OmniaScience

Journal of Industrial Engineering and Management
JIEM, 2021 – 14(1): 55-71 – Online ISSN: 2013-0953 – Print ISSN: 2013-8423
https://doi.org/10.3926/jiem.3284

Fleet Management System for Mobile Robots in Healthcare Environments

Eduardo Guzmán Ortiz[1], Beatriz Andres[1], Francisco Fraile[2], Raul Poler[1], Ángel Ortiz Bas[2]

[1]*Research Centre on Production Management and Engineering (CIGIP), Escuela Politécnica Superior de Alcoy, Universitat Politècnica de València, (Spain)*

[2]*Research Centre on Production Management and Engineering (CIGIP), Universitat Politècnica de València (Spain)*

eguzman@cigip.upv.es, bandres@cigip.upv.es, ffraile@cigip.upv.es, rpoler@cigip.upv.es, aortiz@cigip.upv.es

Received: July 2020
Accepted: November 2020

Abstract:

Purpose: The purpose of this paper is to describe the implementation of a Fleet Management System (FMS) that plans and controls the execution of logistics tasks by a set of mobile robots in a real-world hospital environment. The FMS is developed upon an architecture that hosts a routing engine, a task scheduler, an Endorse Broker, a controller and a backend Application Programming Interface (API). The routing engine handles the geo-referenced data and the calculation of routes; the task scheduler implements algorithms to solve the task allocation problem and the trolley loading problem using Integer Linear Programming (ILP) model and a Genetic Algorithm (GA) depending on the problem size. The Endorse Broker provides a messaging system to exchange information with the robotic fleet, while the controller implements the control rules to ensure the execution of the work plan. Finally, the Backend API exposes some FMS to external systems.

Design/methodology/approach: The first part of the paper, focuses on the dynamic path planning problem of a set of mobile robots in indoor spaces such as hospitals, laboratories, and shopping centres. A review of algorithms developed in the literature, to address dynamic path planning, is carried out; and an analysis of the applications of such algorithms in mobile robots that operate in real in-door spaces is performed. The second part of the paper focuses on the description of the FMS, which consists of five integrated tools to support the multi-robot dynamic path planning and the fleet management.

Findings: The literature review, carried out in the context of path planning problem of multiple mobile robots in in-door spaces, has posed great challenges due to the environment characteristics in which robots move. The developed FMS for mobile robots in healthcare environments has resulted on a tool that enables to: (i) interpret of geo-referenced data; (ii) calculate and recalculate dynamic path plans and task execution plans, through the implementation of advanced algorithms that take into account dynamic events; (iii) track the tasks execution; (iv) fleet traffic control; and (v) to communicate with one another external systems.

Practical implications: The proposed FMS has been developed under the scope of ENDORSE project that seeks to develop safe, efficient, and integrated indoor robotic fleets for logistic applications in healthcare and commercial spaces. Moreover, a computational analysis is performed using a virtual hospital floor-plant.

Originality/value: This work proposes a novel FMS, which consists of integrated tools to support the mobile multi-robot dynamic path planning in a real-world hospital environment. These tools include: a routing engine that handles the geo-referenced data and the calculation of routes. A task scheduler that includes a mathematical model to solve the path planning problem, when a low number of robots is

considered. In order to solve large size problems, a genetic algorithm is also implemented to compute the dynamic path planning with less computational effort. An Endorse broker to exchanges information between the robotic fleet and the FMS in a secure way. A backend API that provides interface to manage the master data of the FMS, to calculate an optimal assignment of a set of tasks to a group of robots to be executed on a specific date and time, and to add a new task to be executed in the current shift. Finally, a controller to ensures that the robots execute the tasks that have been assigned by the task scheduler.

Keywords: dynamic path planning, mobile robots, genetic algorithm, in-door environment, healthcare

To cite this article:

Guzmán Ortiz, E., Andres, B., Fraile, F., Poler, R., & Ortiz Bas, A. (2021). Fleet Management System for Mobile Robots in Healthcare Environments. *Journal of Industrial Engineering and Management*, 14(1), 55-71 https://doi.org/10.3926/jiem.3284

1. Introduction

Over the last years, healthcare centres have become concerned to increase the service quality not only to the patients, but also their workers. In this regard, mobility of medical staff within the hospital has been considered a key aspect to improve their productivity. The reduction of time in which medical staff spends with tasks that do not provide value is crucial. These mobility tasks include the transfer of medicines, medical equipment, biological samples, bedding, pharmaceuticals, postal packages or medical waste, amongst others. Bardram and Bossen (2005) estimates that in one day shift medical staff can walk an average of 6.1 km for 7.9 hours, while nurses and doctors on call walk an average of 6.1 km for 18.9 hours. This study was conducted in the haematology department located in a three-story building with 46 patients hospitalized at the same time. In this regard Huang, Cao and Zhu (2019) state that mobility activities in indoor spaces would improve with the deployment of mobile robots, highlighting the advantages that such mobile robots would have in hospital environments.

Facilitating the deployment of mobile robots in health centres is one of the objectives of the ENDORSE project (A Safe, Efficient and Integrated Indoor Robotic Fleet for Logistic Applications in Healthcare and Commercial Spaces) (Vartholomaios, Ramdani, Christophorou, Georgiadis, Guilcher, Blouin et al., 2019). ENDORSE project looks for the design and development of integrated logistic robotic systems to enable efficient, cost-effective and safe indoor logistic services for hospital spaces (Ramdani, Panayides, Karamousadakis, Mellado, Lopez, Christophorou et al., 2019). Currently, there are limited applications of mobile robots in hospitals, this is because they must meet several functional and non-functional requirements that are more difficult to meet than those arising in industrial spaces. The main difference between industrial and public indoor spaces (hospitals, nursing centres, hotels, museums, malls) is the environment since robots in public spaces face a significantly more dynamic environment, surrounded by people and objects, and with architectural limitations, while in industrial and storage spaces are characterized by a structured and defined environment where robots move along predefined paths and the interaction of people with mobile robots is easier to avoid (Ramdani et al., 2019). In indoor spaces the robots must be able to adhere to a space that has not been created specifically for its deployment and to adapt to small spaces that cannot be modified, they must also be less expensive and more efficient than a team of human workers who perform the same task. Therefore, it is necessary to take into account the impact of such requirements on the design of the robot.

The objective of the paper is to describe the implementation of the Fleet Management System (FMS) that plans and controls the execution of logistic tasks by a set of mobile robots in a real-world hospital environment. To reach this aim, we began a review of the algorithms developed in the literature to address dynamic path planning problem of a set of mobile robots in indoor spaces is carried out. Furthermore, we present the design and implementation of a routing engine, a task scheduler, an Endorse Broker, a controller and a backend API at the core of the FMS. The FMS has been implemented and benchmarked using a virtual hospital floor-plant. The implementation also

encompasses the FMS integration with mobile robots, equipped with a diagnostic support module mounted on the robot chassis.

