Int J Adv Manuf Technol (2016) 82:719–736
DOI 10.1007/s00170-015-7343-4

ORIGINAL ARTICLE

Integrated tasks assignment and routing for the estimation of the optimal number of AGVS

Kelen Vivaldini[1] **· Luís F. Rocha**[2] **· Nádia Junqueira Martarelli**[3] **· Marcelo Becker**[1] **· A. Paulo Moreira**[4]

Received: 17 December 2014 / Accepted: 19 May 2015 / Published online: 23 June 2015

Abstract A fundamental problem in the management of an automated guided vehicle system (AGVS) is the determination of the load to be transported and the vehicle to transport it. The time for the loading and unloading of pallets must be specified as soon as possible. Typical objectives are minimization of travel times and costs by the reduction of the number of vehicles required to fulfill a given transportation order. This article presents a methodology for the estimation the minimum number of AGVs (considering all the available ones at the shop floor level) required to execute a given transportation order within a specific time-window. A comparison is made between the algorithms Shortest Job First and meta-heuristic Tabu Search (applied to an initial solution) for a task assignment. An enhanced Dijkstra algorithm is used for the conflict-free routing task. The number of vehicles is estimated so as to provide an efficient distribution of tasks and reduce the operational costs of the materials handling system. Simulation results of two typical industrial warehouse shop floor scenarios are provided. Although the study focuses on pre-planning of order fulfillment of materials handling, the proposed methodology can also be utilized as an important tool for investment analysis of the warehouse layout design and for estimating the ideal number of AGVs.

Keywords Automated guided vehicles · Task scheduling · Routing · Collision avoidance

✉ Kelen Vivaldini
kteixeira@sc.usp.br

1 EESC-USP, Mechatronics Group - Mobile Robotics Lab, São Carlos, Brazil

2 INESC TEC - INESC Technology and Science (formerly INESC Porto), Porto, Portugal

3 EESC-USP, Mechatronics Group - Mobile Robotics Lab, São Carlos, Brazil

4 INESC TEC - INESC Technology and Science (formerly INESC Porto) and FEUP - Faculty of Engineering, University of Porto, Porto, Portugal

1 Introduction

The automation of logistic systems is essential for the improvements in productivity in warehouses. It is an important factor for competitiveness and increased operational efficiency. Automated logistic systems of distribution areas, such as industries, warehouses, cross docking centers, and container terminals frequently use automated guided vehicles (AGVs) to optimize the production systems in materials handling tasks ([1]).

AGVs are material handling devices used for the transports of pallets (goods and materials) throughout automated areas (i.e., loading and shipping areas, receiving, storage, production stations, and workstations) [2, 3]. The application of AGVs significantly impacts on the execution of tasks because of their advantages and benefits, which include increase of flexibility in processes, low labor costs, 24-h availability (depending on the battery's charging time), and computer integration and control of the materials handling function. Therefore, the number of industries interested in using AGVs as part of a materials handling system has increased [4, 5].

Two very important aspects must be examined for the implementation of an AGV system: quantity of AGVs required for the execution of the tasks and the existence of an efficient task scheduling and routing system that not only minimize the time spent but also avoid collisions and deadlocks [6, 7, 38].

Although the analysis of routing problem started in the early 80s [8–10], approaches related to the minimization of AGV routes in industrial applications gained more attention at the beginning of this century [11–19]. The biggest problem regarding the obtaining of an AGV control of satisfactory performance is the determination of the optimal number of vehicles. Several methodologies have been proposed to achieve this goal ([4, 20–26]) and their main objective is to attend all tasks on time with a sufficient number of vehicles.

A procedure to determine the minimum number of vehicles required can be initiated by the identification of a complete vehicle journey for materials handling tasks. According to [27], the number of vehicles required is the sum of the total travel time (loading/unloading) and waiting times (i.e., amount of congestion) of AGVs in a busy period of time divided by the time during which the AGV is available.

The minimum number of AGVs can be determined by analytic, stochastic, and deterministic models. Deterministic models, such as the network flow model and linear programming models, can be used at the start of a real operation to estimate the number of vehicles required. Stochastic models, as queuing network, aim at incorporating external influences and can be used to determine vehicles requirements. The analytic model determines the number of vehicles considering the total travel time. In a simulated model, real system models are designed and experiments are performed for the understanding of the behavior of AGV systems. Most analytic models reported in the literature have underestimated the number of vehicles required in comparison to the simulation approach [3, 26, 28].

According to [29], the equipment in an automated system can achieve 50 % of the initial investment. For economic reasons, the number of AGVs should not be overestimated [3]. Furthermore, a large number of vehicles may cause more congestion.

Many factors affect the number of vehicles required for handling the throughput in a system. Important vehicle characteristics, such as the guidance type, speed, capacity, and battery life, must be taken into consideration in the dimensioning of the optimal fleet [30].

In general, the routing and determination of number of AGVs are discussed separately. In the AGVs routing problem, no greater focus has been given on how the system defines the number of AGVs necessary to perform tasks. The determination of number of vehicles is commonly tackled only when the AGV System has been designed. Different priority fulfillment, as well as a smaller or larger variety of orders to be attended may also occur. According to such priorities, the orders can be attended for an adjusted number of AGVs. Therefore, such issues can potentially be the focus of future research.

This paper addresses the development of a Task Assignment Module for AGV systems for the determination of the number vehicles required for the execution of a given transportation order. The methodology takes into consideration all available AGVs and dispatches the necessary ones, so that transportation order is performed given a maximum order fulfillment time. These features enable the transportation system to run at reduced operational costs.

The paper is organized as follows: Section 2 presents the methodology developed, Section 3 addresses all the most important issues, Section 4 reports the simulation results, and Section 5 provides the conclusions.

2 Task assignment module

The Task Assignment Module (Fig. 1) estimates the number of AGVs (according to their availability) for the undertaking of tasks, and the time estimated for their fulfillment. After the number of AGVs required has been established, these are assigned to tasks. The routing system calculates the conflict-free routes to minimize the cost and number of maneuvers. After the simulation of routes, the estimated distribution is compared with the simulated results. If the estimation complies with the simulated order fulfillment, the routes are dispatched for the AGVs.

