AUTOMATIC ROUTING OF FORKLIFT ROBOTS IN WAREHOUSE APPLICATIONS

Kelen C. Teixeira Vivaldini, kteixeira@sc.usp.br
Jorge P. M. Galdames, galdames@sc.usp.br
Marcelo Becker, becker@sc.usp.br
Glauco A. P. Caurin, gcaurin@sc.usp.br
Mechatronics Lab. - EESC/USP – Av. do Trabalhador São-Carlense, 400. Zip-code 13566-590 São Carlos/SP – Brazil

Abstract. *Forklift robots are frequently applied in automated logistic systems to optimize transportation tasks and, consequently, reduce costs. Nowadays, in a scenario of extremely fast technological development and constant search for costs minimization, the automation of logistics process is essential to improve productivity and reduce costs. In order to decrease costs of logistics and distribution of goods, it is quite common to find in developed countries mechatronic systems performing several tasks in harbors, warehouses, storages and products distribution centers. Therefore, research into this topic is considered strategic to ensure a greater insertion of the individual countries in to the international trade scenario. In this application, the vehicle routing decision is one of the main issues to be solved. It is important to emphasize that its productivity is highly dependent on the adopted routing scheme. Consequently, it is essential to use efficient routes schemes. This paper proposes an algorithm that produces optimal routes for AGVs (Automated Guided Vehicles) working inside warehouses as forklift robots. The algorithm was conceived to deal with different real situations, such as the need conflict-free paths and presence of obstacles. In the routing algorithm each AGV executes the task starting in an initial position and orientation and moving to a pre-established position and orientation, generating a minimum path. This path is a continuous sequence of positions and orientations of the AGVs. The algorithm is based on Dijkstra's shortest-path method and was implemented in C++. Computer simulation tests were used to validate the algorithm efficiency under different working conditions.*

Keywords*: Routing Algorithm, Obstacle Avoidance, Forklift Robots, Mobile Robots, Mechatronics.*

1. INTRODUCTION

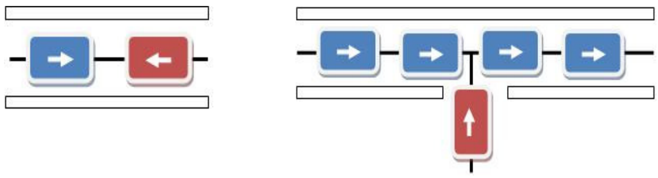
The importance of logistics systems has increased as they are a strategic means of competitiveness for internal and external markets. In large logistic systems, the importance of vehicle routing activities has raised due to their influence on the final costs of the products. Even a small improvement in the routing process may provide significant and positive impacts on costs reduction effort.

In the current scenario of extremely fast technological development, automated logistic systems used by industries, warehouses, seaports, and container terminals are adopting AGVs (Automated Guide Vehicles) as a flexible and scalable alternative to optimize transport tasks. More recently, a new generation of AGVs has provided better reliability, availability robustness, and productivity for the logistics sector, offering significant advantages when compared to conventional material manipulation equipment. The routing task may be understood as the process of simultaneously selecting appropriate paths for the AGVs among different solution possibilities using cost functions in order to increase the system productivity. A routing task consists of an algorithm that selects efficient routes and optimizes them regarding the mission requirements, robot characteristics, and specific environmental conditions. The development of new routing algorithms for AGVs has been a great challenge for robotics researchers to date. Vehicle routing tasks have been studied since 1980. It is possible to find several works in the literature that focus on this subject, e.g. Bodin *et at.* (1983), Psaraftis (1988), and Laporte (1992). More recently, Vis (2006) highlighted the routing importance in current logistics context and emphasized the need of a continuous research in this area.

2. THE ROUTING TASK

It is well-known that the selection of a certain route and a time schedule influences the overall intelligent warehouse systems performance. Therefore, one of the main purposes of AGVs routing systems is to minimize the time spent in cargo transportation. Broadbent *et al.* (1985) published one of the first papers concerning the AGVs routing task. Since then, many authors, like Kim and Tanchoco (1991); Desrosiers *et al.* (1995); Seifert *et al.* (1998); and Möhring *et al.* (2004) have focused their efforts on this issue

According to the authors, the algorithms can be classified into two categories: static and dynamic routing. In the first case, i. e, static routing, all the information, such as position and cargo demand is previously known before the route calculus and does not change. Therefore, the route is calculated without taking into account collision avoidance procedures. This strategy can drastically affect the system's performance due to deadlocks (Fig. 1-a) and traffic jams (Fig. 1-b). In this case, it is necessary to add a new collision avoidance system to deal with unplanned situations,which would alter completely the routing previously calculated, as the obstacle avoidance system acts on the run and can not be predicted during the static routing calculus phase. This is the main disadvantage of applying this kind of routing to systems that depend on the knowledge of the arrival time.



(a) (b)

Figure 1. Example of a typical static routing problem: in (a) deadlocks, both forklifts try to use the same route, but in different directions, and in (b) traffic jam, a line of forklifts is blocking another one preventing it from executing its route. Adapted from Möhring *et al.* (2004).

In order to solve this problem, and taking into account a static routing, Kim and Tanchoco (1991) proposed an algorithm to find a conflict-free shortest-time route for AVGs. They introduced the time window graphic concept. In this graph, each node set represents the free window time and, the arc set, the reachability between the free time windows. Consequently, every route calculated has no collisions during the routing execution and the conclusion time of a solicitation is known right after the route calculus.