In order to deal with the objective, the paper is organised as follows: section 2 presents a literature review on algorithms used to solve dynamic path planning problems in indoor spaces. Section 3 defines the problem to be solved. Section 4 describes the Fleet Management System (FMS) architecture, which consists of a routing engine tool, a task scheduler, an Endorse broker, a backend API, and a controller. Section 5 depicts the computational experiments carried out to validate two proposals used in the task scheduler, ILP and GA. Finally, conclusions and future research lines are outlined in Section 6.

2. Literature Review

The first task to develop in terms of routing and path planning algorithms was to perform a thorough review of current state of the art of algorithms used to solve dynamic robot path planning problems in the context of indoor spaces. Initially, the applications of mobile robots in healthcare environments are studied, and subsequently the applications in other interior spaces with dynamic obstacles. For a better understanding, Table 1 describes the most relevant concepts treated in this paper as regards mobile robot planning, navigation system, localization, mapping, path planning in a variety of scenarios.

Concept	Definition and characteristics	Author
Mobile Robot	Robots that have the ability to move from one place to another autonomously, without the help of external human operators.	Tzafestas (2018)
Robot Planning	Describes how a mobile robot must operate to get from one place to another place (path planning, motion planning) and how to perform a required task (task planning. Robot planning is a key element of autonomous robot systems.	Khatib, Quinlan & Williams (1997) Tzafestas (2018)
Navigation System	The process of guiding a mobile robot between different positions by using sensors to detect the environmental conditions around it. Navigation system requirements for transport tasks involve mapping; localization; and path planning.	Koubaa, Bennaceur, Chaari, Trigui, Sriti, Ammar et al. (2018) Thurow, Zhang, Liu, Junginger, Stoll & Huang, (2019)
Mapping	The process of learning the environment. A mobile robot requires a map of its environment in order to locate itself in a space and move from one point to another. It means that the mobile robot must follow the map of a known environment to identify addresses and locations.	Da Mota, Rocha, Rodrigues, De Albuquerque and De Alexandria (2018) Koubaa et al. (2018) Thurow et al. (2019)
Localization	Determines the detailed position information of the mobile robot; o in other words, localization determines the robot's position in the environment, allowing to use that information to calibrate movement. Localization system for mobile robots uses encoders, gyroscopes, sensors, cameras, ultrasound sensors, laser rangefinder, and radio-frequency identification (RFID).	Choi, Lee, Lee, and Park, (2011). Koubaa et al. (2018) Thurow et al. (2019)
Path Planning	Consists of finding an optimal collision-free path from a starting point to a target in a given environment. The environment can be static (start and destination positions are fixed, and the obstacles do not change with time) or dynamic (unexpected changes in a planned route due to the arrival of new moving obstacles or changes in the target). Based on the knowledge of the robot environment, path planning may be: (i) local, when the robot is in an uncertain environment and, while in motion, builds an estimated map, this means that the robot generates a new route in response to changes in the environment in real-time; or (ii) global, when it is performed in a static environment, the robot has complete knowledge of the path that it must follow before its movement begin.	Raja and Pugazhenthi (2011) Koubaa et al. (2018) Tzafestas (2018)

Table 1. Definition of key concepts

2.1. Applications of Mobile Robots in Healthcare

The environment in healthcare public spaces is characterized by being dynamic, due to human presence and spontaneous arrivals of new service demands during a service. Table 2 characterises the algorithms proposed to deal with the dynamic path planning problem of mobile robots in healthcare.

The algorithms are classified according to the taxonomy proposed by (Andres, Sanchis & Poler, 2016). It comprises Optimizer algorithms (OA) technique that guarantees to find the optimum solution, in problems with large instances are usually slower, Heuristic algorithm (AH) this technique does not ensure to find optimal solutions, but it can reach near-optimal solutions and Metaheuristic algorithm (AM) these techniques can obtain good or almost optimal solutions, this technique uses random search strategies that require a stopping rule.

The column related with the simulation determines if the tests were performed in a simulated environment, in which a robot obtains information from an algorithm. Commonly, in these simulations are sought to analyse the performance of robots with different types of routes and obstacles randomly generated. The objective of simulation in dynamic environments is to analyse how the robots reach the goals and the goodness of the proposed algorithms for re-planning routes when robots encounter obstacles. Moreover, the software in which experiments are developed is identified. Finally, the algorithms application in a single or multi-robot environment is determined. Multi-Robot Path Planning (MRPP) addresses the problem of how autonomous teams of mobile robots are able to navigate collaboratively and share a common workspace and avoid interference between them (to ensure that two robots do not collide when following their respective paths) (Koubaa et al., 2018).

Author	Algorithm	Real Application	Simulation	Experiments software	Single (S)/ Multi-robot (M)
Fung, Leung, Chow, Liu, Xu, Chan et al. (2003)	OA/ AStar (A*) algorithm	Yes	No	N/S (Not specified)	S
Shieh, Hsieh & Cheng (2004)	OA/ Dynamic programming	No	Yes	N/S	S
Wang, Wei, Zhang & Chen (2009)	OA/ Obstacle-rounding path planning algorithm	No	No	N/S	S
Jeon, Lee & Kim (2017)	OA/ Single-task allocation algorithm. OA/ Multi-task allocation algorithm	Yes	Yes	N/S	M
Huang; Cao et al. (2019)	OA/ Mixed path planning algorithm OA/ A* algorithm AH/ Artificial potential field algorithm	No	Yes	MATLAB	M

Table 2. Algorithms for path planning in healthcare applications

2.2. Applications of Mobile Robots in Other Sectors

There exist several types of mobile robots that have been tested and installed in spaces different from healthcare. Gross, Boehme, Schroeter, Mueller, Koenig, Einhorn et al. (2009) introduce TOOMAS, an assistive shopping mobile robot. Proposing a robust autonomous navigation, and a path planning that uses A* algorithm, and a movement control to avoid collisions. Amanatiadis, Henschel, Birkicht, Andel, Charalampous, Kostavelis et al. (2015) present an autonomous vehicle extraction and transportation (AVERT) based on the "a-robot-for-a-wheel" concept. AVERT mobile robot is able to extract vehicles from confined spaces, swiftly and in any direction. AVERT uses a D* Lite algorithm for computing path planning and for position changes. Yuan, Twardon and Hanheide (2010) present to BIRON II, a mobile robot designed for an interactive scenario called home-tour. When BIRON II enters in real-world apartment, it shows the apartment in an autonomous way to new users. This mobile

robot uses a combination of human-aware path planning, dynamic re-planning abilities, and dynamic obstacle avoidance. It uses A* algorithm to estimate the path. Liu, Stoll, Junginger and Thurow (2012) and Liu, Stoll, Junginger and Thurow (20132) calculate the shortest path for tasks through a hybrid algorithm considering the Floyd algorithm and genetic algorithm, in order to manage transport tasks on a single floor, sending available robots to planned routes and informing about real-time transport. Abdulla, Liu, Stoll and Thurow (2017) present a mobile route planning application to move robots within laboratories distributed on different floors. Designing a hybrid Backbone-Floyd algorithm to calculate dynamic path planning.