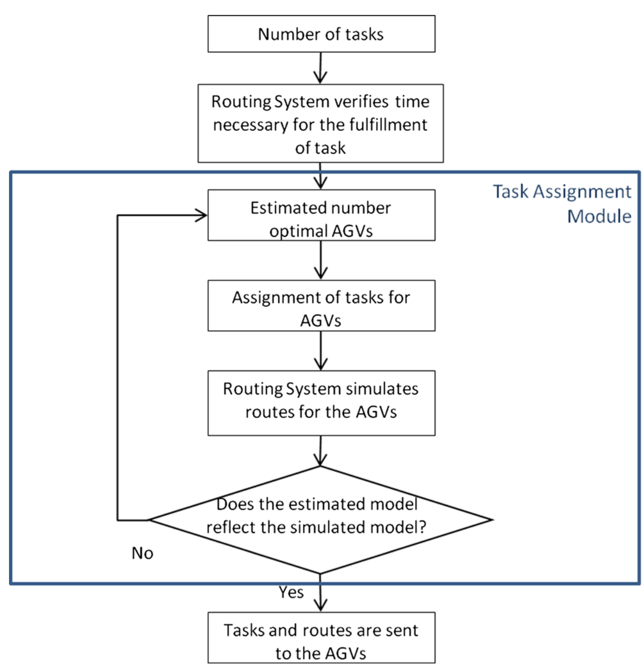
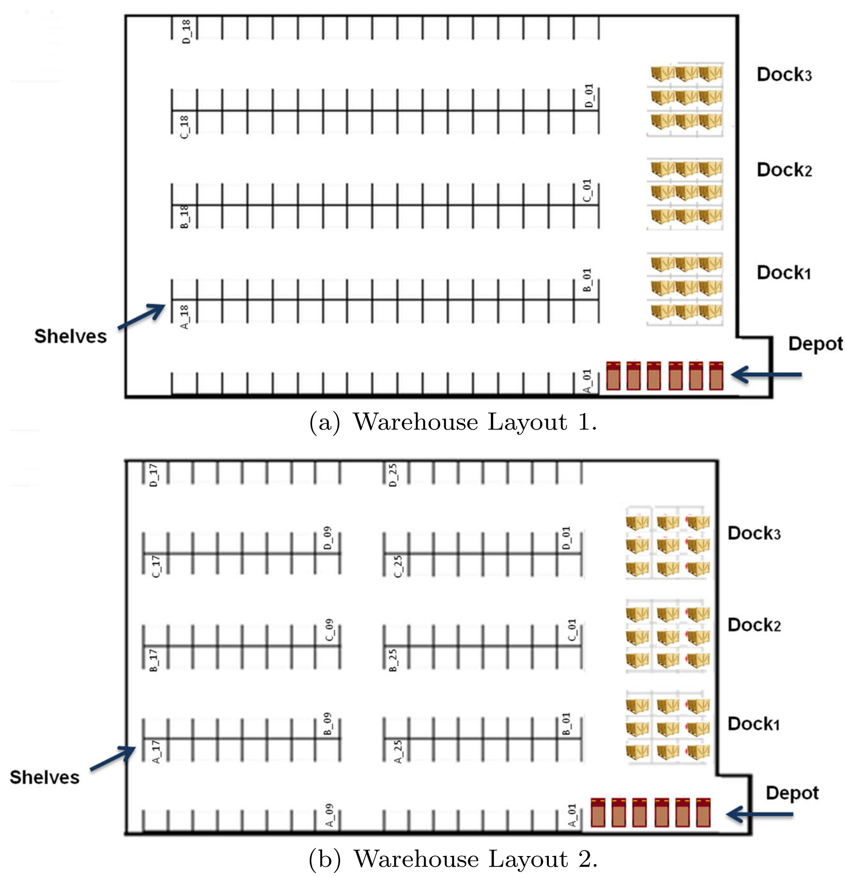


Fig. 1 Task assignment module architecture

The developed architecture can be divided into the following sub-modules:

1. Receiving of the list of transportation orders: transportation orders, which are constituted by a given number of tasks, are provided by the warehouse management system (WMS), and represent the fulfillment of the loading or unloading of a truck. Each task has an origin and a destination point of the pallet.
2. Computation of the time required for the fulfillment of each task by the routing system. A function was implemented for the estimate of the time spent by each AGV in each task, and the time is calculated based on the optimal route. Therefore, the total time of the order fulfillment and the total spent on each task are obtained.
3. Estimate of the number of AGVs required: Taking into consideration the estimated time spent by the routing system on each task, this sub-module determines the estimated optimal number of AGVs. The ratio between the constraint of execution time (release maximum time of the dock defined for example, by the WMS) and the total time spent provides an initial estimate of the number of AGVs. Note that the AGV battery load is not considered (it will be addressed in future work).
4. Assignment of the tasks for AGVs: two heuristics, namely shortest-job-first (SJF) and tabu search were compared. The scheduling algorithm SJF was used with an aging index for the analysis of the estimate obtained in sub-module 3 of our architecture. This analysis indicates the tasks to be attributed to each AGV. SJF [35] minimizes the average execution time of the tasks using the aging index. Two types of aging index were employed: prioritization of the release of dock and imposition of a uniform use of the AGVs available in the warehouse. The tabu search meta-heuristic was considered as an alternative for the SJF and to minimize the order (set of tasks) execution time. After the generation of an initial solution, the tabu search algorithm is applied to explore its neighborhood so as to improve its quality (avoiding local optimal). The result is the set of tasks assigned to each considered AGV.
5. Model simulation: the routing system simulates routes for the AGVs, it calculates conflict-free route at a minimized cost, and provides the total time spent and the total time estimated for the execution of all tasks assigned to each AGV.
6. Does the estimated model reflect the simulated model?: In this sub-module, the result is validated by the comparison between the time spent on the simulation routes (sub-module 5) and the estimated time obtained in the

Fig. 2 The two 2D layouts of the warehouse adopted in simulations



sub-modules 3 and 4. If the result is within the time constraints imposed, the sub-module 7 is performed. Otherwise, the algorithm returns to sub-module 3 to reestimate the number of AGVs. The initial estimated time may increase since collisions cannot be predicted in sub-modules 2 and 3, but only in the sub-module 5.

7. The tasks and routes are sent to the AGVs.

The warehouse models adopted in the methodology and developed methods are described as follows.

2.1 Layouts

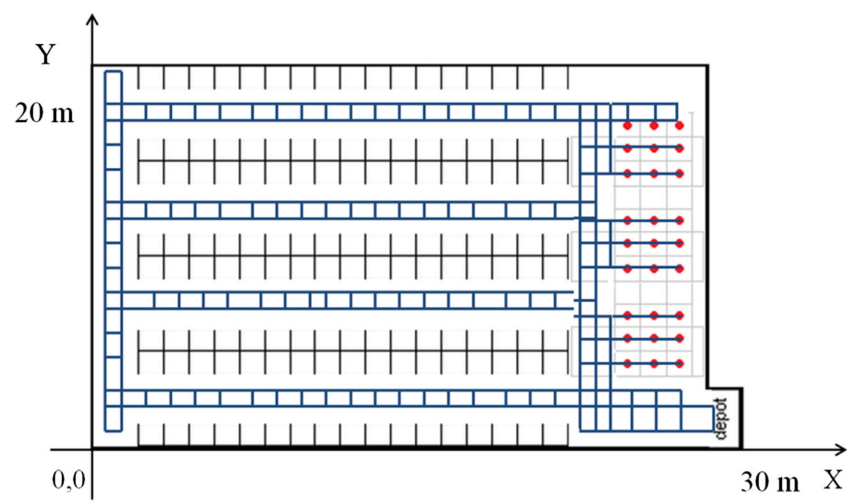
Figure 2a shows the first warehouse layout considered. It has five shelves, three docks, and six depots. Figure 2b shows the second layout, which has ten shelves, three docks, and six depots. These places, except the depots, are stations where AGVs can load and unload pallets.

In both layouts, the AGVs move in a bi-directional path. Layout 1 (Fig. 3a) is composed of 249 nodes interconnected by 351 edge, and layout 2 (Fig. 3b) is composed of 241 nodes interconnected by 337 edge. The routing system calculates the route, represented by nodes of a topological map.

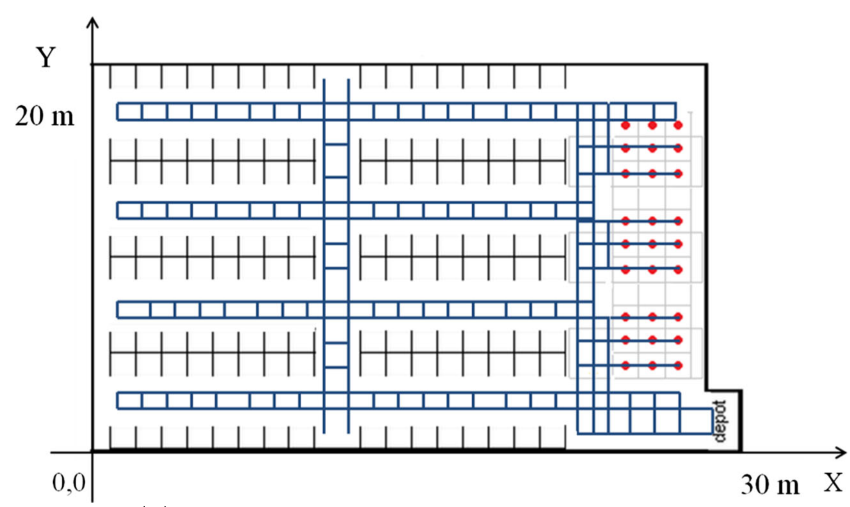
The map is modeled by graph $G(N, E)$, where N is the set of nodes and E is the set of edges (Fig. 4). Matrix NxN represents the set of nodes and each edge represents the connections between them, where the length of each edge is a constant value in meters. The time can be divided into discrete units and each AGV always arrives in the intersection node at some discrete time.