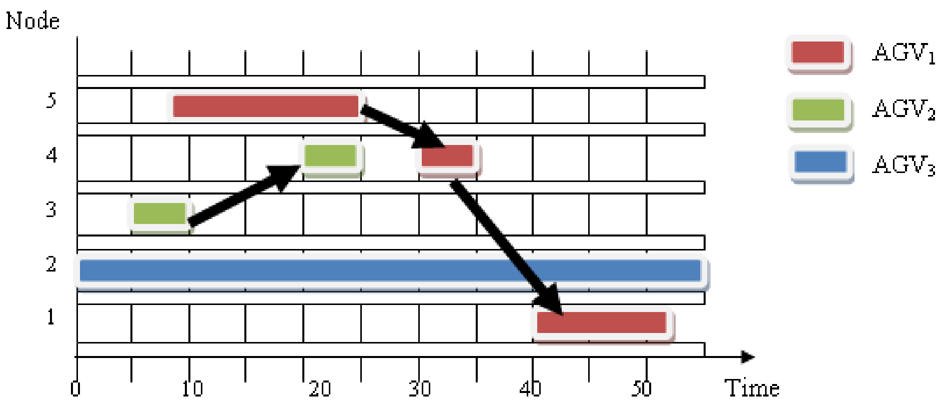


Figure 2: Graphic representation of a forklift entry and exit versus time on each node on the routes. In this example: AGV_1: (N5,[8,25])→ (N4,[30,35])→ (N1,[40,52]); AGV_2: (N3,[5,10])→ (N2,[20,25]) and AGV_3: (N2,[0,55]). The following notation has been adopted: (node,[entry time, exit time]). Adapted from Kim and Tanchoco (1991).

In the case of dynamic routing, all the service orders, or a high percentage of them, occur while the routes are executed. The dynamic routing task consists in finding in real-time the most efficient routes for all forklifts based on several environment data and warehouse priorities information. Therefore, it is not possible to define a complete solution when the operation begins. It is updated as the system receives new information, and it is necessary to react to events that occur in real time, such as new service orders, unpredicted delays, failures, and accidents (Larsen, 2000). In order to avoid the difficulties found in static routing, Möhring *et al.* (2008) computed conflict-free routes. They considered that in a conflict-free approach there is no need for any additional collision avoidance algorithm. This approach considers the physical dimensions of the AGV and is also time-dependent. As a result, it brings great advantage to the management control system (higher control level), which computes the intelligent warehouse solicitations. Apart from searching for conflict-free routes, the routing algorithm must also observe the system interruption occurrence. In some cases, interruptions may occur due to broken vehicles, obstacles on the AGVs way, manual intervention, etc. As a consequence, the AGVs may be blocked and will not be able to complete their tasks. If an interruption occurs, it is necessary to recalculate the overall warehouse system route. The advance of technology used by AGVs has caused a need for new routing systems able to deal with the different situations faced nowadays. In this context, this paper focuses on the development of an algorithm that aims to calculate efficient routes and is able to deal with many variations under environmental conditions. The paper is organized as follows: section 3 shows the basics of storage activities; sections 4 and 5 present the methodology used; section 6 introduces the routing algorithm proposed and the new techniques adopted; finally, section 7 presents some numerical results and section 8 the conclusions.

3. STORAGE LOGISTIC ACTIVITY

The storage activities consume a significant part of the costs involved in the logistics process. It consists of approximately 25% of the sales and 20% of the product costs. However, in order to obtain reasonable results in the logistics project, it is necessary to have an information structure that could satisfy the system requirements, providing fast decisions for the solicitations (Arbache *et al*., 2004). These activities may be classified through the followings basic processes: receiving goods, storage, transfer, orders management, and goods expedition (Figure 3).

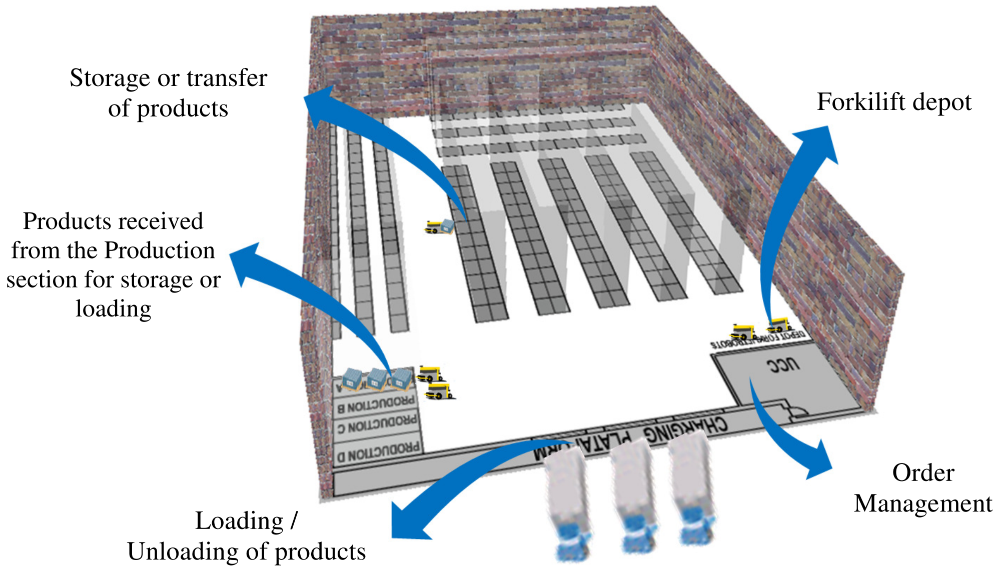


Figure 3. Storage Logistic Activity

Forklift depot is a place where forklifts are kept when not in use.

