}$ *if* $(i, j) \in A$,
$F(\{1, j\}, j, t) = \infty$ *if* $(i, j) \notin A$,
with $a_j \le t \le b_j$, $t = \max\{a_1 + s_1 + t_{ij}, a_j\}$.
Step 2. Displacement of each k-1 state set. For each state (S, i, t) of the $k\text{-}1$ state set, repeat step 3.
Step 3. Build state on state k that can be achieved every state (S, i, t) on the set k-1.
$F(S, i, t) = \min\{F(S = \{j\}, i, t') + c + \mid t \ge t', a_1 \le t' \le b_1\}$, *for* $S \subseteq V', j \in S$, and $a_j \le t \le b_j$.
Obtained *state* $(S', j, t) = (S \cup \{j\}, j, \max\{a_j, t + s_i + t_{ij}\})$ as a feasible extension of the state (S, i, t).
Step 4. Point of view k *with* $k + 1$, if $k \le n$, then proceed to step 2.
Step 5. Calculate the optimal solution. The optimal z distance solution is given by:
$$z = \min_{(i, n) \in A} \ \min_{a_1 \le t \le b_1} \{F\{V'', i, t) + c_{in} \mid t \le b_n - t_{in} - S_1\}.$$

3. RESULT

The following is given Table 1 which contains the distance of 12 nodes that represent the calculated workstation and Table 2 which contains the AGV third time window. The results of the calculation of the algorithm applied to the grid topology path produce the value of the distance between the nodes in Table 1 as the shortest distance on a predetermined workstation. The shortest distance between nodes on a workstation will define the formation of a state graph (S, i, t), with the state formed as a node and the state transition as an arc.

The path selected from its predecessor on the three AGVs when searching for the shortest path is shown in Table 2, so that when the state graph is formed, the start time of the P/D process depends on the time windows for each i is $a_i \le t \le b_i$ with time windows 2 and conditions $t = \max \{a_1 + s_1 + t_{ij}, a_j\}, j \in S$ for stage 1 and $t_j = \max \{aj, t_i, + s_i + t_{ij}\}$ for other stages, with the length of distribution s_i and the travel time t_{ij}.

Table 1. Distance between nodes

Workstation/Node	1	2	3	4	5	6	7	8	9	10	11	12
1	0	40	80	120	160	240	80	280	280	200	1200	80
2	40	0	40	80	120	200	240	240	200	160	160	120
3	80	40	0	40	80	160	200	200	160	120	200	160
4	120	80	40	0	40	120	160	160	120	160	240	200
5	160	120	80	40	0	80	120	120	160	200	280	240
7	240	200	160	120	80	40	0	120	160	200	280	240
13	80	160	120	200	240	240	200	200	160	120	40	0
18	280	240	200	160	120	40	0	80	120	120	240	200
20	280	240	200	160	120	120	80	0	40	80	120	200
21	240	200	160	120	160	160	120	40	0	40	120	160
22	200	160	120	160	200	200	160	800	40	0	80	120
24	120	160	200	240	280	280	240	160	120	80	0	40

Information:

Workstation :	1	2	3	4	5	6	7	8	9	10	11	12
node :	1	2	3	4	5	7	18	20	21	22	24	13

After the shortest path value of the dijkstra and floyd-warshall algorithm calculations is known, a new state (S, i, t) will be forme and the weighting of the distance between the origin node and the destination node already known using the forward tracking system uses computational calculations, as in Table 3. In Table 3, dynamic route calculation with a time window in P/D is given according to the scheduling task for each AGV. Data from Table 4 shows that three parameters (delivery, route and scheduling) are able to overcome conflict-free AGV routes that work autonomously. The experimental results obtained data that the parameters do not affect the number and speed of the vehicle, but affect the arrival interval when demand increases. The more number of vehicles used, the faster the P/D process at the destination workstation. So the work does not spend long time in the queue with a percentage of 1.6% of data processing time and 63.56% of AGV idle positions as the fastest time and included in the medium category.

Table 2. The path selected from the predecessor path

Workstation	Nodes	Rute	Shortest Path
1	1	12-1	40
2	2	12-1-2-11-12-1	40
3	3	1-2-3-10-11-12-1	80
4	4	1-2-3-4--9-10-11-12-1	120
5	5	1-2-3-4-5-8-9-10-11-12-1	160
6	7	1-2-3-4-5-6-7-8-9-10-11-12-1	240
7	18	1-2-3-4-5-6-7-18-17-16-15-14-3-12-1	280
8	20	1-2-3-4-5-8-17-20-21-22-23-24-13-12-1	280
9	21	1-2-3-4-9-16-21-22-23-24-13-12-1	240
10	22	1-2-3-10-15-22-23-24-13-12-1	200
11	24	1-2-11-14-23-24-13-12-1	120
12	13	12-1-12-13	40