Loading and unloading points (shelves and docks) are called stations. All stations are considered as nodes

Fig. 3 The two 2D layouts of the warehouse adopted in the simulations



(a) Topological Map coonsidered for layout 1.



(b) Topological Map considered for layout 2.

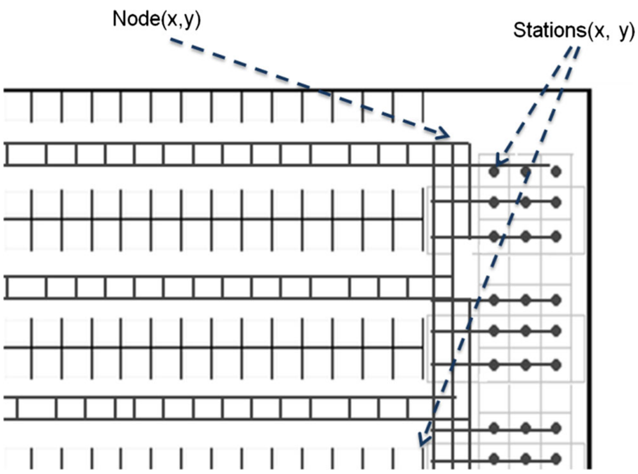


Fig. 4 Graph model example. Stations and nodes have Cartesian coordinates (x, y) as their address, and each line indicates a bidirectional path between adjacent nodes. Applicable to both layouts previously presented in Fig. 3

(Fig. 4) and in the beginning and end of each task the AGV center of mass is considered to be on a node. Based on these definitions and assumptions, each node of the graph used has its address represented by Cartesian coordinates (x, y).

Based on the two warehouse models, we have formally established that each $order_w$ has a set of tasks $qt[w] = [t_1, t_2, .., t_k, t_{k+1}] \forall w, k \in \mathbb{N}$, where each task t_k is identified by an ordered pair of initial and final nodes $((Ix_{t_k}, Iy_{t_k}), (Fx_{t_k}, Fy_{t_k}))$.

(Ix_{t_k}, Iy_{t_k}) represents the origin address, (Fx_{t_k}, Fy_{t_k}) represents the destination address, and $(Ix_{t_k}, Iy_{t_k}) \neq (Fx_{t_k}, Fy_{t_k})$. Each task has an origin and a distinct (and different) destination and can be attributed to only one AGV. Therefore, the routing system uses the environment topological map (Fig. 3) to determine the shortest path between these two points through the methodology developed by [1].

Vivaldini et al. [1] developed a routing system that reduces both time and quantity of unnecessary maneuvers. The algorithm based on Dijkstra's shortest-path method [33] calculates the routes of a robotic forklift adding an heuristic functions to optimize the quantity of maneuvers, using at the same time the method of routing with time-windows [34] to ensure conflict-free routes.

3 Methodology

Please refer to Fig. 1 to follow the methodology description presented in this section.

3.1 Routing system estimates the time necessary to the fulfillment of each task

In the routing system proposed by Vilvadini et al. [1], a function was implemented to estimate the time spent by each task and the total time of order fulfillment. The optimal route executed by the robot in each task (without considering collisions and dead-lock between vehicles) must be calculated for the obtaining of these times.

Firstly, the order type is identified (loading or unloading of trucks). In the loading operation, the pallet is removed from the shelves and dispatched in the correct dock, whereas in the unloading, the pallet is removed from the dock and dispatched to the shelves.

In both procedures, a central point (CP_b) was considered in each dock (D_b), where $b \in \mathbb{N}$ represents the dock number, as shown in Fig. 5. These points were adopted so that a mathematical model could be designed and used in both cases. Each dock has p positions, identified as $D_{(b,p)} \forall p \in \{1, ..9\}$.

This set-up enabled the definition of the distance traveled cost c_u and the travel time $edgeT_u$ of an edge in the graph, where u represents a edge between two nodes in the route.

The variable used in the mathematical formulation is given by

$$s_{(t_k,u)} = \begin{cases} 1, \; if \; edge \; u \; is \; used \; for \; the \; AGV \; in \; task \; (t_k), \\ \quad where \; u \; \in E \\ 0, \; otherwise \end{cases} \tag{1}$$

Equation 2 shows the objective function constructed according to the travel cost sum of the AGV from the origin node to the destination node for each task and the AGV

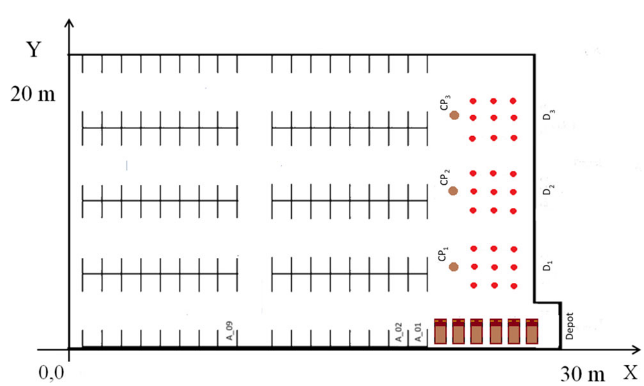


Fig. 5 Shelves, docks, and depots model adopted for the estimation of the time spent on each task

route optimization in relation to the number of maneuvers in the path (qm). Variable β functions as a weight parameter.

$$\min \sum_{u \in E} (c_u \cdot s_{(t_k,u)}) + \beta qm \tag{2}$$

In Eq. 3, the time given by $taskT_{t_k}$ is the task fulfillment time.

$$taskT_{t_k} = \sum_{u \in E} (edgeT_u \cdot s_{(t_k,u)}) \tag{3}$$

Finally, Eq. 4 represents the total time for order fulfillment (T_{order_w}).

$$T_{order_w} = \sum_{k \in qt[w]} taskT_{t_k} \tag{4}$$

After completion of the simulation, the estimated number of vehicles required is calculated.

3.2 Estimation of the number of vehicles required

The number of AGVs required is estimated so that a certain number of tasks can be executed in a previously defined maximum order fulfillment time (T_0). From the time required to perform each task provided by the routing system, it is performed an estimate to determine the initial number of AGVs. For this initial estimation, only one AGV is considered.

$$AGV_{num} = ceil\left(\frac{T_{order_w}}{T_0}\right) \tag{5}$$

Where and just to remember:

$T_{order_w}-$ total time spent by an AGV to perform all tasks (order fulfillment);
$AGV_{num}-$ initial number of AGVs;
T_0- time Limit;

3.3 Assignment of tasks for the AGVs

Two scheduling algorithms shortest-job-first (SJF) applying an index of aging and tabu search were considered for the assignment of the transportation orders.

3.3.1 Shortest job first

The shortest job first has been widely used in computing area (CPU-process) to distribute tasks to the processor. It performs, among processes equally important, the shortest first and minimizes the average execution time of the process ensuring that the final time will be minimal. One can prove mathematically that the SJF always provides the shortest average waiting times [36].