4. SHORTEST PATH METHOD USING DIKSTRA'S ALGORITHM

One of the simplest problems in graph theory is the determination of the minimum path between two nodes in graphs. In logistics, this problem frequently appears in practical applications directly and as a subtask of more complex tasks.

4.1 Mathematical formulation of the shortest path problem between two nodes

The graph $G = (N,E)$, $N = \{1,2,...,n\}$ (where N is the set of nodes and E the set of edges) is used to map the possible routes inside of the operation environment and as a means for the search for the shortest path from source node 1 to node n of the graph (both nodes represent, respectively, the start and end point of a desired path that will be used to transport goods). The length of each arc in the graph is defined as the cost of connection between two nodes in the environment. Based on this, the equations Eq. (1) to Eq. (4) present the shortest path problem formulation. We considered node 1the origin node and node n the destination node.

$$\text{Minimize } f(x) = \sum_{i=1}^{n} \sum_{j \in S(i)} c_{ij} x_{ij} \tag{1}$$

$$\sum_{j \in S(1)} x_{1j} = 1 \tag{2}$$

$$\sum_{i \in P(n)} x_{in} = \sum_{k \in S(j)} x_{jk}, \quad \text{where } j = 2,...,n-1 \tag{3}$$

$$x_{ij} \geq 0, \qquad \text{where } i = 2,...,n \text{ and } j = 1,...,n \tag{4}$$

where: $S(j)$ is a set of successor nodes j; $P(j)$ is a set of predecessor nodes j; x_{ij} represents the quantities transported from the origin position i to destination j using the arc (i, j); c_{ij} is the arc cost (i, j).

It is important to highlight that both formulation and algorithm can be extended to any two nodes of the graph network. For the solution of the shortest path problem, Dijkstra's Algorithm (Dijkstra, 1959) described in the following section is applied.

4.2. Dijkstra's Algorithm

Dijkstra's algorithm searches for the shortest path between any two nodes of a network, when all arcs have non-negative costs. It uses an iterative procedure that establishes in the first iteration the closest node to the source node. In the second iteration, the subsequent closest node is obtained and so on, until reaching n node.

The algorithm is shown below in a pseudo-code structure. Given a graph G = (N,E), with arc cost $(i, j) \in E$ (hypothesis: $c(i,j) \geq 0$, 1 is the initial node and n is the final node. The set R is composed of previously ordered labeled nodes, i.e., the closest node, the subsequent closest node, etc. The NR set is composed of non-labeled nodes. To recover the minimum shortest path for a determined node k, it will be allocated from the previous node until node k in a path called $p(k)$, i.e., the path from 1 to k is established from path 1 to node $p(k)$ and the arc $(p(k),k)$. If $p(k) = 1$, then the shortest path that attaches node 1 to node k is uniquely established by the arc $(1, k)$. The shortest path from node 1 to node n will be indicated by $d(n)$.

***Step* 1**: Beginning

$R = \{1\}$	: initially, node 1 is labeled
$NR = \{2, ..., n\}$	: the other nodes are not labeled
$d(1) = 0$	: the distance between node 1 and node 1 is zero
$p(1) = 0$	: node 1 is declared as the initial node
For $i \in NR$	
$d(i) = +\infty$	: the distance from node 1 to any non-labeled node i is $+\infty$
$p(i) = n+1$	: node i has a predecessor
$a = 1$	: the last node that has been included in R set

***Step* 2**: For every $i \in NR$, $d(i) = minimum\{d(i), d(a) + c(a, i)\}$ and $p(i) = a$, if $d(i) = d(a) + c(a,i)$.

If $d(i) = +\infty$ for every $i \in NR$, then stop {there is no path from 1 to any nodes in NR}

Else if, establish $k \in NR$ so that $d(k) = minimum\ \{d(i),\ i \in NR\}$. Eliminate node k from NR (i.e., $NR \longleftarrow NR - \{k\}$), include R (i.e., $R \longleftarrow R \cup \{k\}$) in it and do $a=k$.

***Step* 3**: If $a = n$, then recover the C minimum path from the values stored in $p(.)$, starting at $k_1 = p(n)$, and following, $k_2 = p(k_1)$, until node 1 has been reached.

Else if, (i.e., $a \neq n$), return to *Step 2*.

According to Brassard and Bratley (1996), in Dijkstra's algorithm each step needs a number of operations proportional to N, and the steps are iterated $|N - 1|$ times, introducing a complexity of $O(n^2)$. In this paper, Dijkstra's minimum path method is used to calculate the AGVs routes.

5. ROUTING METHOD WITH TIME WINDOWS

This section presents the routing task with time windows restrictions. It consists in finding the shortest distance path, or the lowest path cost, between an origin node and a destination node in a network, based on scheduling restrictions (time windows) for each of the path nodes. The main purpose is the minimization of the total cost of the transportation task. This model is a generalization of the Routing Task.

Considering a set of routes, where each route i is specified by a pair of points (origin and destination), a cost value, a duration and time intervals $[a_i, b_i]$, when the route must begin, the corresponding graphic representation is a node of the network. The routing is defined as the routes sequence performed by a number of forklifts, and a route is the positions sequence and its intermediary points conducted to one of the forklift. Consider P as being the set of routes and I o the set of intermediary points.

