Two agents with GBFS algorithms working cooperatively to get a shortest path

Dos agentes con algoritmos GBFS trabajando de forma cooperativa para obtener la ruta más corta

J. A. Chaves-Osorio; J. B. Gómez-Mendoza; E. A. González-Ríos

Abstract— **This study is carried out in order to verify if the implementation of the concept of cooperative work among two agents, that use path planners A* to obtain the shortest path (previous work of the authors) is also valid when the cooperative strategy is applied using another path planner such as the so-called GBFS (Greedy Best First Search). In this sense, this paper shows a path planning strategy that combines the capabilities of two Agents each one with its own path planner GBFS (slightly different from each other) in order to obtain the shortest path. The comparisons between paths are made by analyzing the behavior and results obtained from the agents operating in different forms: (1) Working individually; (2) Working as a team (cooperating and exchanging information). The results show that in all analyzed situations are obtained shortest traveled distances when the path planners work as a cooperative team.**

Index Terms— **Autonomous robots; navigation; particle tracking; path planning; teamwork.**

Resumen— **Este estudio se lleva a cabo con el fin de verificar si la implementación del concepto de trabajo cooperativo entre dos agentes, usado con planificadores A* para obtener la ruta más corta (trabajo previo de los autores) también es válida cuando la estrategia cooperativa es aplicada usando otro planificador de rutas como el llamado GBFS (Greedy Best First Search). En este sentido, el articulo muestra una estrategia de planificación de rutas que combina las capacidades de dos agentes cada uno con su propio planificador de rutas GBFS (ligeramente diferentes entre sí) para obtener la ruta más corta. La comparación entre las dos rutas se realiza analizando el comportamiento y comparando los resultados obtenidos para cada uno de los que operan en diferentes formas: (1) Trabajando individualmente; (2) Trabajando como un equipo (cooperando e intercambiando información). Los resultados muestran que para todos los casos analizados se obtiene la distancia recorrida más corta cuando los planificadores de ruta trabajan como un equipo colaborativo.**

Palabras clave— **Navegación; planificación de rutas; Robots autónomos; seguimiento de partículas; trabajo en equipo.**

I. Introduction

Path planning has been a very important topic in autonomous mobile robot's research. As the robots must reach the destination in the best conditions, some of the principal requirements for the path planners are: (1) avoid colliding, (2) to get shortest paths. Many techniques are used to path planning, described in a wide variety of papers, in some cases are considered the robot movement restrictions, even a particular robot like Kephera II [1]-[2] and in another cases are considered the robot like a point (located in the robot mass center) [3]-[5]. In this work, two path planners GBFS are used to determine a recommended path, one without collisions (using offset and sensors [23]) and with a shortest traveled distance (using cooperative work [6]). The hypothesis to be verified in this paper, is related to the fact that, if the GBFS planner point of view is changed (switching start and goal) two paths are obtained (using two different path planners GBFS). Then comparing and combining the obtained paths, it is possible to obtain a third path; a shortest path between Start point and Goal point. From this method, it is possible to estimate the shortest path from cooperation between agents (This hypothesis was verified using path planners A* in the previous work "A new approach of two Agents with path planners A* working cooperatively to get the shortest path")

A. Path Planning Algorithms

With the passage of time and constant evolution of the Artificial Intelligence (AI, for the case of the present paper related to algorithms of search), different algorithms for path planning have been created [7]-[12], some are deterministic and others stochastic [13]-[16].

Some of the most used algorithms are: (1) GBFS (Greedy Best First Search), (2) DFS (Depth First Search), (3) Dijkstra, (4) A*, (5) LPA*, (6) D*, (7) D* Lite, (8) Bug, among others; these

This manuscript was submitted on May 28, 2020 and accepted for publication on September 28, 2020.

This work has been supported by Tecnoacademia Manizales (Research Group EAYER), Universidad Tecnológica de Pereira (Research Group GIRA) and Universidad Nacional de Colombia (Research Group SHAC).

J.C. Author is Ph.D student in Engineering from Universidad Nacional de Colombia, is M.Sc. in in Physics Instrumentation from Universidad Tecnológica de Pereira, and professor of same University (e-mail: jachaves@utp.edu.co, jachaveso@unal.edu.co).

J.G. Author is Ph.D in Engineering from Universidad Nacional de Colombia and professor of same university (e-mail: jbgomezm@unal.edu.co).

E.G. Author is M.Sc. in Physics Instrumentation, professor of robotics in Tecnoacademia Manizales, Industrial Automatization Center, SENA Caldas, Colombia (e-mail: edgonzales@sena.edu.co).

algorithms are different from each other and each one has advantages and disadvantages compared to the others.

B. GBFS Algorithm (Greedy Best Firs Search Algorithm)

The algorithm GBFS uses a heuristic function, which takes into account the next distances:

• h(n), which is an estimate of the distance from the current state of the intelligent agent or explorer to the desired state, this estimate is commonly made using concepts such as Euclidean distance or Manhattan distance, is performed depending on the application where the algorithm GBFS to be implemented.

• n, denotes the nodes of the space to explore.