The objective of the SJF for the determination of the number of AGVs is the same: to reduce waiting times and to distribute the tasks for each available AGV.

Algorithm 1 Shortest Job First.

```
procedure VARIABLES i = 1,...,n, n: number of AVGs in depot.
procedure SHORTESTJOBFIRST_VEHICLES
    while TotalNumberPalletSystem ≠ 0 do
        if NumberAGVinDepot ≠ 0 then
            S ← indexShelf(min{TimeFromPalletDockToShelf})
            D ← indexDock(min{TimeFromPalletDockToShelf})
            TotalTimeFromDepotToShelf ← min{TimeFromPalletDockToShelf} + TimeSpentAGVFromDepotToPalletDock(i, S)
            TotalTimeAVGWork[i] ← TotalTimeAVGWork[i] + TotalTimeFromDepotToShelf
            TotalNumberPalletInDock[D] ← TotalNumberPalletInDock[D] − 1
            NumberAGVinDepot ← NumberAGVinDepot − 1
            i ← i + 1
        else
            TotalTimeFromDepotToShelf ← TotalTimeFromDepotToShelf ∗ TimeIndexAGV
            I ← indexAGV(min{TimeFromPalletDockToShelf})
            TotalTimeAVGWork[I] ← TotalTimeAVGWork[I] + min{TotalTimeFromDepotToShelf}
            D ← indexDock(min{TotalTimeFromDepotToShelf})
            TotalNumberPalletInDock[D] ← TotalNumberPalletInDock[D] − 1
            TimeFromPalletDockToShelf ← TimeFromPalletDockToShelf ∗ AgingDock
    return (AGV[I], IDPallet(D, J));
```

The preemptive scheduler approach was used in the SJF algorithm, where the scheduler compares the expected duration of each new task in the system with the remaining processing time for other tasks. A problem associated with the SJF scheduling algorithm is the possibility of starvation of the longest tasks. If the flow of short tasks to the system is high, the long tasks will never be chosen and will be waiting to be attended. This problem can be solved by techniques of aging, since the use of the aging index makes the process more priority in each iteration [37].

In our algorithm, two aging indexes were determined. The first is ratio between the number of pallets on the dock and the maximum number of pallets that the dock may have ($AgingDock$). This index reflects the dock's priority with lower number of pallets, i.e., which is to be released first, compared to the others.

Moreover, the homogeneous fulfillment use of AGVs available in the warehouse must be ensured. For this purpose, we considered a second aging index that is based on the ratio between the total time spent of each AGV and the highest total time of the AGVs ($TimeIndexAGV$). This ratio is multiplied by the time of the tasks and it will influence the AGV chosen (see algorithm 1).

The homogenization of the use of AGVs becomes important so there is no extra work in some equipment, causing them to request maintenance sooner than others, and other consequences, such the efficiency in the order and preservation of the battery. Thus, tasks are distributed so that the average time that each AGV load and unload the pallets is the same.

In contrast, the aging index used to prioritize the release of the dock, works for that each dock to be released more quickly it. In other words, the aging index prioritizes the loading of pallets of the dock with a smaller number of pallets.

3.3.2 Tabu search

The tabu search heuristic is an adaptive local search in continuous operation within a search space. It moves from one solution to another and diversifies the solutions aiming finding a better one [32]. In each tabu search iteration, the best admissible movement is the one of highest evaluation (considering a maximization problem) in the neighborhood of the current solution, in terms of value of the objective function and tabu restrictions. The meta-heuristic tabu search algorithm is an iterative search characterized by the use of dynamic memory and consists of two parts, namely initialization and exploration.

From an initial solution generated randomly or that uses a heuristic, the tabu search will assess a number of different mutations (the vicinity operation) of the current solution at each iteration. The best mutation will be accepted and the changes made are stored in a tabu list which are classified as prohibited in later number of iterations. This strategy avoids a return to already explored solutions. Therefore, at each iteration, the evaluation function validates a certain number of new solutions of which the best based on the objective function is accepted, even if the cost is lower than the cost of the current one. Thus, the algorithm chooses the new solution that produces an improvement or a less decline in the actual value of the cost function (attempt to escape from local minimal).

In the AGVs scheduling problem, the final objective is the allocation of the $order_w$, which have a set of tasks $qt[w]$, by a number of available AGV's to minimize the service time of the order fulfillment. The initial solution (sequence of tasks assigned to each AGV) is generated by the nearest neighbor heuristic (see algorithm 2). In other words, for each AGV (and considering its last performed

Algorithm 2 Generate Initial Solution.

procedure NearestNeighbour
    $s0 \leftarrow empty$
    $ListOfTasksAGV \leftarrow empty$
    $EarlierTask \leftarrow empty$
    $AllAvailableTasks \leftarrow AllTasks$
    $NumberofTasks \leftarrow sizeof(AllTasks)$
    while $k \neq NumberofTasks$ **do**
        for $i = 0 \rightarrow NumAGVs$ **do**
            $NewTask[i] \leftarrow NNeighbour(EarlierTask[i], AllAvailableTasks)$
            $EalierTask[i] \leftarrow NewTask[i]$
            $AllAvailableTasks \leftarrow delete(NewTask[i])$
            $ListofTasksAGV[i]) \leftarrow add(NewTasks[i])$
            $k \leftarrow k + 1$
        end
    end
    $s0 \leftarrow ListOfTasksAGV$
    return $s0$

Algorithm 3 Explore Neighbor solutions.

```
procedure TABU SEARCH
    s ← s0
    BestSolution ← s
    for NumIter = 0 → TabuSerachMaxIter do
        candidateList ← null
        for sCandidate ∈ sNeighborhood do
            if not containsTabuElements(sCandidate, TabuList) then
                candidateList ← candidateList + sCandidate
            end
        end
        sCandidate ← LocateBestCandidate(Candidatelist)
        tabuList ← addFeatureDifferences(sCandidate, sBest)
        s ← sCandidate
        sBest ← s
        if fitness(sBest) < fitness(BestSolution) then
            BestSolution ← sBest
        UpdateTabuList(tabuList)
    return(BestSolution)
```

task, which influences its position on the map), the following task is assigned at a lower cost (shortest travel time/distance). This process is executed cyclically until all the tasks belonging to the $order_w$ have been assigned to one and only one AGV.

Starting from this initial solution, the tabu Search is applied to 500 iterations (this value was experimentally set to the problem at hand) so as to improve the solution obtained. In each tabu search iteration 3, distinct mutations are analyzed. Each mutation corresponds to the replacement of two tasks randomly chosen. Just to highlight that this exchange of tasks can be performed:

- Among tasks belonging to the same AGV (i.e., the order/sequence of execution of tasks is changed)
- Among tasks associated with different AGVs.

The mutation that has improved the current solution or resulted in a lower penalization of the cost function (attempt to escape from local minimal) and for which none of the tasks to be exchanged is in the tabu list is selected. It is then added to the tabu list and cannot be used to generate new mutations during the following 3 Tabu Search iterations.

At the end of the iterations, the best sequence of tasks is delivered for each AGV to minimize the overall time required for the $order_w$ fulfillment. From now on, this algorithm will be addressed as $NN + TS$ (see Algorithm 3).

Just to highlight, and contrarily to the SJF, in the tabu search procedure, any kind of prioritization was taken, both in terms of the release of the dock and the homogeneous use of the AGVs. The idea was to test and compare completely different solutions, and show the modularity of the presented methodology.

After, presented the methods adopted in the assignment of tasks for the available AGVs, the next sub-module would be to use the routing system with collision avoidance (presented in [1]) to simulate the AGVs routes and check if the dock releasing times were met. Results regarding this sub-module will be presented in the following section.