A route is represented by arc (i, j) which goes from the end of the route i to its beginning j. For each arc (i, j), we associated a duration time given by t_{ij} and a cost given by c_{ij}, where the arc can be defined only if it is possible to realize route j after route i, respecting the time interval $(a_i + t_{ij} \leq b_i)$.

The problem is described for a warehouse, where each robotic forklift (AGV) leaves its station once. Nodes s and t represent the exit and the entry nodes to the depot. Additionally, the depot is also single, so s and t are coincident. However, they are represented separately in such a way that the network con be easily understood. The network used for the AGV is defined as a set of nodes $N = P \cup \{s, t\}$ and a set of arcs oriented given by $E = I \cup (\{s\} \times P) \cup (P \times \{t\})$.

The variables used in the mathematical formulation are given by

$$x_{ij} = \begin{cases} 1 & \text{if the arc } (i,j) \text{ it is used for the forkilift} \\ 0 & \text{otherwise.} \end{cases} \quad \text{, where } (i,j) \in E$$

t_i = variable that represents the time associated with the beginning of each route i, with $i \in P$.

Following this formulation, optimal routes that respect the constraints of schedules are considered solutions of the problem:

$$\text{Minimize} \quad \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{5}$$

$$\text{Subject to: } \sum_{j\in N} x_{ij} = 1, \qquad i \in P \tag{6}$$

$$\sum_{j\in N} x_{ji} = 1, \qquad i \in P \tag{7}$$

$$x_{ij} \geq 0, \qquad (i, j) \in E \tag{8}$$

$$x_{ij} > 0 \Rightarrow t_i + t_{ij} \leq t_j, \qquad (i, j) \in I \tag{9}$$

$$a_i \leq t_i \leq b_i, \qquad (i, j) \in E \tag{10}$$

$$x_{ij} = \{0,1\}; \qquad (i, j) \in E \tag{11}$$

The relationships in equations (5) to Eq. (8) form a simple routing problem without scheduling constraints. This is a minimum cost flow problem that has an integer solution. The constraint of Eq. (6) informs that starting from a single node *j*, the forklift will arrive in node *i*. The constraint of Eq. (7) establishes that the forklift that arrived at the node *i* will have to leave for a single node *j*. Eq. (9) describes the compatibility requirements between the routes and the schedules, while Eq. (10) establishes the time intervals within which routes must begin. It can be shown that there exits an optimal integer solution to the routing problem with scheduling constraints defined by Eq. (5) to Eq. (10). However, this optimal integer solution can not be obtained directly by linear programming, as Eq. (9) has no linear constraints, but it can be written in a linear form (Desrosiers *et al.*, 1986).

$$t_i + t_{ij} - t_j \leq (1 - x_{ij}) M_{ij}, \qquad (i, j) \in I \tag{12}$$

with $M_{ij} \geq b_i + t_{ij} - a_j$. This formulation is equivalent only if x_{ij} is a binary variable (Eq. 11). In the special case where $[a_{ij}, b_{ij}] = [0, |P| - 1]$ for $i \in P$ and $t_{ij} = 1$ $(i, j) \in I$, we may set $M_{ij} = |P|$ and constraints Eq. (12) become the sub tour elimination constraints proposed by Miller *et al.* (1960) for the travelling salesman problem (Eq. 13):

$$t_i - t_j + |P| x_{ij} \leq |P| - 1, \qquad (i, j) \in I \tag{13}$$

A shortest path between nodes *s* and *t*, considering the schedule constraints (time windows) is obtained by finding the optimal solution of the above model. This formulation forces variable x_{ij} to be an integer. If we relax the constraint integrality of variable x_{ij}, so that $0 < x_{ij} < 1$, we will have the following equation (Eq. 14):

$$0 < t_i + t_{ij} + t_j \leq (1 - x_{ij}) \tag{14}$$

However, Eq. (14) does not satisfy the constraints of the initial problem presented in Eq. (9). Therefore variable x_{ij} must be an integer (Desrosiers *et al.*, 1986). Concluding, one of the main contributions of this method to the routing task with time window is the improvement of the forklift path planning.

6. THE MODEL

In the proposed model, our routing algorithm performs the selection of optimized routes for the forklift robots along the network nodes, being responsible for sending information from the origin to the destination positions. Taking into account the storage activities previously, presented a summary of the tasks is presented in Tab. 1:

Table 1. Information treated in the proposed model.

Information	Entry	Exit
Request (Orders)	Quantities, loading data / unloading data of each product	Necessary number of forklifts and allocation of each request in a route for the forklifts.
Loading	Location point for loading the pallets.	Routes that the forklifts should execute to carry the pallets.
Unloading	Location point for delivering the pallets	Routes that the forklifts should execute to unload the pallets.
Problems in route execution	Route can not be concluded.	Inform: position and problems found.