2.3. Dynamic Path Planning of Mobile Robots in Indoor Spaces: Study Cases

The interest of researchers in the dynamic path planning of mobile robots is acquiring great relevance, due to new technological developments in mobile robots. Furthermore, path planning for mobile robots is one of the most important aspects of robot navigation (Miao & Tian, 2013).

Author	Algorithm	1	2	3	4
Elshamli, Abdullah & Areibi (2004)	AM/ Genetic algorithm	C	S	S-D	N
Ma & Lei (2010)	AM/ Artificial Bee Colony Algorithm (ABC)	MATLAB	S	S-D	N
Shi & Cui (2010)	AM/ Genetic algorithm	VISUAL C++	S	S-D	N
(Chung & Huang (2011)	AH/ Dynamic AO* (DAO*) AH/ DAO* with D* (DDAO*) algorithm	N/S	S	S-D	N
Yun, Parasuraman & Ganapathy (2011)	AM/ Genetic algorithm	MATLAB	M	S-D	Y
Langerwisch & Wagner (2011)	OA/ Lifelong Planning A* (LPA*) algorithm OA/ Time D* algorithm	MATLAB	S	S-D	N
Valero-Gomez, Valero-Gomez, Castro-Gonzalez & Moreno (2011)	AM/ Genetic algorithm	C++	M	S-D	N
Tuncer & Yildirim (2012)	AM/ Genetic algorithm	N/S	S	D	N
Miao & Tian (2013)	AM/Simulated Annealing (SA)	MATLAB	S	S-D	N
Othman, Samadi & Asl (2013)	OA/ D* algorithm	C++	S	D	N
Serpen & Dou (2015)	AH/ D* lite path search algorithm OA/ A* algorithm AH/ Uniform cost search (UCS) algorithm	Java	M	S-D	N
Li, Xu & Zuo (2015)	AH/ Improved Q-Learning algorithm	N/S	S	S-D	N
Zhang, Zhao, Deng & Guo (2016)	AM/ Improved Genetic algorithm	C#	S	S-D	Y
Das, Behera, Jena & Panigrahi (2016)	AH/ Improved gravitational search algorithm (IGSA)	N/S	M	S-D	Y
Song, Gao, Cao & Huang (2018)	OA/ A* algorithm	MATLAB	S	S-D	Y
Haj Darwish, Joukhadar & Kashkash (2018)	AM/ Bees Algorithm	MATLAB	S	S-D	Y
Faridi, Sharma, Shukla, Tiwari & Dhar (2018)	AM/ Artificial Bee Colony Algorithm (ABC) AH/ Evolutionary programming (EP)	N/S	M	S-D	N
Zhang & Wang (2018)	OA/ A* algorithm	N/S	S	S-D	Y
Huang, Huang, Zhong, Long, Wang, Qiang, et al. (2019)	OA/ D* algorithm	N/S	S	S-D	N

(1) Programming Languages; (2) Single (S)/Multi-robot(M); (3): Static(S) or dynamic (D) obstacles; (4) Real Application Yes (Y), No (N).

Table 3. Study Cases on dynamic path planning of mobile robots in indoor spaces

Dynamic path planning problems for mobile robots in indoor spaces are generally presented as NP-Hard (Nazarahari, Khanmirza & Doostie, 2019) due to computational complexity. Researchers have always looked for efficient techniques or methods to solve these problems. In the literature, we find different types of techniques, including heuristic or metaheuristic algorithms. The algorithms in route planning generally look for the mobile robot to reach its objectives, consuming a minimum time, traveling a minimum distance with the least energy consumption and finding a free route in an environment full of objects (dynamic and static) that includes more robots. Table 3 presents a compilation of these techniques. In this table, it is indicated that the most used algorithms are the Genetic algorithms followed by the A* and D* algorithms.

3. Problem Description

In the problem, there are different types of robots, conforming a robotic fleet that can execute a variety of tasks, either logistic tasks, which imply the transport of goods (medicines, medical equipment, biological samples, bedding, pharmaceuticals, postal packages or medical wastes, amongst others) through the hospital floor plant(s) or other auxiliar sanitary tasks (measure blood-pressure, or perform a COVID-19 test) performed via specialized e-health modules.

To perform the logistic tasks, mobile robots pick trolleys of specific characteristics that are loaded with a set of objects to be transported possibly to different locations. The trolley imposes limits on the dimension and weight of the loads (see Figure 1.). Each logistic task can be divided into a set of low-level instructions or commands that need to be executed by the robot. Examples of low-level commands involved in logistics tasks are GOTO (move to a specific location), PICK (pick a trolley to carry a set of objects), or PLACE (place the trolley in the current location).

From an operational point of view, the Fleet Management System needs to determine the load of each trolley, the optimal routes between different locations and the optimal allocation of tasks to robots. Depending on the task, the load can have different properties, like size, weight, and type (food, sanitary equipment, medicines). There might be additional constraints other than the capacity constraints of trolleys (size and weight of load) related to the type of objects to transport. In general, it is not recommended to transport in the same trolley two different types of objects, but depending on the destination, there might be additional regulatory or sanitary constraints (e.g. it might not be recommended to transport in the same trolley medicines for specific types of patients or food containing specific allergens).

The strategy is to separate the trolley loading problem, the robot routing problem, and the task allocation problem: First calculate optimal paths between locations, then decide the optimal allocation of tasks to robots and finally determine the load of each trolley, each problem subject to its specific constraints. Determine the shortest path is computationally inexpensive and can be calculated for every pair of locations at low computational cost. Once the distances are known, the task allocation problem can be seen as an instance of the Multiple Travelling Salesman Problem (mTSP), where the goal is to minimise an objective function proportional to the total distance, subject to a set of constraints (e.g. payload, battery). Finally, the solution to the mTSP problem determines the optimal load of trolleys which is a trivial assignment problem.

Robots on the other hand have dynamic properties that are relevant in the context of MRPP. Each robot may be at a different position when the MRPP is solved and clearly, it is important to take this aspect into account. Another important parameter to take into consideration is the battery level. Robots have built-in functions to navigate back to the charging station when the battery decreases below a certain level. The MRPP also needs to take this design aspect into account.

Additionally, the number of tasks is also dynamic. Users may plan a set of logistic tasks to be performed at a given periodicity (each shift, every day, once a week, etc), but the robotic fleet must be able to perform tasks on demand. Users can define a new task to be executed now by any robot which is available or if no robots are available, as soon as possible. The navigation path is also dynamic. Aisles might be blocked for robots due to different reasons (cleaning, congestion, or any situation that requires that there are no robots in a given installation). Therefore, the FMS must have the ability to manage the planning of task in a time horizon and also re-scheduling the plan whenever this is required.