Table 1 Time needed in milliseconds by each AGV to go from each dock position $D_{(b,p)}$ to the respective central point CP_b

$D_{(b,p)}$	CP_b
0	3833
1	1000
2	3833
10	4500
11	1667
12	4500
20	5167
21	2334
22	5167

Table 2 Time needed in milliseconds by each AGV to go from the depot to docks central point CP_b

Depot	CP_1	CP_2	CP_3
1	16,164	12,832	9500
2	15,497	12,165	8833
3	14,830	11,495	8166
4	14,163	10,831	7499
5	13,496	10,164	6832
6	12,663	9331	5999

Table 3 Total time (s), distance traveled (m), and number of tasks for each AGV for the order fulfillment Order 1. Values returned by the routing system (Sub-module 5). Tasks were distributed to each AGV by algorithms SJF and $NN + TS$

	Total time for AGV(s)		Distance traveled (m)		Nº of tasks	
-	SJF	NN+TS	SJF	NN+TS	SJF	NN+TS
out[1]						
AGV 1	1319	1328	1057	1074	27	27
out[2]						
AGV 1	643	664	469	532	13	14
AGV 2	673	657	577	526	14	13
out[3]						
AGV 1	393	433	310	335	8	9
AGV 2	527	445	413	350	10	9
AGV 3	401	456	333	360	9	9
out[4]						
AGV 1	400	349	289	271	8	7
AGV 2	316	321	255	254	7	7
AGV 3	280	363	263	292	6	7
AGV 4	297	304	243	245	6	6
out[5]						
AGV 1	208	303	172	232	4	6
AGV 2	365	279	268	222	7	6
AGV 3	196	233	170	188	4	5
AGV 4	271	282	240	202	6	5
AGV 5	264	260	217	210	6	5
out[6]						
AGV 1	263	226	209	192	5	5
AGV 2	184	253	143	195	4	5
AGV 3	273	254	215	194	5	5
AGV 4	165	188	135	166	4	4
AGV 5	204	188	174	154	4	4
AGV 6	259	185	202	147	5	4

4 Proposed methodologies simulation results

The shop floor warehouses presented in Fig. 2 were used for the validation of the task assignment module. These shop floors have an area of 600 m^2 and three docks (Fig. 3). Four different orders were considered for our trials (two for each layout). Each order correspond to the unloading of three trucks, through the release of 9 euro pallets in each dock (27 unloading tasks). Each AVG has dimensions of 1.2 m x 1 m x 1.5 m, weighs approximately 1.800 kg and travels at an average speed of 1.5 m/s.

Firstly, the time standards were verified for unloading operation (Tables 1 and 2). These values were defined to be equal for both layouts. Furthermore, the time spent by each AGV to load and unload a pallet was established to 10 s.

From the origin and destination point of each pallet, the routing system calculates the estimated time of each task. Please refer to Tables 7 and 8 for Layout 1, and Tables 9 and 10 for Layout 2 available in Appendix 1.

Based on the information of the optimal route provided by the Routing system (Section 3.1), SJF and $NN + TS$ algorithms distribute the tasks to the available AGVs. To validate if the distribution made optimizes the material handling operations, we forced in the estimation the use of the minimal and maximum number of AGVs available ($NumAGV$ equals 1 to $NumAGV$ equals 6).[1]

[1]Note that, in the methodology normal running the initial estimation of the number of AGV would be retrieved by the sub-module presented in Section 3.2. In the end of the simulation, if the simulated model reflect the estimated one the trajectories are sent to the AGV. If not, the number of AGV's is increased.

Table 4 Total time (s), distance travel (m), and number of tasks for each AGV for the order fulfillment of Order 2. Values returned by the routing system (sub-module 5). Tasks were distributed for each AGV using algorithms SJF and $NN + TS$

	Total time for AGV (s)		Distance traveled (m)		Nº of tasks	
-	SJF	NN+TS	SJF	NN+TS	SJF	NN+TS
out[1]						
AGV 1	1217	1280	976	1024	27	27
out[2]						
AGV 1	584	632	452	506	12	14
AGV 2	718	598	560	491	15	13
out[3]						
AGV 1	360	375	301	309	7	9
AGV 2	488	431	358	350	11	9
AGV 3	431	411	348	331	9	9
out[4]						
AGV 1	352	333	272	277	8	7
AGV 2	334	306	260	248	7	7
AGV 3	340	324	266	278	6	7
AGV 4	271	260	240	232	6	6
out[5]						
AGV 1	307	298	237	233	7	6
AGV 2	257	253	219	176	5	6
AGV 3	204	245	157	219	4	5
AGV 4	189	263	166	203	4	5
AGV 5	314	221	247	191	7	5
out[6]						
AGV 1	180	249	146	188	3	5
AGV 2	212	189	162	143	5	5
AGV 3	224	250	187	196	5	5
AGV 4	188	179	158	154	4	4
AGV 5	248	196	187	152	5	4
AGV 6	254	215	216	181	5	4

In the end of the simulation, 6 files were obtained ($out[NumAGV]$) satisfying the distribution of 27 tasks for $NumAGV$. For each $out[NumAGV]$, the a simulated routes were generated by the routing system (sub-module 5 of our architecture) and the travel time, distance traveled (meters), and number of tasks for each AGV were obtained. In this results were already consider conflict-free routes.

4.1 Simulation results—layout 1

4.1.1 Order 1

The analysis of the results for the first order fulfillment (Table 3) show that the best result for the docks released was achieved for six AGVs using the method $NN+TS$ at 254 s. The best solution for the SFJ algorithm was achieved also for six AGVs but at 273 s.

In the normal running of our methodology, and admitting a time-window execution of 310 seconds, the number of AGVs selected would be five AGV considering the $NN + TS$ task assignment meta-heuristic or six AGVs in the case of the SJF.

4.1.2 Order 2

Regarding the order fulfillment of order 2 (Table 4), it is possible to see that the best result to docks released was obtained for the six AGVs at 250 s for the $NN + TS$ algorithm.

Note that it is not straightforward to say that the increase in the number of AGVs will decreased the execution time of the order. The shop floor may become congested, and delay in the transportation time spent by an AGV to execute each task may occur.

Table 5 Total time (s), distance traveled(m), and number of tasks for each AGV for the order fulfillment of Order 1. Values returned by the routing system (sub-module 5). Tasks where distributed for each AGV by algorithms SJF and meta-heuristic $NN + TS$

	Total time for AGV (s)		Distance traveled (m)		Nº of tasks	
-	SJF	NN+TS	SJF	NN+TS	SJF	NN+TS
out[1]						
AGV 1	1264	1287	1042	1072	27	27
out[2]						
AGV 1	691	713	527	562	14	14
AGV 2	584	578	519	496	13	13
out[3]						
AGV 1	410	449	335	362	9	9
AGV 2	521	425	391	336	10	9
AGV 3	368	417	322	345	8	9
out[4]						
AGV 1	275	331	198	261	6	7
AGV 2	271	320	271	261	6	7
AGV 3	384	337	351	278	8	7
AGV 4	323	297	279	250	7	6
out[5]						
AGV 1	199	253	172	253	4	6
AGV 2	274	256	204	256	6	6
AGV 3	230	182	192	182	5	5
AGV 4	248	194	239	194	5	5
AGV 5	315	182	255	182	7	5
out[6]						
AGV 1	220	258	186	201	4	5
AGV 2	193	238	167	178	5	5
AGV 3	216	230	170	212	4	5
AGV 4	162	174	141	158	4	4
AGV 5	226	173	201	154	5	4
AGV 6	261	171	188	140	5	4

4.2 Simulation results—layout 2

4.2.1 Order 1

Concerning the execution time, the analysis of the results for the second layout, and the first order (Table 5), the best result was obtained for the $NN + TS$ algorithm (with five AGVs). With this number of vehicles, the docks are released after 256 s.