Concerning storage activities, the routing algorithm considers the objectives below, analyzing sequentially the instructions:

1. Check the numbers of forklifts necessary and the strip of service schedules: calculate the forklift quantities necessary according to the request.
2. Pre-calculate the route (in order to minimize the relation distance and total cost): the distance traveled by forklifts involves the whole course, starting from an initial position and orientation and moving to a pre-established position and orientation, generating a minimum path. This path is a continuous sequence of positions and orientations for the forklift, as well as all loading and unloading points. The objective of this function is to reduce variable costs and select the best route to be executed (Dijkstra's Algorithm).
3. Check the existence of traffic jams and deadlocks in the computed route (route x time): analyze the routes computed verifying the time windows, in case of collisions or deadlock, it is necessary to recalculate the route. The objective is to avoid collisions and deadlocks in the computed route (Rerouting with time Windows).
4. Optimize the route in order to minimize the maneuvers x total cost relation: the maneuvers made in the path of the forklifts consume longer times. The objective of this function is to verify previously possible reductions in the maneuver quantities that do not exceed the cost of the appraised time. The objective is to verify and validate the time of execution of the tasks attributed x position of the forklifts.

6.1. Description of the model

The warehouse model used in the simulation is composed of a fleet of six forklifts that move in a bi-directional circuit composed of 360 nodes interconnected by 652 arcs, as shown in Figure 4. There are some stations: 6 Depot stations for forklift robots, 4 Production stations (A, B, C, and D), 11 Shelves with various stations, and 6 Charging Platform stations. We considered that the Central Control Unit (CCU) calculates the routes using the routing algorithm and sends them to the forklift robots. They navigate through the warehouse connected by a communication system. It is necessary to highlight that the proposed algorithm calculates the route, represented by check-points (nodes in a topological map). In order to apply Dijkstra's algorithm, it will be necessary to previously have the environment map (Fig. 4-a) and an estimation of the forklift robots position. Then, the environmental topological map is used (Fig. 4-b). On this map, the relevant environment features are modeled as check-points (nodes). We assumed that the stations (Shelves, Platform Charging and Production) are placed in nodes belonging to our warehouse topological map, and that each forklift robot in the warehouse can be guaranteed as being in a node. Based on these assumptions, the graph used in the simulations represented in Fig. 4-c, where each node has its address represented by coordinates (x, y), where x and y represent in the map, respectively, the row and column of address in meters. This map is modeled for graph $G(N,E)$. The NxN nodes of the graph represent the intersection of nodes, edges represent two paths between two adjacent intersection nodes, and the length of each edge is a constant value in meters. The attribution of loading and unloading tasks will be omitted in this work. We assumed that the time can be divided into discreet units and that each forklift always arrives at the intersection node at some discreet time. Based on the forklift routing model of the map, we have formally define that: a Storage task is task identified by an ordered pair of *initial* and *final* nodes $((I_x,I_y),(F_x,F_y))$, where (I_x,I_y) represents the origin address, (F_x,F_y) represents the destination address, and $(I_x,I_y) \neq (F_x,F_y)$. Assuming that a task has a distinct origin and also a distinct (but different) destination and each task has only one handled forklift, each forklift begins its activities.

6.2.The Algorithm

The problem of finding routes with the shortest path plays a fundamental role in the area of distribution and logistics management. Most of the routing problems can be resolved through the shortest path method, once the appropriate cost is given for each connection (arcs in the graph).

The proposed algorithm (Fig. 5) was applied in this study based on the programming dynamic approach, which consists in the division of the original problem into smaller and simpler problems. This approach is very useful to the routing problem, as it presents a sequence of decisions to be taken along a time sequence (Arenales *et al.*, 2007). Therefore, the algorithm verifies the task quantity and determines how many forklift robots will be necessary to perform them. It also calculates the minimum route for each robot taking into account its tasks, verifies collisions and traffic jams (if they occur, the route is eliminated and another one is calculated based on time windows). Then, the algorithm tries to optimize (minimize) the robot maneuvers quantity considering the costs and times previously established. Finally, the route is sent to the forklift robots and, if there are more tasks to be executed, it returns to its beginning.