Based on these requirements, the objective is to provide the optimal allocation of tasks to robots according to a set of constraints and optimization criteria. In this sense, the primary goal, as mentioned above, is to finalize the set of tasks as quickly as possible, by minimising the total distance. Robots on the other hand need to solve local navigation problems, avoiding collision with other robots or humans, thus dodging static and dynamic obstacles. Therefore, in the design, it is also important to take into account the possible interactions between the global navigation optimisation features of the FMS and the local navigation features of robots. Other important objectives are for instance, achieve a human friendly navigation, avoiding crowded aisles or routing robots through narrow aisles. The following section describes the main design considerations taken to solve the MRPP and implement the FMS. Figure 1 depicts a prototype of the trolley mobile robot used to test the virtual hospital environment.

Figure 1. Trolley mobile robot prototype

4. Fleet Management System

The FMS consists of two integrated tools to support the multi-robot dynamic path planning. The architecture of the FMS is depicted in Figure 2. On the one hand, the routing engine handles the geo-referenced data and the calculation of routes. On the other hand, the task scheduler implements algorithms to solve the task allocation problem and the trolley loading problem using different techniques depending on the problem size. The Endorse Broker provides a messaging system to exchange information with the robotic fleet (model the environment and send commands), while the controller implements the control rules to ensure the execution of the work plan. Finally, the Backend (Application Programming Interface (API) exposes some FMS to external systems. The frontend consumes these services and implements a user interface to manage the robotic fleet. The following sections describe to some level of detail the different building blocks of the FMS architecture.

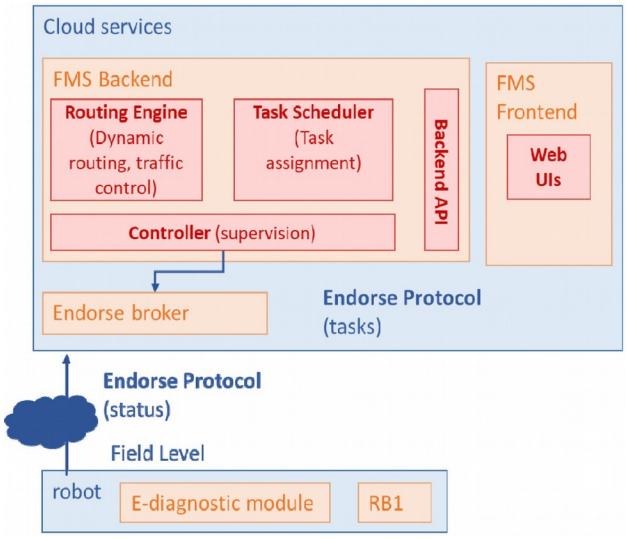

Figure 2. Fleet Management System Architecture

4.1. Routing Engine

4.1.1. Environmental Modelling

The routing engine handles the first step to plan the optimal execution of tasks, though creating a model of the environment that collects all possible routes in the floor-plant. This environment, in which the robots move, is modelled with a graph traversal problem where nodes V in the graph G (V, E) represent locations in the floor-plant, and edges E represent possible routes between locations. This model is consistent with the local navigation model used by the robots. In the model of the routing engine, each node V is characterised with a tuple of values in a 3-dimensional coordinate system (x, y, z), and a type. The z coordinate represents the floor-plant number and the x, y coordinates represent Cartesian coordinates in each floor-plant. The type is used to learn the coordinates of some nodes that need to be treated differently by the routing engine, particularly elevators. This model is consistent with the local navigation model used by the robots as described below.

There are different methods devised to obtain such network model. The first one is used before the robot fleet is deployed and is based on CAD files. Through this method, CAD files containing the blueprint of the floor plant are converted to the GEOJSON format used in the database (MongoDB) of the routing engine. This allows the routing engine to efficiently query geo-reference data using geospatial queries, e.g. to fetch only parts of the graph to re-sequence a subset of tasks when there is a problem at a specific location. In the current version, the user can edit the navigation graph and the location of the destination nodes (rooms, pharmacy, elevators) in separate layers within the same CAD file which is later imported into the routing engine and converted to GEOJSON shapes. The GEOJSON file is later processed to calculate the distances and generate the graph. Euclidean distances determine the weight of the edges between adjacent nodes in the same floor plant. Elevators, on the other hand, join the sub-graphs of the different floor-plants. In the second method, there is a master robot builds the graph based on its navigation history using a specific ROS module (http://wiki.ros.org/voronoi_planner). The master robot shares the generated graph with the FMS and the other robots using the Endorse protocol.

To validate the environmental modelling and benchmark the different candidate algorithms used in the FMS, the authors have used different virtual floor-plants with combed and central courtyard layouts found in the targeted scenarios. Figure 3**.** shows one of the floor-plants used.

Figure 3**.** Floor-plant with combed and central courtyard

As explained in the next section, the graph created by the user will be used to calculate optimal paths to complete the missions and to implement traffic control. This way, robots are routed through the graph and the edges created by the user act as 'virtual lanes' that restricts the traffic of the robot. This navigation strategy is beneficial in crowded environments, or in certain situations, like emergencies, where a predictable behaviour of the robots facilitates the work of human agents.

4.1.2. Cost Function

The routing engine uses the graph G(V, E) to implement the cost function of the optimisation algorithms. First, the current locations of the robots and the destinations of the logistic tasks are added to the graph, adjacent to the closest nodes using the same distance function used to construct the graph in the first place. Edges (eij) have a distance attribute that represents the distance between vertex i and vertex j and a weight attribute that indicates the cost of travelling from vertex i to vertex j. The FMS can use different data sources and apply different functions to compute the cost based on the distance. Through the endorse protocol, robots are able to indicate that a path is blocked, and the FMS can use this information to weight the edges accordingly so that they are not included in the shortest paths. It is possible to calculate the weight based on historic navigation data (e.g. to avoid crowded paths at specific times of the day).

Once the weights are calculated, the Dijkstra algorithm is used to calculate the minimum cost path between nodes in the graph. The routing engine stores the edge sequence (v1, v2, v3, vn) of each path and the cost in a matrix that represents the minimum cost paths between every robot and every destination. The cost values are later used by the task scheduler to find the optimal (minimum cost) allocation of tasks to robots. Basically, the task scheduler uses indexes in this matrix to indicate the optimal allocation of tasks to robots. With this information, the fleet management system is able to control the routing of robots through optimal paths, taking into account distances and other factors that determine the costs of the different paths, as well as the objectives and constraints specified in the task allocation problem.

Based on the shortest paths, the task scheduler implements two different techniques to solve the task assignment problem, a mathematical programming model used when the size of the problem is small and a genetic algorithm to compute the solution for large problems.