Note that, in this case, the results obtained (docks releasing time) for six AGV with the SJF algorithm are worse than those obtained by five vehicles for the $NN + TS$ meta-heuristic (261 vs 256). This shows the great importance of not only have an efficient routing system with conflict-free and deadlock avoidance features, but also a scheduling system that optimizes the dispatched task to each AGV.

4.2.2 Order 2

Looking now for the order fulfillment of the order 2, it is possible to see that the best result was obtained for the number of AGVs equal to six with the docks complete release being made after 241 s.

Until now, we have presented our methodology as a pre-planning algorithm that allows to select the minimum number of AGV required to execute a certain transportation order with time restrictions. However, this methodology can also be utilized in the designing phase of the AGV system. In here, and considering the order information in the WMS and warehouse layout, it would be possible to make an investment analyses and define the number of vehicles to be acquired so the transportation time restriction were met. In this context, the results presented earlier also allow to analyze what is the added value of introducing for example six

Table 6 Total time (s), distance traveled (m), and number of tasks for each AGV for the completion of the dispatched tasks of Order 2. Values returned by the Routing Simulation system (Sub-module 5). Tasks were distributed to each AGV by algorithms SJF and $NN+TS$

	Total time for AGV (s)		Distance traveled (m)		Nº of tasks	
-	SJF	NN+TS	SJF	NN+TS	SJF	NN+TS
out[1]						
AGV 1	1192	1269	979	1012	27	27
out[2]						
AGV 1	684	595	562	469	14	14
AGV 2	567	596	453	506	13	13
out[3]						
AGV 1	347	418	291	328	8	9
AGV 2	532	380	423	322	11	9
AGV 3	360	395	286	330	8	9
out[4]						
AGV 1	297	293	252	206	6	7
AGV 2	285	304	226	256	7	7
AGV 3	307	301	245	250	7	7
AGV 4	332	286	275	264	7	6
out[5]						
AGV 1	253	252	213	193	5	6
AGV 2	253	246	221	206	6	6
AGV 3	298	238	215	177	6	5
AGV 4	193	218	174	194	4	5
AGV 5	245	247	200	206	6	5
out[6]						
AGV 1	181	232	147	189	5	5
AGV 2	189	188	138	142	5	5
AGV 3	223	241	184	203	5	5
AGV 4	273	203	222	175	4	4
AGV 5	258	199	216	147	4	4
AGV 6	177	178	149	138	4	4

AGVs instead of five. In other words, considering the high cost of these vehicles, the results allow to have a cost/benefit trade-off overview between the number of AGVs and the production rate at the shop floor level. Analyzing for exam-

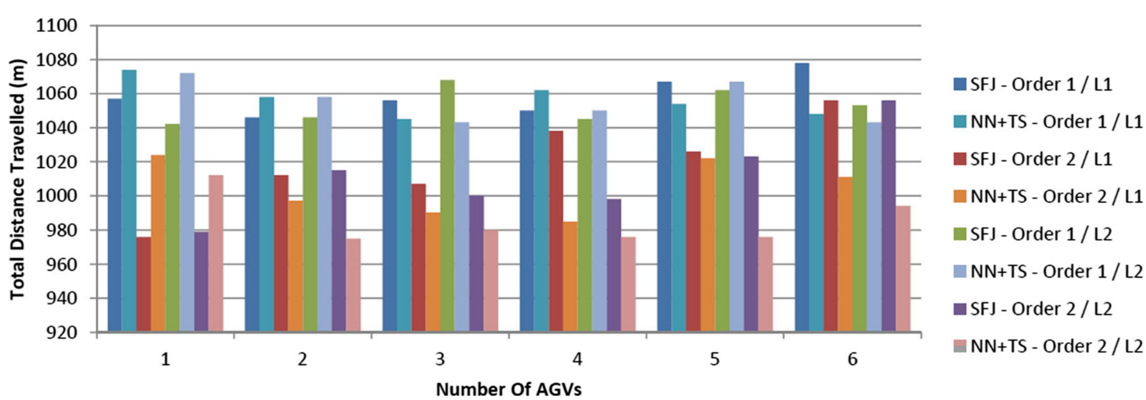


Fig. 6 Total distance traveled by each set of AGVs considering the variation in the number of AGVs and the heuristics used for the task assignment

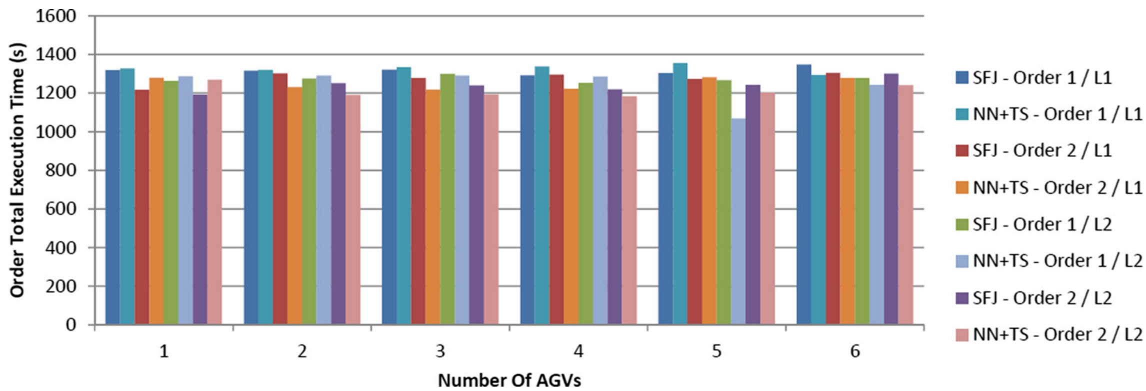


Fig. 7 Total execution time of orders considering the variation in the number of AGVs and the heuristics used for the task assignment

ple Table 6, it is possible to see an improvement of ≃9.5 % in the docks releasing time when comparing the solution based on six AGVs and five AGVs for the meta-heuristic Tabu Search.

4.3 Overall comparison between SJF and $NN + TS$

In both layouts, considering the results presented previously, the total time spent in each order fulfillment and the total traveled distance by AGVs were analyzed separately for the SJF and $NN + TS$ (Figs. 6 and 7).

We can observe that considering the total distance traveled, and in the majority of cases, the SJF for one AGV obtained a better solution compared with $NN + TS$. For the rest of the cases, normally $NN + TS$ present better results.

It is important to minimize the total traveled distance of AGV to reduce consumption of energy (battery). Reducing the energy consumption, the vehicles will stop fewer times for refueling which maximizes the time available for receiving transportation tasks.

One important note, and from all the performed tests, it stays the strong interoperability between the task assignment module (where in our module was used the $NN + TS$ and/or the SJF algorithm) and the conflict free routing algorithm (in our case the enhanced Diksjtra algorithm). In other words, the decision made at higher level has impacted on the decisions made by the routing system, and vice-versa. Therefore, in the design of an AGVs system, these problems should not be taken into consideration separately. Otherwise, the risk of sub(over)-scaling the equipment of transportation system is increased.

5 Conclusions

This paper presented a new module for checking the number of AGVs necessary for the execution of a set of tasks on a warehouse shop floor. The module is employed as a pre-planning, since in real warehouses environment, the transportation data, such as applications, delay of a given load or the need to advance a particular task and relocate activities, are normally known in advance. Furthermore, it enables the estimate of a satisfactory number of AGVs, obtain modifications and rearrangements according to the real needs of the company.