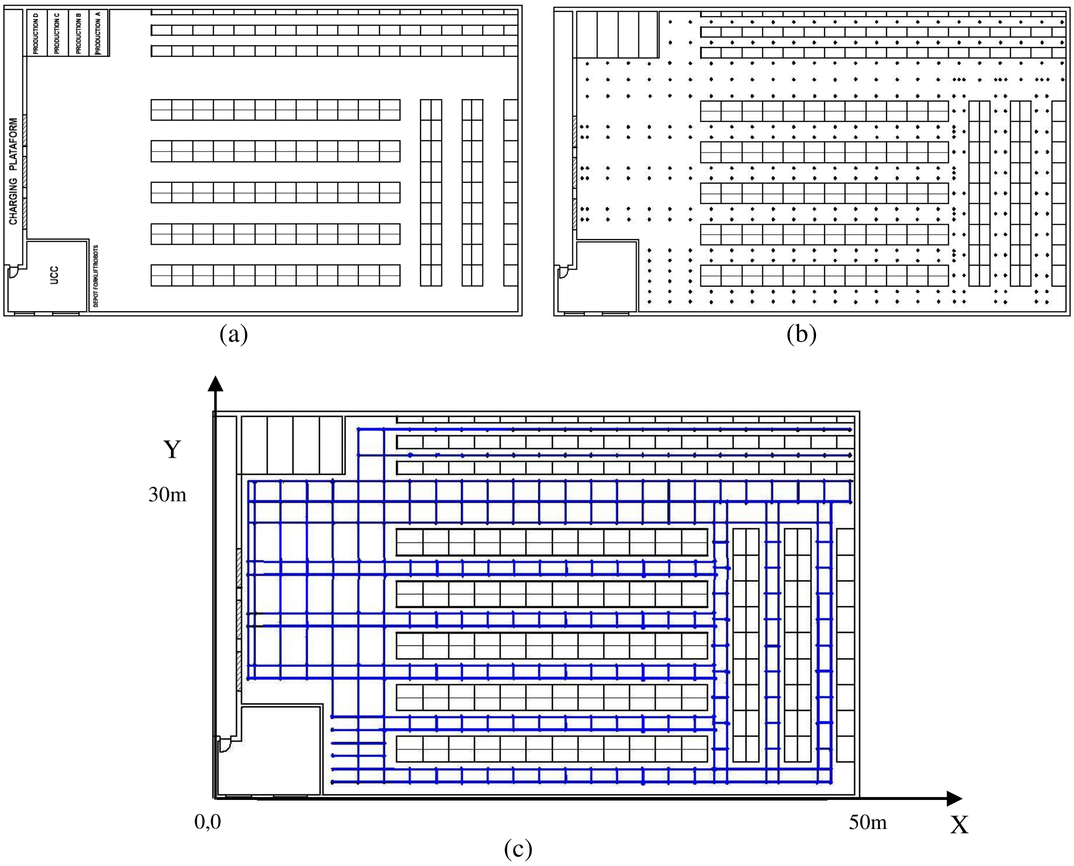


Figure 4. Mapping model considered in the simulations: (a) the warehouse 2D model ;(b) the topologic map with the nodes, and (c) final graph used in the simulations.

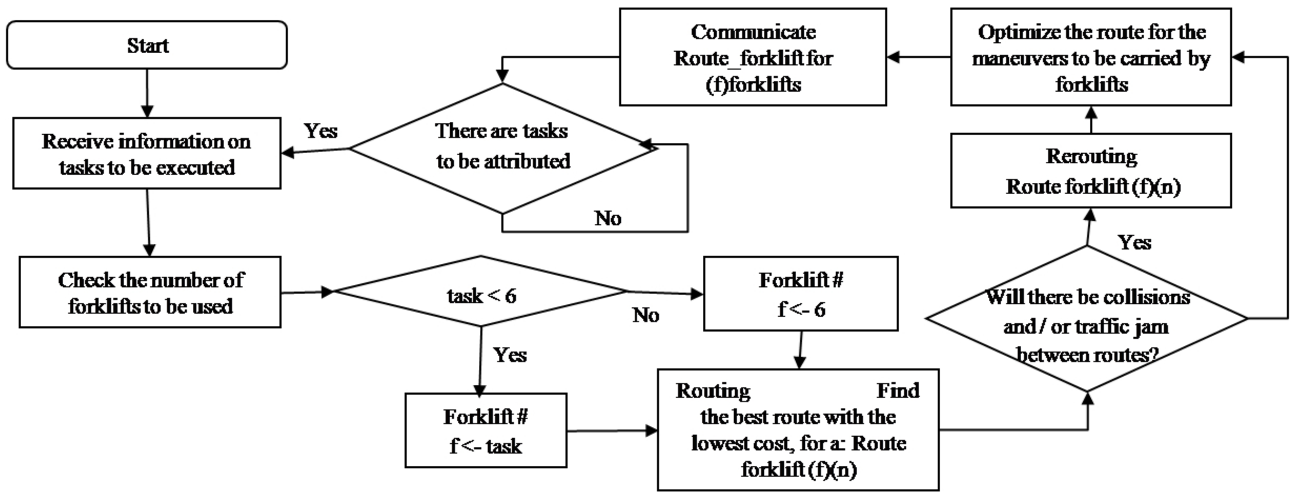


Figure 5. Proposed algorithm

The Dijkstra's shortest path is used in this paper to calculate the routes of the forklifts considering the cost. Basically, the method of routing time window consists in verifying traffic jams and deadlocks in order to improve the route. Next, we verify the route optimization reducing the quantity of curves using again Dijkstra's algorithm with time-window.

7. RESULTS

To assess the efficiency of the algorithm, several simulations were performed inside the virtual warehouse built in the Player/Stage Simulator. Our algorithm was applied to control this warehouse. Simulations were carried out using 6 forklift robots. Each forklift robot had 2 loading and unloading tasks to execute (Table 2, task #1 and #2). Therefore, in the proposed scenario, each route related constitutes a set of 5 sub-routes automatically generated by the algorithm. We tested the three stages of routing, which left the depot and returned to it after finishing the tasks. In order to measure the improvement of the computed routes in the stages, the total time spent by tasks, was compared to obtain the total duration of each task (Tables 3, 4, and 5).

Table 2. Information concerning the forklift robots routes.

Forklift robots	Depot	Task #1		Task #2		Depot
		Origin	Destination	Origin	Destination	
1	Depot # 1	SHELF_A_03	SHELF_A_15	PRODUCTION_A	CHANGING_F	Depot # 1
2	Depot # 2	CHANGING_E	PRODUCTION_C	SHELF_E_14	CHANGING_A	Depot # 2
3	Depot # 3	CHANGING_D	PRODUCTION_B	SHELF_E_01	CHANGING_B	Depot # 3
4	Depot # 4	SHELF_A_17	CHANGING_F	SHELF_C_01	CHANGING_F	Depot # 4
5	Depot # 5	SHELF_A_15	CHANGING_E	SHELF_C_14	CHANGING_F	Depot # 5
6	Depot # 6	SHELF_A_15	SHELF_A_03	SHELF_C_01	CHANGING_F	Depot # 6