4.2. Task Scheduler

4.2.1. Robot Path Planning Using an Integer Linear Programming (ILP) Model

An integer linear programming formulation is used for the multiple traveling salesmen problem (mTSP) problem. In the FMS task scheduler we adapt the ILP developed by (Kara & Bektas, 2006) to deal with the robot path planning problem. In this regard, (Kara & Bektas, 2006) determines a graph G = (V, A), where V is the set of n nodes (rooms), A is the set of arcs, and C = (cij) is the distance matrix that associates each arc (i, j) belonging to A. There are m robots at the base where they receive the load. Kara and Bektas (2006) propose two ILP models: the single depot and multiple depot mTSP. We employ the single mTSP depot model that consists of finding routes for m robots in such a way that all of them start and end in the same depot. Each of the nodes (rooms) are exactly on a route, the number of nodes visited by a robot is predetermined within an interval, and the total cost of visiting all nodes is minimized. The restrictions described in this model ensure that exactly m robots leave and return to the warehouse, in addition, the maximum and minimum number of nodes visited by each robot is limited, which prohibits a robot from visiting only one node and restricting the formation of subtours between nodes. The ILP model is described next, considering the nomenclature (Table 4) and the mathematical notation (Kara & Bektas, 2006).

Sets and Index	
V	set of *n* nodes (vertices)
i	origin node
j	destination node
A	set of arcs *arc (i, j)*
Parameters	
c_{ij}	cost (distance) matrix associated with each arc $(i, j) \in A$
m	robot located at the depot 1
L	maximum number of nodes a robot may visit
K	minimum number of nodes a robot must visit
Decision Variables	
x_{ij}	1 if arc *(i, j)* is in the optimal solution 0 otherwise
u_i	number of nodes visited on that robot path from the origin up to node *i* • $1 \leq u_i \leq L$ for all $i \geq 2$ • if $x_{i1} = 1$, then $K \leq ui \leq L$

Table 4. ILP nomenclature for the multiple traveling salesmen problem (mTSP) (Kara & Bektas, 2006)

Minimize

$$\sum_{(i,j)\in A} c_{ij} x_{ij} \tag{1}$$

s.t

$$\sum_{j=2}^{n} x_{1j} = m, \tag{2}$$

$$\sum_{j=2}^{n} x_{j1} = m, \tag{3}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \quad j = 2, \ldots, n, \tag{4}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \quad i = 2, \ldots, n, \tag{5}$$

$$u_i + (L-2)\, x_{i1} \leq L-1, \quad i = 2, \ldots, n, \tag{6}$$

$$u_i + x_{i1} + (2-K)\, x_{i1} \geq 2, \quad i = 2, \ldots, n, \tag{7}$$

$$x_{1i} + x_{i1} \leq 1, \quad i = 2, \ldots, n, \tag{8}$$

$$u_i - u_j + L x_{ij} + (L-2)\, x_{ji} \leq L-1, \quad 2 \leq i \neq \leq n \tag{9}$$

$$x_{ji} \in \{0,1\}, \quad \forall (i,j) \in A \tag{10}$$

According to (Kara & Bektas, 2006) the ILP model formulation is valid when $2 \leq K \leq [(n-1)/m]$ and $L \geq K$. When $K \geq 4$, constraints (6) and (7) do not allow the situation $x_{1i} = x_{i1} = 1$, i.e., constraint (8) becomes redundant when $K \geq 4$. Thus, we need constraint (8) only for the cases K=3 or K=2. In this formulation, constraints (2) and (3) ensure that exactly m robots leave from and return to the depot. Constraints (4) and (5) are the degree constraints, the inequalities given in (6) and (7) serve as upper and lower bound constraints on the number of nodes visited by each robot, and initialize the value of u_i to 1 if and only if i is the first node on the tour for any robot. Inequality (8) forbids a robot from visiting only a single node. The inequalities given in (9) ensure that $u_j = u_i + 1$ if and only if $x_{ij} = 1$. Thus, they prohibit the formation of any subtour between nodes in V\{1}, so they are the subtour elimination constraints (SECs) of the formulation.

4.2.2. Genetic Algorithm

The problem to be solved in the mTSP considers that m robots must visit n points, which are not initially assigned to any particular robot. When the calculation time of the exact method (ILP model) is computationally inefficient, due to the high number of destinations and robots, a genetic algorithm (GA) is proposed. The proposed GA, based on Valero-Gomez et al. (2011) with some modifications, is built under the following characteristics:

Individuals notation:

- m: robot index ∈ [0, m-1]
- n: target points index ∈ [m, m+n-1]

Chromosome: The chromosome has the following characteristics

- The chromosome individuals can refer to a robot index or a target point index
- Individuals cannot be repeated in the chromosome
- The sequence of points assigned to each robot starts on the target points beginning after robots' index and finishes when the next robot index is found (when the end of the chromosome is reached, the sequence corresponding to the last robot continues with the first individual of the chromosome).

Initial Population: A random population is generated guaranteeing the non-repetition of chromosomes.

Crossover: A order crossover is used. two parents are selected, and from one of them the crossover operator selects a succession of individuals (the number of individuals to select is randomly generated). The child chromosome is produced by copying the selected succession of individuals (substring) into the same location as it was in the first parent chromosome. Then the crossover operator fills the empty individuals of the child chromosome starting from the first non-used individual of the second parent chromosome.

Selection: a selection procedure is considering a probabilistic function used as a fitness function equation based on the proposal of (Valero-Gomez et al., 2011):

$$Fitness(indi) = \sum_{i=0}^{m-1} (R_i) + 0.9 \; x \; max(R_i) - 0.9 \; x \; (R_i) \tag{11}$$

where, R_i is the total length of the route assigned to robot i. This function seeks to minimize the total length but also to maximize the length of the shortest individual path and to minimize the longest individual path, therefore all robots will be assigned a path of similar length.

Mutation: Individuals to mutate are chosen randomly among the whole offspring with a uniform probability. The mutation probability is set to 1.

Insertion: Both, the normal child chromosome and the mutated one are candidates to be inserted in the population. A previous check is done in order to avoid the insertion of an already existing chromosome in the population. Each new member of the population is inserted by replacing the worst one.

Termination rule: GA calculation ends when there is not improvement in the best fitness of the population after a number of generations.

4.3. Endorse Broker

The Endorse Broker provides a message bus to exchange information between the robotic fleet and the FMS in a secure way. The Endorse Broker is a Message Queuing Telemetry Transport (MQTT) (Banks & Gupta, 2014) broker used to exchange messages defined according to a proprietary message protocol (Endorse Protocol).

The Endorse Protocol defines a messaging pattern where the robots and modules can receive commands (e.g. to tell a robot to start or cancel a mission) from the FMS, provide feedback information on their environment (e.g. update the navigation map, the battery status and current position), as well as to report errors.

Robots and modules publish messages in the message broker and subscribe to the topics they are interested in. Therefore, all communications are initiated in the field level of the robotic fleet. This facilitates network management and security, because all the robots and modules can be grouped into separate network segments behind a firewall with no incoming communications.

4.4. Backend API

The backend API provides a REST interface to manage the master data of the FMS, to calculate an optimal assignment of a set of tasks to a group of robots to be executed on a specific date and time, and to add a new task to be executed in the current shift.

Master data includes user information (user roles), robotic fleet master data (e.g. types of robots, robot groups and robot members), navigation information (navigation map, location markers, etc), and task information (types of task, task command execution plans, etc). Backend API consumers (i.e. users via the frontend interface or other systems or applications integrated with the FMS) can apply CRUD (Create Request Update or Delete) to manage master data.