Table 3. Forward tracking state

AGV	Label	Workstation	Nodes	Route State (s, i, t)	Time windows		F(s, i, t)
					Load	Unload	
AGV1	Warehouse	1	1	12-1	-	-	80
	DMD	2	2	1-2-11-12-1	41.80	74.74	40
	Die casting	3	3	1-2-3-10-11-12-1	97.47	243.82	80
	Machining	4	4	1-2-3-4-9-10-11-12-1	276.59	422.85	120
	Painting	5	5	1-2-3-4-5-8-9-10-11-12-1	465.66	591.91	160
AGV2	Plastic Injection	6	7	12-11-10-9-8-7-8-9-10-11-12-1	72.99	200.94	200
	Welding	7	18	1-2-11-10-9-8-17-18-17-16-15-14-13-12-1	273.52	369.93	280
	Assembly	8	20	1-2-11-10-9-8-17-20-17-8-9-10-11-12-1	442.50	528.89	280
AGV3	Repair	9	21	13-24-23-22-21-22-10-15-14-13-12-1	57.90	137.86	200
	Final Inspection	10	22	1-2-11-10-15-22-15-14-13-12-1	190.60	296.91	200
	PPIC	11	24	1-2-11-14-23-24-13-12-1	349.56	455.92	200
	Parking	12	13	1-12-13-1	-	-	80

Table 4. The average number of AGVs in the queue

Condition	Low	Medium	High
No. of AGVs	1	2	3
AGV speeds	4	10	30
Demand arrival interval	42.7	32.58	36.56
Idle AGV Positioning	83.62 %	63.56 %	71.42 %
Processing	1.8 %	1.6 %	1.7%

4. CONCLUSION

This paper present a bidirectional path layout with a routing algorithm that allows AGV task to complete pick-up/drop-off (P/D). The path layout consists of two parallel linear lines assuming a length of 40 cm that is connected by vertices as many as 24 points connected to a predetermined P/D work station. A combination of the dijkstra and floyd-warshall algorithms proved to be efficient given for the AGV route on a predetermined path. The results of the development of these two algorithms can be used as an autonomous multi-AGV controller in overcoming conflict-free routes in the manufacturing industry warehouse.

REFERENCES

[1] W. Malopolski, "A Sustainable and Conflict-Free Operation of AGVs in a Square Topology," *Computers and Industrial Engineering,* vol. 126, pp. 472-481, 2018.

[2] I.F.A. Vis, "Survey of Research in The Design and Control of Aautomated Guided Vehicle Systems," *European Journal Operational Research,* vol. 170, no. 3, pp. 677-709, 2006.

[3] W.J. Hsu, *et al.,* "Scheduling and Routing Algorithms for AGVs: a Survey," *International Journal Production Research,* vol. 40, no. 3, pp. 745-760, 2002.

[4] A. Langevin, D. Lauzon and D. Riopel, "Dispatching, Routing, and Scheduling of Two Automated Guided Vehicles in a Flexible Manufacturing System," *Int. J. Flexible Manufacturing System,* vol. 8, pp. 247-262, 1996.

[5] E. Gawrilow, M. Klimm, R. H. Mo, and B. Stenzel, "Conflict-Free Vehicle Routing," *Euro Journal Transpot Logistic,* vol. 1, pp. 87-111, 2012.

[6] X. Sun, *et al.,* "Scheduling Multiple AGVs with Dynamic Time-windows for Smart Indoor Parking Lot," in *2018 IEEE 22nd Int. Conference on Computer Supported Cooperative Work in Design,* pp. 864-868, 2018.

[7] V. Jaiganesh, D. K. Jayashankar, and J. Girijadevi, "Automated Guided Vehicle with Robotic Logistics System," in *Procedia Engineering,* vol. 97, pp. 2011-2021, 2014.

[8] D. Antakly, J-J. Loiseau, and R. Abbou, "A Temporised Conflict-Free Routing Policy for AGVs," *IFAC- Pap.,* vol. 50, no. 1, pp. 1-7, 2018.

[9] Q.I.U. Ling and H.S.U. Wen-jing, "Conflict-Free AGV Routing in a Bi-Directional Path Layout," *In Proceedings of the 5th International Conference on Computer Integrated Manufacturing,* vol. 1, pp. 392-403, 2000.

[10] S. Peyer, D. Rautenbach, and J. Vygen, "A Generalization of Dijkstra's Shortest Path Algorithm with Applications to VLSI Routing," *Journal of Discrete Algorithms,* vol. 7, no. 4, pp. 377-390, 2009.

[11] M. Hamzeei, R. Zanjirani, and H. Rashidi-Bejgan, "An Exact and A Simulated Annealing Algorithm for Simultaneously Determining Flow Path and The Location of P/D Stations in Bidirectional Path," *Journal of Manufacturing System,* vol. 32, no. 4, pp. 648-654, 2013.

[12] T. Nishi and Y. Tanaka, "Petri Net Decomposition Approach for Dispatching and Conflict-Free Routing of Bidirectional Automated Guided Vehicle Systems," in The *4th IEEE International Conference on Automation Science and Engineering,* vol. 42, no. 5, pp. 1230-1243, 2012.