Table 3. Information on the time of route execution – Distance X cost

Forklift Robots	Depot	Task #1		Task #2		Depot
		Origin	Destination	Origin	Destination	
1	0	88	208	440	612	760
2	0	148	344	464	604	816
3	0	164	320	432	540	736
4	0	120	316	424	532	656
5	0	80	252	392	540	656
6	0	88	200	328	436	544

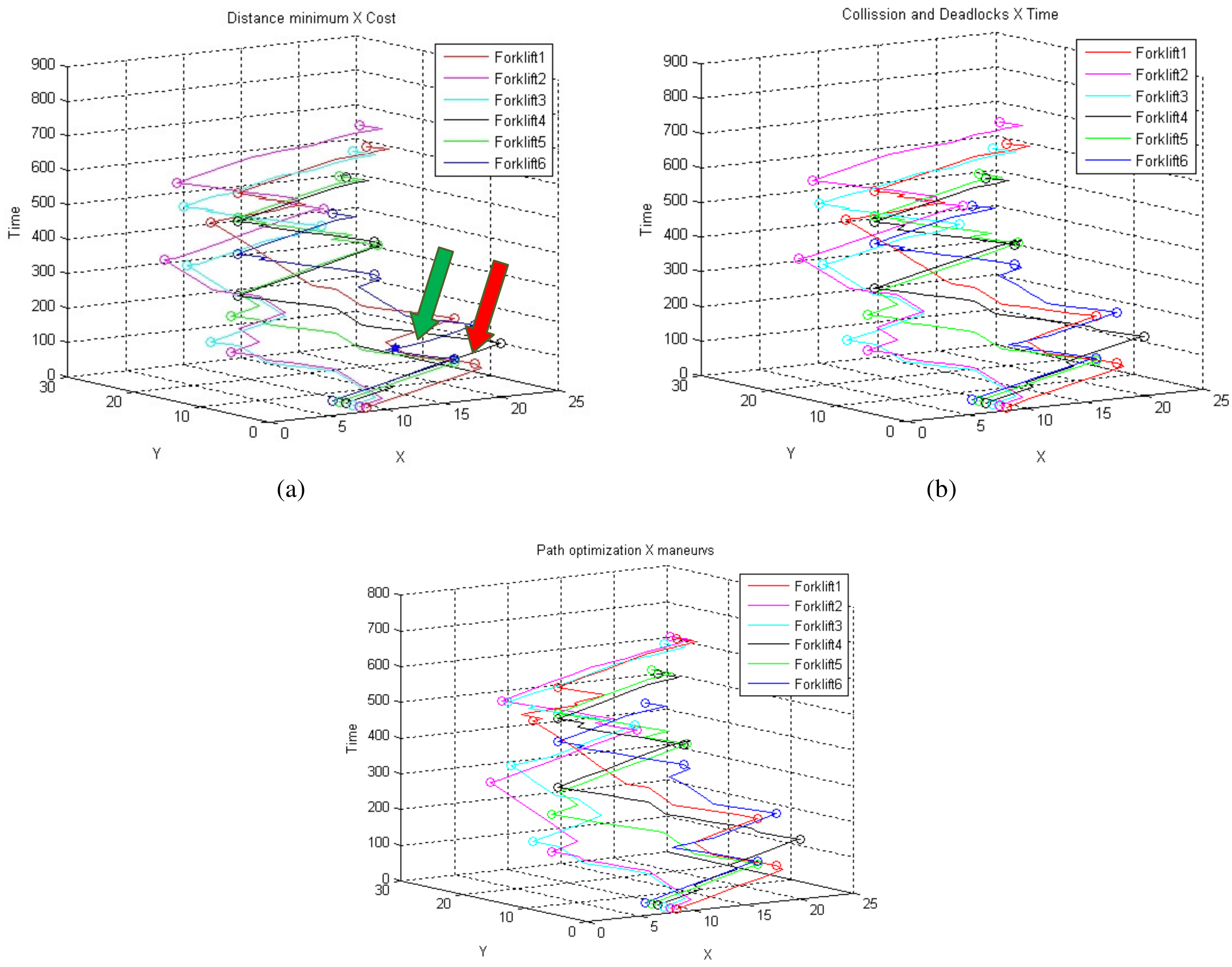
Table 4. Information on the time of route execution – Conflict-free

Forklift robots	Depot	Task #1		Task #2		Depot
		Origin	Destination	Origin	Destination	
1	0	88	208	440	612	760
2	0	148	344	464	604	816
3	0	164	320	432	540	736
4	0	136	332	408	524	648
5	0	80	252	392	540	656
6	0	88	232	352	460	560

Table 5. Information on the time of route execution – Optimization maneuvers

Forklift robots	Depot	Task #1		Task #2		Depot
		Origin	Destination	Origin	Destination	
1	0	88	208	440	612	760
2	0	148	280	400	540	762
3	0	164	320	432	540	736
4	0	136	332	408	524	648
5	0	80	252	392	540	656
6	0	88	232	352	460	560

The results show that in Fig. 6-a Forklift # 6 has two routing problems: a traffic jam with Forklift # 1 (detail represented by a green arrow) and a deadlock with Forklift # 4 (detail represented by a red arrow). In addition to this, in the conflict-free stage (Fig. 6-b), the time cost can be higher than the distance x cost routing (Fig 6-a), as it deals with the traffic jams and dead-locks. Finally, in the stage of maneuvers optimization (Fig. 6-c) the best results for the calculated routes were obtained, demonstrating the method efficiency.