The REST backend API provides an endpoint to calculate a new work plan. A work plan consists of a set of tasks to be executed by a group of robots during a shift at a specific date. To process calls to this endpoint, the FMS solves the planning problem as described above and returns the results: Task allocated to each robot and estimated completion time. Consumers can edit and validate the work plan so that it is later processed by the Controller.

The REST backend API also provides an endpoint to add a new task to the current plan. When a consumer uses this endpoint, the FMS first solves again the planning problem adding the new task to the list of pending tasks.

4.5. Controller

The Controller implements the rules to ensure the execution of the work plan, that is, ensure that the robots execute the tasks that have been assigned by the task scheduler. The controller implements a control loop between the task scheduler and the robotic fleet via the message broker. First, the controller loads the current plan that has been validated by the user and builds a model of the execution of the work plan by robots. The controller sends the commands to start the work plan to the robotic fleet via the message bus. Periodically, the robots report their respective status (current location, current command, percentage of completion of current command, current battery level). Upon reception of the reports, the controller uses the backend to calculate a new work plan with updated status information, compares the result to the previous setup and if need be, sends the (delta) commands to the robots to re-schedule the tasks. This control loop somewhat resembles a Model Predictive Control (MPC) (Kim, Necsulescu & Sasiadek, 2007) control loop using the path planning solution as optimizer.

Note that in this process the task scheduler, the FMS controller, and the robot control work at different levels of granularity. The task scheduler manages tasks for the entire robotic fleet. The controller processes low level commands that together compose a task. For instance, moving a robot through the map may imply several GOTO commands to ensure that the robot follows the optimal path according to the cost function. Robots on the other hand are able to take

local decisions to avoid obstacles and avoid blocking situations. Robots are better at taking local decisions, but they can also benefit from the global perspective and coordination capabilities of the FMS. Clearly, there is a trade-off between both and therefore the level of granularity used at the controller to control the robot path planning can be configured to achieve optimal results. It is important however to bound the time consumed by the task scheduler to generate the optimal plan, because it determines the control loop cycle time and in consequence, the level of granularity of the FMS controller. Next sections present the results of some computational experiments to better manage this trade-off.

5. Computational Experiments

In order to validate the proposed algorithm, a set of experiments have been carried out obtaining a numerical comparison between the IPL (Kara & Bektas, 2006) and the GA (Valero-Gomez et al., 2011). We perform a set of experiments with different values were the number of robots m = {5, 10, 15, 20} and the number of destination locations n = {20, 40, 60, 80}. The parameters of the genetic algorithm were: termination condition was to reach a Gap less than 1% in regard to the objective function value obtained from the ILP model, and the population size for all tests was 50. The variables measured were the cost of the solution (calculated with equation 11), the path length of each robot, the number of generations to find the solution, and CPU time. The algorithm was implemented in Python 3.8.2 and the ILP model was tested using Gurobi 9.0, running on a PC equipped with an Intel (R) Core (TM) I5-8500 @ 3.00 GHz Processor and 8 GB RAM.

The results presented in Table 5 depict the CPU time and the gap considering different instances that are characterised by the number of robots and the number of target points. The gap (in %) between the ILP and the GA solutions is calculated using the following formula:

$$Gap = \frac{GA\,(best\ solution) - ILP\ solution}{ILP\ solution} \tag{12}$$

For small instances (5 robots and 20 locations), the GA required a CPU time of fewer than 10 seconds with a GAP of less than 1%, but the ILP model computation time is better and reach an optimal result, however, for large cases consisting of 15 robots and 60 locations, or 20 robots and 80 locations, the ILP consumed the defined set time of 7,200 seconds. The genetic algorithm in these cases reached almost optimal solutions in less time (188 and 354 seconds respectively), which shows that it is useful for real cases.

Figure 4 shows the speed of the GA model and the ILP model. It can be observed that the ILP model achieve optimal results for small and medium instances but, for large instances, the computation time is not efficient. As in real applications, such as hospital spaces or dynamic environments, the computation time is a significantly important parameter, the GA can be used to obtain near-optimal or good solutions with less computation efforts.

Instances	Gurobi (ILP model)			Genetic Algorithm		Gap (%)
INS_m_n	OBJ	CPU Time	Status	Best-obj.	CPU Time	
INS_5_20	686	0.12	Optimal solution	692	9.56	0.875%
INS_10_40	1844	256.58	Optimal solution	1856	21.02	0.651%
INS_15_60	3038	7201.08	Time limit exceeded	3052	188.18	0.461%
INS_20_80	4574	7201.65	Time limit exceeded	4614	354.15	0.875%

Table 5. Detailed comparison between GA and ILP model

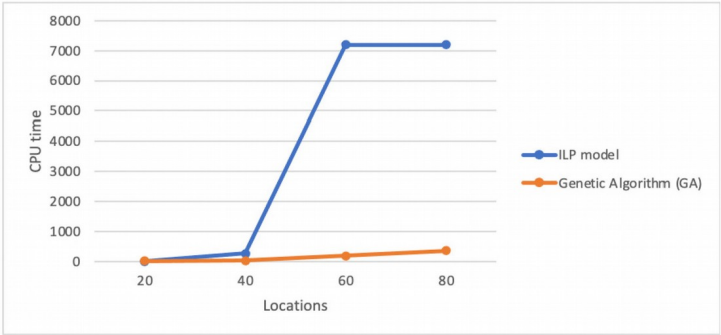

Figure 4. CPU Time comparison between the ILP model and the GA

6. Conclusions

This paper outlines the state of the art performed within the topic under review, the mobile robot path planning problem in interior spaces. A taxonomy of algorithms is used to classify the solution approaches proposed to address the aforementioned problem. From the review we can conclude that there are several algorithms that address the dynamic path planning problem of mobile robots in healthcare environments. The most used is the A * algorithm, providing good solutions in short calculation times. The review was carried out considering the real applications not only in the healthcare environment but also in other interesting contexts, due to its potential to adapt to healthcare environment. In this regard, choosing an algorithm for route planning will allow the planner to design routing policies that can be useful and flexible in a dynamic environment, like hospitals, in which people or objects that are constantly moving and changing positions.

In addition, this paper describes the implementation of a Fleet Management System (FMS) that plans and controls the execution of logistic tasks by a set of mobile robots, as a result of the work carried out in ENDORSE project. ENDORSE focuses on mobile robots in healthcare and commercial spaces environments. As a result of the FMS, five tools have been developed: the routing engine and the task scheduler, the FMS Endorse broker, the backend API and the controller.

The routing engine integrates the environment modelling, though the interpretation of geo-referenced data. Moreover, it computes the costs of moving from one point to another in the floor-plants. With regards the routing engine, the main drawback of the current approach is that the user needs to edit the map manually using a CAD tool, which might be time consuming and not very user friendly. In next releases, the routing engine will be integrated with a navigation system (Anyplace) (Zeinalipour-Yazti & Laoudias, 2017) which is also part of the solution. With this integration, users will be able to create the map using a mobile phone application that allows to tag locations just using the phone.