[13] B. Rahnama, S. Member, K. Ebedi, and H. M. Sadeghi, "Self-Corrective Cascade Control Obstacle Avoidance and Deviation Correction System for Robotics Systems," In *IEEE RO-MAN. IEEE,* pp. 133-136, 2013.

[14] A. Bahari, "Automated Guided Vehicles Routing," *Technical Journal of Engineering and Applied Sciences,* vol. 4, no. 2, pp. 60-66, 2014.

[15] N.V. Kumar and C. S. Kumar, "Development of Collision Free Path Planning Algorithm for Warehouse Mobile Robot," in *Procedia Computer Science*, vol. 133, pp. 456-463, 2018.

[16] J.P. Ko, J.W. Jung, and J.W. Jeon, "Anti-Collision Method for AGV Using RFID and ZigBee Network," in *13th Interntional Conference on Control, Automation and Systems,* pp. 599-604, 2013.

[17] M. Polanczyk, M. Strzelecki and K. Slot, "Obstacle Avoidance Procedure and Lee Algorithm Based Path Replanner for Autonomous Mobile Platforms," *Int. J. of Electronics and Telecomm,* vol. 59, no. 1, pp. 85-91, 2013.

[18] R. Tai, *et al.*, "A Time-Efficient Approach to Solve Conflicts and Deadlocks for Scheduling AGVs in Warehousing Applications," in *International Conference on Real-time Computing and Robotics,* pp. 166-171, 2018.

[19] K. Arutselvan, and A. Vijayakumari, "Assistive Autonomous Ground Vehicles in Smart Grid," in *Procedia Technology,* vol. 21, pp. 232-239, 2015.

[20] Y. Wan, *et al.,* "Controller Design for Avoiding Collisions in Automated Guided Vehicle Systems Via Labeled Petri Nets," *IFAC-PapersOnLine Conference Paper Archive,* vol. 51, no. 7, pp. 139-144, 2018.

[21] C.M. Klei and J. Kim, "AGV Dispatching," *International Journal Production Research,* vol. 34, no. 1, pp. 95-110, 2007.

[22] U.A. Umar, M.K.A. Ariffin, N. Ismail, and S.H. Tang, "Conflict-Free Automated Guided Vehicles Routing Using Multi-Objective Genetic Algorithm," *Research Journal of Applied science, Engineering and Technology,* vol. 6, no. 14, pp. 2681-2684, 2013.

[23] A.I. Corréa, A. Langevin, and L. Rousseau, "Scheduling and Routing of Automated Guided Vehicles: A Hybrid Approach," *in Computers & Operations Research,* vol. 34, no. 6, pp. 1688-1707, 2007.

[24] C. Liu, *et al.,* "Path Planning and Intelligent Scheduling of Multi-AGV Systems in Workshop," in *Proceedings of The 36th Chinese Control Conference*, pp. 2735-2739, 2017.

[25] Z. Zhang, Q. Guo, J. Chen, and P. Yuan, "Collision-Free Route Planning for Multiple AGVs in an Automated Warehouse Based on Collision Classification," in *IEEE Access,* vol. 6, pp. 26022-26035, 2018.

[26] D. Giglio, "Task Scheduling for Multiple Forklift AGVs in Distribution Warehouses," *in Emerging Technology and Factory Automation*, pp. 1-6, 2014.

[27] N.A. El-sherbeny, "The Algorithm of The Time-Dependent Shortest Path Problem with Time Windows,"*Applied Mathematics,* vol. 5, no. 17. pp. 2764-2770, 2014.

[28] N.A. El-Sherbeny, "Vehicle Routing with Time Windows: An Overview of Exact, Heuristic and Metaheuristic Methods," *Journal of King Saud University - Science,* vol. 22, no. 3, pp. 123-131, 2010.

BIOGRAPHIES OF AUTHORS

Solichudin, obtained his Bachelor of Education (S.Pd), from the Department of Electrical Engineering Education, Semarang State University (UNNES). Currently pursuing education in the Master of Electrical Engineering Postgraduate Program, Diponegoro University (UNDIP) Semarang.

Aris Triwiyatno, Receive ST., MT., and Dr. degrees in Department of Electrical Engineering, Faculty of Engineering, Sepuluh November Institute of Technology (ITS) Surabaya. Now a permanent lecturer in the Department of Electrical Engineering, Faculty of Engineering, Diponegoro University with research interest: Artificial Intelligent, Software Engineering and Control System.

Munawar A. Riyadi, Received degree ST. and MT. at the School of Electrical and Information Engineering, Bandung Institute of Technology (ITB). PhD degree obtained from Universiti Teknikal Malaysia Melaka (UTeM). His research interests: gate arrays that can be programmed in the field, biomedical measurements, synthesis control systems, cryptography, and microcontrollers. Now a permanent lecturer in the Department of Electrical Engineering, Faculty of Engineering, Diponegoro University.