(a) (b) (c)

Figure 6. Forklift positions on *x-y* plane versus time. The nodes of table 2 are represented by circles. Collisions and traffic jams are represented by blue stars. In (a) the minimum route was calculated for the desired tasks; in (b) routes without deadlocks and traffic jams after the rerouting tasks; and in (c) the path was optimized, reducing the total maneuver quantity.

8. CONCLUSIONS

This paper presented a routing algorithm applied to six forklift robots in a simulated warehouse. In order to verify the algorithm's performance, the forklift robots were tested executing two tasks that simulated the load and unload of products. They left their depots and returned after finishing their tasks. Initially, the algorithm calculated the route that each forklift needed to fulfill its tasks, minimizing the traveled distance in relation to the total distance (cost) of the route. It guaranteed that the set of smaller routes will be executed by the forklift robots.

In a second stage, when the routes were already established, the algorithm check the nodes time interval and verified that some routes produced traffic jams and deadlocks. In this case, the problematic sub-route was discharged, the collision point was blocked, and a new sub-route was calculated taking into account time windows. It guaranteed that the routes will be conflict and traffic jam free. Finally, before sending the routes to the forklift robots, the algorithm tried to optimize the paths, reducing the maneuvers quantities (in this case, it also considered previously calculated times and costs).

It is important to emphasize that in the current version, the algorithm assumes that the forklifts use a constant speed during the path, but when getting the smallest runtime route and reduce the maneuvers quantity, and there is an increase in straight line routes. This reduction is particularly interesting, as the hypothesis of constant speed in curves leads to differences between simulated and real processes. Another limitation of the algorithm is that collisions are always solved by finding a new route. A possible solution would be the application of stoppages to one of the forklifts during a certain time interval, or the reduction its speed. Both options will be investigated in the future. In addition, we are planning to introduce other aspects of the storage activities to the algorithm, as high rates of production (high quantities of missions, restricted spaces to move, large number of mobile elements in the environment, and characteristics of the application - e. g. attendance and mounting applications).

Concluding, we may cite as the strongest point of our algorithm the fact that it always find a route and the processing time for computing the route for all cases tested is approximately 1.40 seconds using an AMD Athlon ™ 2600 +, 1.14Ghz with 512MB RAM. (The processing time will be reduced by means of code optimization.)

9. ACKNOWLEDGEMENTS

The authors would like to thank FAPESP (Process 2008/10477-0), CAPES (Process 33002045011P8) and CNPQ (Process 134214/2009-9) for the financial support given to this research.

10. REFERENCES

Arenales, M., *et al* 2007, "Pesquisa Operacional", Elsiever, Rio de Janeiro, Brazil, pp. 524.

Arbache, F.S., Santos, A.G., Montenegro, C. and Salles, W.F., 2004, "Gestão de logística, distribuição e trademarketing", Editora FGV, Rio de Janeiro, Brazil.

Bodin, L.D., et al., 1983, "Routing and scheduling of vehicles and crews: The state of the art", Computers and Operations Research, Vol.10, No. 2, pp. 63–211.

Brassard, G., Bratley, P., (1996), "Fundamentals of Algoritmics", Prentice-Hall, Inc. Pearson Education, Upper Saddle River, New Jersey

Broadbent, A.J. et al., 1985, "Free ranging AGV systems: promises, problems and pathways" Proceedings of 2nd International Conf erence on Automated Materials Handling, UK, pp. 221-237 .

Dijkstra, E.W., 1959 "A note on two problems in connexion with graphs", In: Numerische Mathematik, Vol.1, pp. 269–271.

Desrosiers, F. et al., 1986, "Methods for routing with time windows", Europena Journal Research (North – Holland), Vol. 23, pp.236-245.

Desrosiers, J. et al., 1995, "Time Constrained Routing and Scheduling", Handbooks in Operations Research and Management Science, Elsevier Science, Amsterdam, Vol.8, pp. 35-140.

Kim, Ch.W., Tanchoco, J.M.A., 1991, "Conflict-free shortest-time bidirectional AGV routing", International Journal of Production Research Vol.29, No.12, pp.2377–2391.

Laporte, G., 1992, "The vehicle routing problem: An overview of exact and approximate algorithms", European Journal of Operational Research, Vol. 59, pp. 345–358.

Larsen, A., 2000, "The dynamic Vehicle Routing Problem", IMM, DTU Bookbinder Hans Meyer.

Möhring, R.H. et al, 2004, "Conflict-free Real-time AGV Routing", Operations Research Proceedings Vol. 2004, Springer Berlin.

Möhring, R. H., et al., 2008, "Dynamic Routing of Automated Guided Vehicles in Real-time", Technische Universit¨at Berlin, Book: Mathematics - KeyTechniology for the future, pp. 165-177.

Psaraftis, H.N., 1988 "Dynamic vehicle routing problems", In: Golden, B.L., Assad, A.A. (Editors.), Vehicle Routing: Methods and Studies. Elsevier Sciende Publishers B. V (North-Holland), pp. 223-248.

Seifert, R.W., Kay, M.G., Wilson, J.R., 1998, "Evaluation of AGV routeing strategies using hierarchical simulation. International Journal of Production Research", Vol.36, No.7, July, pp. 1961-1976.

Vis, I.F.A., 2006, "Survey of research in the design and control of automated guided vehicle systems", European Journal of Operational Research., Vol.170 Elsevier Science, Amsterdam, pp. 677-709.

11. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.