The task executor contains a mTSP ILP that minimizes largest routes. Moreover, the task executor also has a genetic algorithm to deal with mTSP when a greater number of robots and locations is given, making the problem computationally efficient. In the scope of the task executor, two future research lines are identified. Firstly, future work should be aimed at proposing algorithms to address the problem of path planning with multiple robots in dynamic environments, paying special attention to the simulation and application in real cases in healthcare sector. Second research line leds to balance the routes considering a minimum and maximum of places to visit for each robot, minimizing the longest routes.

Declaration of Conflicting Interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This work has received funding from the European Union's Horizon 2020 re-search and innovation programme under the Marie Skłodowska-Curie grant agreement No 823887. The authors would like to acknowledge the

support of the researchers participating in the collaborative project "Safe, Efficient and Integrat-ed Indoor Robotic Fleet for Logistic Applications in Healthcare and Commercial Spaces" (ENDORSE) (www.endorse-project.eu); and the Conselleria de Educación, Investigación, Cultura y Deporte - Generalitat Valenciana for hiring pre-doctoral research staff with Grant (ACIF/2018/170); European Social Fund with Grant Operational Program of FSE 2014-2020, the Valencian Community.

References

Abdulla, A.A., Liu, H., Stoll, N., & Thurow, K. (2017). A backbone-floyd hybrid path planning method for mobile robot transportation in multi-floor life science laboratories. *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, Mfi* (406-411). https://doi.org/10.1109/MFI.2016.7849522

Amanatiadis, A., Henschel, C., Birkicht, B., Andel, B., Charalampous, K., Kostavelis, I. et al. (2015). AVERT: An autonomous multi-robot system for vehicle extraction and transportation. *Proceedings - IEEE International Conference on Robotics and Automation,* (1662-1669). https://doi.org/10.1109/ICRA.2015.7139411

Andres, B., Sanchis, R., & Poler, R. (2016). A Cloud Platform to support Collaboration in Supply Networks. *International Journal of Production Management and Engineering,* 4(1), 5. https://doi.org/10.4995/ijpme.2016.4418

Banks, A., & Gupta, R. (2014). *MQTT Version 3.1.1*. OASIS Standard.

Bardram, J., & Bossen, C. (2005). Mobility work: The spatial dimension of collaboration at a hospital. *Computer Supported Cooperative Work: CSCW: An International Journal,* 14(2), 131-160. https://doi.org/10.1007/s10606-005-0989-y

Choi, B., Lee, J., Lee, J., & Park, K. (2011). A Hierarchical Algorithm for Indoor Mobile Robot Localization Using RFID Sensor Fusion. *IEEE Transactions on Industrial Electronics,* 58(6), 2226-2235. https://doi.org/10.1109/TIE.2011.2109330

Chung, S.Y., & Huang, H.P. (2011). Robot Motion Planning in Dynamic Uncertain Environments. *Advanced Robotics,* 25(6-7), 849-870. https://doi.org/10.1163/016918611X563337

Da Mota, F.A.X., Rocha, M.X., Rodrigues, J.J.P.C., De Albuquerque, V.H.C., & De Alexandria, A.R. (2018). Localization and Navigation for Autonomous Mobile Robots Using Petri Nets in Indoor Environments. *IEEE Access,* 6, 31665-31676. https://doi.org/10.1109/ACCESS.2018.2846554

Das, P.K., Behera, H.S., Jena, P.K., & Panigrahi, B.K. (2016). Multi-robot path planning in a dynamic environment using improved gravitational search algorithm. *Journal of Electrical Systems and Information Technology,* 3(2), 295-313. https://doi.org/10.1016/j.jesit.2015.12.003

Elshamli, A., Abdullah, H., & Areibi, S. (2004). Genetic algorithm for dynamic path planning. In *Canadian Conference on Electrical and Computer Engineering 2004* (IEEE Cat. No. 04CH37513) (2, 677-680). Niagara Falls, Ontario, Canada. https://doi.org/10.1109/CCECE.2004.1345203

Faridi, A.Q., Sharma, S., Shukla, A., Tiwari, R., & Dhar, J. (2018). Multi-robot multi-target dynamic path planning using artificial bee colony and evolutionary programming in unknown environment. *Intelligent Service Robotics,* 11(2), 171-186. https://doi.org/10.1007/s11370-017-0244-7

Fung, W.K., Leung, Y.Y., Chow, M.K., Liu, Y.H., Xu, Y., Chan, W. et al. (2003). Development of a Hospital Service Robot for Transporting Task *. *IEEE International Conference on Robotics, Intelligent Systems and Signal Processing*, 1, 628-633. https://doi.org/10.1109/RISSP.2003.1285647

Gross, H., Boehme, H., Schroeter, C., Mueller, S., Koenig, A., Einhorn, E., et al. (2009). TOOMAS: Interactive Shopping Guide Robots in Everyday Use - Final Implementation and Experiences from Long-term Field Trials. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2005-2012). https://doi.org/10.1109/IROS.2009.5354497

Haj Darwish, A., Joukhadar, A., & Kashkash, M. (2018). Using the Bees Algorithm for wheeled mobile robot path planning in an indoor dynamic environment. *Cogent Engineering*, 5(1). https://doi.org/10.1080/23311916.2018.1426539

Huang, H., Huang, P., Zhong, S., Long, T., Wang, S., Qiang, E. et al. (2019). Dynamic path planning based on improved D*algorithms of Gaode map. 2019 *IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC* (1121-1124). https://doi.org/10.1109/ITNEC.2019.8729438

Huang, X., Cao, Q., & Zhu, X. (2019). Mixed path planning for multi-robots in structured hospital environment. *The Journal of Engineering,* 2019(14), 512-516. https://doi.org/10.1049/joe.2018.9409

Jeon, S., Lee, J., & Kim, J. (2017). Multi-robot task allocation for real-time hospital logistics. *2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017* (2465-2470). https://doi.org/10.1109/SMC.2017.8122993

Kara, I., & Bektas, T. (2006). Integer linear programming formulations of multiple salesman problems and its variations. *European Journal of Operational Research,* 174(3), 1449-1458. https://doi.org/10.1016/j.ejor.2005.03.008

Khatib, O., Quinlan, S., & Williams, D. (1997). Robot planning and control. *Robotics and Autonomous Systems,* 21(3), 249-261. https://doi.org/10.1016/S0921-8890(96)00078-4

Kim, B., Necsulescu, D., & Sasiadek, J. (2007). Autonomous mobile robot model predictive control. *International Journal of Control,* 71(16), 1438-1445. https://doi.org/10.1080/00207170412331317738

Koubaa, A., Bennaceur, H., Chaari, I., Trigui, S., Sriti, A.A.M.-F., Ammar, A. et al. (2018). *Robot Path Planning and Cooperation Foundations, Algorithms and Experimentations, XXII* (Issue 2). https://doi.org/10.1353/sais.2002.0030