Our module aim at providing much more efficient order fulfillment to the AGVs system, as, in future work, to analyze the estimate of the optimal number of AGVs to different companies layouts and areas of activity. Therefore, the layout of the warehouse and the order information in the WMS will be used in simulations to determine the number of AGVs that the company must acquire to meet its demand, and achieve a more accurate/"real" estimate of the needs to install an automated transportation system in their facilities.

For the continuation of this work, it will also be addressed the idle position of the AGV. Idleness of the AGV is inevitable in the transportation. Instead of causing the vehicle to return to the depot, it is best to park them in maintenance locations or points which are close to the release load. According to Carida [31], the sites for parking of vehicles shall be selected to minimize the response time to requests. MacHaney (1995) presents batteries management policies, which leverage the "idle" time for refueling.

Acknowledgements Project NORTE-07-0124-FEDER-000057 and NORTE-07-0124-FEDER-000060 financed by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF), and by national funds, through the Portuguese funding agency, Fundação para a Ciência e a Tecnologia (FCT). And, CNPq (Grant 142184/2010-1) and FAPESP (Grant # 2008/10477-0) for the financial support provided to this research.

Appendix 1

Table 7 Total time estimated (milliseconds) by the routing system (optimal routes) for the Order 1 Layout 1

Order 1 / Layout 1

Origin D_B	Origin $D_{b,p}$	Destination	TimeSpent	Return CP_1	Return CP_2	Return CP_3
1	0	A_16	29,500	26,333	23,000	20,500
1	1	B_15	23,500	23,500	20,166	17,666
1	2	D_18	13,000	17,666	16,000	18,500
1	10	B_07	18,166	18,166	14,833	12,333
1	11	A_11	25,500	22,333	19,000	16,500
1	12	B_06	17,500	17,500	14,166	11,666
1	20	C_17	21,500	21,500	18,166	12,666
1	21	D_34	15,333	15,333	11,000	85,00
1	22	D_20	5000	9666	8000	10,500
2	0	D_32	4666	3666	6666	9166
2	1	C_22	11,333	15,666	12,333	10,500
2	2	B_12	16,166	21,500	18,166	15,666
2	10	A_21	22,333	23,000	19,666	17,166
2	11	A_30	16,333	17,000	13,666	11,166
2	12	B_20	16,000	20,333	17,000	14,500
2	20	D_19	13,333	17,000	15,333	17,833
2	21	D_14	12,500	15,166	13,500	16,000
2	22	C_05	8166	13,500	10,166	4666
3	0	C_09	7333	16,166	12,833	7333
3	1	C_13	10,000	18,833	15,500	10,000
3	2	B_23	11,500	18,333	15,000	12,500
3	10	B_04	8333	16,166	12,833	10,333
3	11	A_29	14,500	17,666	14,333	11,833
3	12	C_16	12,000	20,833	17,500	12,000
3	20	C_28	6500	11,666	8333	6500
3	21	B_24	10,833	17,666	14,333	11,833
3	22	B_30	6833	13,666	10,333	7833

Table 8 Total time estimated (milliseconds) by the routing system (optimal routes) for the Order 2 Layout 1

Order 2 / Layout 1						
Origin D_B	Origin $D_{b,p}$	Destination	TimeSpent	Return CP_1	Return CP_2	Return CP_3
1	0	B_12	21,500	21,500	18,166	15,666
1	1	D_18	13,666	18,333	16,666	19,166
1	2	D_14	14,166	15,166	13,500	16,000
1	10	C_18	19,333	18,333	15,000	13,166
1	11	A_11	26,166	23,000	19,666	17,166
1	12	B_20	21,333	20,333	17,000	14,500
1	20	A_20	28,333	23,666	20,333	17,833
1	21	D_30	5000	9666	8000	10,500
1	22	C_22	16,666	15,666	12,333	10,500
2	0	B_33	7333	11,666	8333	5833
2	1	A_04	16,166	18,333	15,000	12,500
2	2	C_13	13,500	18,833	15,500	10,000
2	10	B_08	13,500	18,833	15,500	13,000
2	11	C_07	9500	14,833	11,500	6000
2	12	A_14	22,833	25,000	21,666	19,166
2	20	B_03	10,166	15,500	12,166	9666
2	21	D_11	10,500	13,166	11,500	14,000
2	22	B_30	9333	13,666	10,333	7833
3	0	A_29	14,500	17,666	14,333	11,833
3	1	C_27	7166	12,333	9000	7166
3	2	D_13	14,333	14,500	12,833	15,333
3	10	B_31	6166	13,000	9666	7166
3	11	A_24	17,833	21,000	17,666	15,166
3	12	A_20	20,500	23,666	20,333	17,833
3	20	D_29	9166	10333	8666	11,166
3	21	C_30	5166	10,333	7000	5166
3	22	C_11	8666	17,500	14,166	8666

Table 9 Total time estimated (milliseconds) by the routing system (optimal routes) for the Order 1 Layout 2

Order 1 / Layout 2

Origin D_B	Origin $D_{b,p}$	Destination	TimeSpent	Return CP_1	Return CP_2	Return CP_3
1	0	A_14	29,500	26,333	23,000	20,500
1	1	B_13	23,500	23,500	22,166	17,666
1	2	D_18	13,000	13,000	16,000	18,500
1	10	B_07	18,166	18,166	14,833	12,333
1	11	A_09	25,500	22,333	19,000	16,500
1	12	B_06	17,500	17,500	14,166	11,666
1	20	C_15	21,500	21,500	18,166	12,666
1	21	D_32	15,333	15,333	11,000	8500
1	22	D_28	5000	5000	8000	10,500
2	0	D_30	4666	3666	6666	9166
2	1	C_22	11,333	15,666	12,333	10,500
2	2	B_10	16,166	21,500	20,166	15,666
2	10	A_21	19,666	22,000	20,666	16,166
2	11	A_28	16,333	17,000	13,666	11,166
2	12	B_20	16,000	20,333	19,000	14,500
2	20	D_19	13,333	12,333	15,333	17,833
2	21	D_12	12,500	14,166	13,500	16,000
2	22	C_05	8166	13,500	10,166	4666
3	0	C_08	7333	14,166	12,833	7333
3	1	C_11	10,000	18,833	15,500	10,000
3	2	B_23	11,500	18,333	17,000	12,500
3	10	B_04	8333	16,166	12,833	10,333
3	11	A_27	14,500	17,666	14,333	11,833
3	12	C_13	12,000	20,833	17,500	12,000
3	20	C_26	6500	11,666	8333	6500
3	21	B_24	10,833	16,666	15,333	10,833
3	22	B_28	6833	13,666	10,333	7833

Table 10 Total time estimated (milliseconds) by the routing system (optimal routes) for the order 2 layout 2

Order 2 / Layout 2						
Origin D_B	Origin $D_{b,p}$	Destination	TimeSpent	Return CP_1	Return CP_2	Return CP_3
1	0	B_10	215,00	21,500	20,166	15,666
1	1	D_17	13,666	13,666	16,666	19,166
1	2	D_12	14,166	14,166	13,500	16,000
1	10	C_18	19,333	18,333	15,000	13,166
1	11	A_09	26,166	23,000	19,666	17,166
1	12	B_20	21,333	20,333	19,000	14,500
1	20	A_20	25,666	22,666	21,333	16,833
1	21	D_28	5000	5000	8000	10,500
1	22	C_22	16,666	15,666	12,333	10,500
2	0	B_31	7333	11,666	8333	5833
2	1	A_04	16,166	18,333	15,000	12,500
2	2	C_11	13,500	18,833	15,500	10,000
2	10	B_08	13,500	16,833	15,500	11,000
2	11	C_07	9500	14,833	11,500	6000
2	12	A_12	22,833	25,000	21,666	19,166
2	20	B_03	10,166	15,500	12,166	9666
2	21	D_09	10,500	12,166	11,500	14,000
2	22	B_28	9333	13,666	10,333	7833
3	0	A_27	14,500	17,666	14,333	11,833
3	1	C_25	7166	12,333	9000	7166
3	2	D_11	14,333	13,500	12,833	15,333
3	10	A_29	6166	13,000	9666	7166
3	11	A_24	15,166	20,000	18,666	14,166
3	12	A_20	17,833	22,666	21,333	16,833
3	20	D_27	9166	5666	8666	11,166
3	21	C_28	5166	10,333	7000	5166
3	22	C_09	8666	17,500	14,166	8666