Langerwisch, M., & Wagner, B. (2011). Dynamic path planning for coordinated motion of multiple mobile robots. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* (1989-1994). https://doi.org/10.1109/ITSC.2011.6083047

Li, S., Xu, X., & Zuo, L. (2015). Dynamic Path Planning of a Mobile Robot with Improved Q-Learning algorithm. *2015 IEEE International Conference on Information and Automation* (409-414). https://doi.org/10.1109/ICInfA.2015.7279322

Liu, H., Stoll, N., Junginger, S., & Thurow, K. (2012). A floyd-genetic algorithm based path planning system for mobile robots in laboratory automation. *2012 IEEE International Conference on Robotics and Biomimetics, ROBIO 2012 Conference Digest* (1550-1555). https://doi.org/10.1109/ROBIO.2012.6491188

Liu, H., Stoll, N., Junginger, S., & Thurow, K. (2013). Mobile robot for life science automation. *International Journal of Advanced Robotic Systems,* 10, 1-14. https://doi.org/10.5772/56670

Ma, Q., & Lei, X. (2010). *Dynamic path planning of mobile robots based on ABC algorithm.* Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 6320 LNAI(Part 2), 267-274. https://doi.org/10.1007/978-3-642-16527-6_34

Miao, H., & Tian, Y.C. (2013). Dynamic robot path planning using an enhanced simulated annealing approach. *Applied Mathematics and Computation,* 222, 420-437. https://doi.org/10.1016/j.amc.2013.07.022

Nazarahari, M., Khanmirza, E., & Doostie, S. (2019). Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. *Expert Systems with Applications,* 115, 106-120. https://doi.org/10.1016/j.eswa.2018.08.008

Othman, M.F., Samadi, M., & Asl, M.H. (2013). Simulation of Dynamic Path Planning for Real-Time Vision-Base Robots. In Omar, K., Nordin, M.J., Vadakkepat, P., Prabuwono, A.S., Abdullah, S.N.H.S., Baltes, J. et al. (Eds.), *Intelligent Robotics Systems: Inspiring the NEXT* (1-10). Springer Berlin Heidelberg.

Raja, P., & Pugazhenthi, S. (2011). Path planning for a mobile robot in dynamic environments. *International Journal of the Physical Sciences,* 6(20), 4721-4731. https://doi.org/10.5897/IJPS11.573

Ramdani, N., Panayides, A., Karamousadakis, M., Mellado, M., Lopez, R., Christophorou, C., et al. (2019). A safe, efficient and integrated indoor robotic fleet for logistic applications in healthcare and commercial spaces: The endorse concept. *Proceedings - IEEE International Conference on Mobile Data Management, (MDM)* (425-430). https://doi.org/10.1109/MDM.2019.000-8

Serpen, G., & Dou, C. (2015). Automated robotic parking systems: real-time, concurrent and multi-robot path planning in dynamic environments. *Applied Intelligence,* 42(2), 231-251. https://doi.org/10.1007/s10489-014-0598-x

Shi, P., & Cui, Y. (2010). Dynamic path planning for mobile robot based on genetic algorithm in unknown environment. *Chinese Control and Decision Conference, CCDC* (4325-4329). https://doi.org/10.1109/CCDC.2010.5498349

Shieh, M.Y., Hsieh, J.C., & Cheng, C.P. (2004). Design of an intelligent hospital service robot and its applications. *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics* (5, 4377-4382). https://doi.org/10.1109/ICSMC.2004.1401220

Song, X., Gao, S., Cao, K., & Huang, J. (2018). A new hybrid method in global dynamic path planning of mobile robot. *International Journal of Computers, Communications and Control,* 13(6), 1032-1046. https://doi.org/10.15837/ijccc.2018.6.3153

Thurow, K., Zhang, L., Liu, H., Junginger, S., Stoll, N., & Huang, J. (2019). Multi-floor laboratory transportation technologies based on intelligent mobile robots. *Transportation Safety and Environment,* 00(00), 1-17. https://doi.org/10.1093/tse/tdy002

Tuncer, A., & Yildirim, M. (2012). Dynamic path planning of mobile robots with improved genetic algorithm. *Computers and Electrical Engineering,* 38(6), 1564-1572. https://doi.org/10.1016/j.compeleceng.2012.06.016

Tzafestas, S.G. (2018). Mobile Robot Control and Navigation: A Global Overview. *Journal of Intelligent and Robotic Systems: Theory and Applications,* 91(1), 35-58. https://doi.org/10.1007/s10846-018-0805-9

Valero-Gomez, A., Valero-Gomez, J., Castro-Gonzalez, A., & Moreno, L. (2011). Use of genetic algorithms for target distribution and sequencing in multiple robot operations. *2011 IEEE International Conference on Robotics and Biomimetics, ROBIO* (2718-2724). https://doi.org/10.1109/ROBIO.2011.6181716

Vartholomaios, P., Ramdani, N., Christophorou, C., Georgiadis, D., Guilcher, T., Blouin, M., et al. (2019). ENDORSE Concept An Integrated Indoor Mobile Robotic System for Medical Diagnostic Support Panagiotis. *International Journal of Reliable and Quality E-Healthcare,* 8(3), 47-59. https://doi.org/10.4018/ijrqeh.2019070104

Wang, J., Wei, B.Y., Zhang, Y., & Chen, H. (2009). Design of an autonomous mobile robot for hospital. *ITME2009 - Proceedings 2009 IEEE International Symposium on IT in Medicine and Education* (1, 1181-1186). https://doi.org/10.1109/ITIME.2009.5236275

Yuan, F., Twardon, L., & Hanheide, M. (2010). Dynamic path planning adopting human navigation strategies for a domestic mobile robot. *IEEE/RSJ 2010 International Conference on Intelligent Robots and Systems, IROS* (3275-3281). https://doi.org/10.1109/IROS.2010.5650307

Yun, S.C., Parasuraman, S., & Ganapathy, V. (2011). Dynamic path planning algorithm in mobile robot navigation. *2011 IEEE Symposium on Industrial Electronics and Applications, ISIEA* (364-369). https://doi.org/10.1109/ISIEA.2011.6108732

Zeinalipour-Yazti, D., & Laoudias, C. (2017). The anatomy of the anyplace indoor navigation service. *SIGSPATIAL Special*, 9(2), 3-10. https://doi.org/10.1145/3151123.3151125

Zhang, X., Zhao, Y., Deng, N., & Guo, K. (2016). Dynamic Path Planning Algorithm for a Mobile Robot Based on Visible Space and an Improved Genetic Algorithm. *International Journal of Advanced Robotic Systems,* 13(3). https://doi.org/10.5772/63484

Zhang, Y., & Wang, C. (2018). Navigation Scheme of Mobile Robots and Its Application at Airport Environment. *13th World Congress on Intelligent Control and Automation (WCICA)* (1379-1384).