References

1. Vivaldini KCT, Galdamens JPM, Pasqual TB, Sobral RM, Araujo RC, Becker M, Caurin GAP (2010) Automatic routing system for intelligent warehouses. In: IEEE International Conference on Robotics and Automation, vol 1, pp 1–6
2. Yifei T, Junruo C, Meihong L, Xianxi L, Yali F (2010) An estimate and simulation approach to determining the automated guided vehicle fleet size in FMS. In: 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT Chengdu, China 2010, vol 9, pp 432–435
3. Vis IFA (2006) Survey of research in the design and control of automated guided vehicle systems. Eur J Oper Res Amsterdam 170(3):677-709
4. Arifin R, Egbelu P (2000) Determination of vehicle requirements in automated guided vehicle systems: a statistical approach. Prod Plan Control 11(3):258–270
5. Fauadi MHFM, Yahaya SH, Murata T (2013) Intelligent combinatorial auctions of decentralized task assignment for AGV with multiple loading capacity. IEEJ Trans Electr Electron Eng 8:371–379
6. Van Der Meer JR (2000) Operational control of internal transport. ERIM Ph.D. Series Research in Management 1, Erasmus University Rotterdam
7. Yoo J, Sim E, Cao C, Park J (2005) An algorithm for deadlock avoidance in an AGV system. Int J Adv Manuf Technol 26(5-6):659–668
8. Bodin LD, Golden BL (1981) Classification vehicle routing and scheduling. Networks, New York 11(2):97–108
9. Kolen AWJ et al. (1987) Vehicle routing with time windows. Oper Res 35(2):266–273
10. Broadbent AJ et al. (1987) Free-ranging AGV and scheduling system. In: Automated guided vehicle systems, pp 301–309
11. Chiba R, Ota J, Arai T (2002) Integrated design with classification of transporter routing for AGV systems. In: IEEE/RSJ International Conference Intelligent Robots and Systems, 2002, Switzerland. Proceedings Lausanne: EPFL, vol 2, pp 1820–1825
12. Duinkerken MB, Ottjes JA (2000) A simulation model for automated container terminals. In: Business Industry Simulation Symposium, 2000, Washington, D.C., 2000. Proceedings [S.l.:s.n.], p 134149
13. Jula H et al. (2000) Container terminals using automated shuttles driven by linear motors. In: IFAC Symposius Control in Transportation System, 9., 2000, Braunschweig. Proceedings Oxford: Pergamon, pp 1–6
14. Kim KH, Bae JW (2004) A look-ahead dispatching method for automated guided vehicles in automated port container terminals. Transfus Sci Baltimore 38(2):224–234
15. Lee C, Ventura JA (2001) Optimal dwell point location of automated guided vehicle to minimize mean response time in a loop layout. Int J Protein Res Londom 39(17):4013–4031
16. Liu CI et al. (2000) Comparing different technologies for containers movement in marine container terminals. In: IEEE International Conference Intelligent Transportation Systems, 2000. Proceedings New York: IEEE, pp 488–493
17. Mhring RH et al. (2004) Conflict-free real-time AGV routing. In: Hein F, Dic H, Peter K (eds) Operations research proceedings 2004. Springer Berlin Heidellberg, Berlin, pp 18–24
18. Sai-Nan L (2008) Optimization problem for AGV in automated warehouse system. In: IEEE International Conference on Service Operations and Logistics, and Informatics, New York. Proceedings New York: IEEE, vol 2, pp 1640–1642
19. Liu CI, Jula H, Ioannou PA (2002) Design, simulation, and evaluation of automated container terminals. IEEE Trans Intell Transp Syst 3(1):12–26
20. Egbelu PJ (1987) The use of non-simulation approaches in estimating vehicle requirements in an automated guided vehicle based transport system. Material Flow 4:17–32
21. Kasilingam RG (1991) Mathematical modeling of the AGVS capacity requirements planning problem. Engrneering Costs and Production Economics 2(1):171–175
22. Mahadevan B, Narendrau TT (1993) Estimation of number of AGVS for an FMS: an analytical model. Int J Protein Res 31(7):1655–1670
23. Rajotia S, Shanker K, Batra JL (1998) Determination of optimal AGV fleet size for an FMS. Int J Protein Res 36(5):1177–1198
24. Vis IFA, De Koster R, Roodbergen KJ, Peeters LWP (2001) Determination of the number of automated guided vehicles required at a semi-automated container terminal. J Oper Res Soc 52(4): 409–417
25. Koo P, Jang J, Huh J (2005) Estimation of part waiting time and fleet sizing in AGV systems. Int J Flex Manuf Syst 16:211–228
26. Ji M, Xia J (2010) Analysis of vehicle requirements in a general automated guided vehicle system based transportation system. Computers & Industrial Engineering 59:544–551
27. Mantel RJ, Landeweerd HRA (1995) Design and operational control of an AGV system. International Journal Production Economics, pp 257–266
28. Vosniakos G-C, Davies BJ (1988) Simulation study of an AGV system in an FMS environment. Int J Adv Manuf Technol 3(4): 33–46
29. Bowersox DJ, Closs DJ, Cooper MB (2007) Supply chain logistics management 2nd ed. McGraw-Hill, New York, pp 108-109,176,241
30. Le-anh T (2005) Intelligent control of vehicle-based internal transport systems, Ph.D Dissertation, ERIM Ph.D series research in management 51, Erasmus University Rotterdam
31. Morandin O, Carida Vinicius F, Kato Edilson RR, Tuma Carlos, C M Adaptive genetic fuzzy, predictive and multiobjective approach for AGVs dispatching. In: IECON 2011 37th Annual Conference of IEEE Industrial Electronics, 2011, Melbourne. IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society, vol 1, pp 2317–2322
32. Glover F (1986) Future paths for integer programming and links to artificial intelligence. Comput Oper Res 13:533–549
33. Dijkstra EW (1959) A note on two problems in connection with graphs. Numerische Mathematik 1:269–271
34. Desrosiers J, Soumis F, Descrochers M, Sauv M (1986) Vehicle routing and scheduling with time windows, Netflow at Pisa. Springer Berlin Heidelberg, pp 249-251
35. Silberschatz A, Galvin PB, Gagne G (2005) Operating systems concepts (7th ed). Wiley, p 161. ISBN 0-471-69466-5
36. Tanenbaum Adrew S (2007) Modern operating systems, 3rd ed. Prentice Hall Press, Upper Saddle River, NJ, USA
37. Ishwari SR, Deepa G (2012) A priority based round robin CPU scheduling algorithm for real time systems. International Journal of Innovations in Engineering and Technology (IJIET) 1(3):1–11
38. Vivaldini KCT, Rocha LF, Becker M, Moreira AP (2015) Comprehensive Review of the Dispatching, Scheduling and Routing of AGVs. In: CONTROLO, 2014, Proceedings of the 11th Portuguese Conference on Automatic Control. Lecture Notes in Electrical Engineering, vol 321. Springer International Publishing, pp 505–514