Therefore, the typical heuristic function for GBFS, is shown in (1).

$$f(n) = h(n) \tag{1}$$

However, in this document is added a new element p(n) defined as:

• p(n), which is a cost function that increases the value of a node by 1 each time it is visited; adding this function is intended to prevent the planner from returning over nodes or paths already traveled. Therefore, the heuristic function for a new GBFS, shows in (2).

This heuristic function is applied to each of the successor states of the current state and is chosen as the next best step to the one containing the smallest value of f(n).

II. Methodology

This section describes the principles that define this novel cooperative teamwork between two Agents with path planners GBFS. In this work, two GBFS path planners are used because GBFS is an algorithm deeply tested in the state of art, it is stable, its implementation is easy, it has not lost its usability from its creation; besides is a fast algorithm that consumes few processing resources and is used for researchers in different current applications [17], [18].

A. Considerations for proposed algorithm GBFS

Initially a traditional GBFS path planner was built implementing the pseudo code proposed in the Algorithm shown in Fig. 1.

```
Pseudocode for algorithm IH-GBFS
  textbfInivar() $\Rightarrow$ Initialize
$E = MxN$
$Vroute = [\ ]$
$n = n_s \in E$
  $C(\ ) \Rightarrow$ Cost function
$h(n', n_g) = d_E(n', n_g)$
$C(n, n') = f(n, n') = h(n) + p(n)$
  $Planif(\ ) \Rightarrow$ Search process
$n' = Succ(n)$
$Vroute = C(n, n') \forall n' \in E$
$Bpi = \min(Vopen) \in E$
$n = n_i^{'} \mid n_i \exists B_{pi} \in E$
$return(n)$
  $Main(\ ) \Rightarrow$ Main Process
$Ini\,\mathrm{var}(\ )$
$for(1)$
    $Planif(\ )$
    $if(n == n_g)$
        End of the search
    $end\ if$
    Update vertex
$end\ for$
```

Fig. 1 Pseudocode for a traditional algorithm GBFS

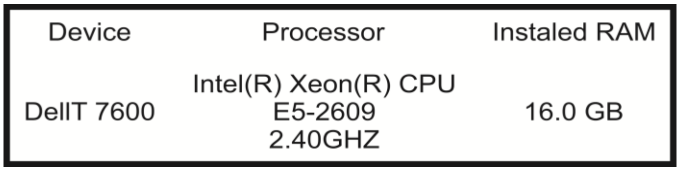
Then, as already mentioned, A1 and A2 are two Agents, which, to fulfill the tasks of exploration and path planning using GBFS algorithms, on which, some modifications were made, as adding it memory related to past events, shifting the old GBFS in a new path planner with an incremental-heuristic search strategy shown in (2).

$$f(n) = h(n) + p(n) \tag{2}$$

As it is known, a characteristic that is considered important of the GBFS algorithm, is its low computation cost; due to the simplicity of its implementation, since it is a purely heuristic path planner. However, for the realization of this work; it is necessary to store information during the exploration and path planning process, this new implementation allows to create a modified GBFS algorithm, similar to the A* algorithm. As shown in the introduction of this paper, in its B section, the heuristic function of algorithm GBFS is defined according to (1); where it can be observed that this function f(n)=h(n); is made up of one sub function h(n) that allow it to calculate the lowest path cost, measured in distance [19]. Other algorithms can base its heuristic function (cost function) on other variables associated to: computational time, control effort, energy consumption, among others.

B. GBFS implementation

The tests with GBFS algorithms were implemented in Matlab R2016a in a computer whose features are described in Fig 2.



Device	Processor	Instaled RAM
DellT 7600	Intel(R) Xeon(R) CPU E5-2609 2.40GHZ	16.0 GB

Fig. 2 Computer features

The algorithm was used on a test environment or map divided with a grid of 100 x 100 cells (as see in Fig. 3). In this stage, white points represents empty cells, black points represents obstacles and blue points represents the “offset” necessary to avoid collisions [20]-[22] between the robot and the boundary when the robot pass close to the edge of the obstacles, this consideration is important because is considered the robot dimensions and not only the robot like a point.

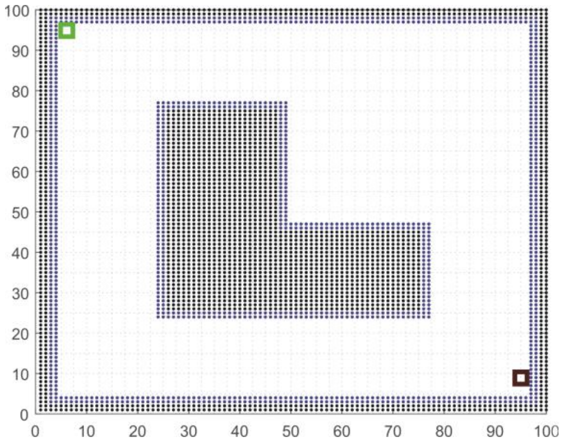


Fig. 3 Test environment

When GBFS is applied considering the start point (in green, see Fig. 4) and the goal point (or end point in brown, see Fig. 4); an orange path is created like one shown in Fig. 4; this path is stored for further processing.

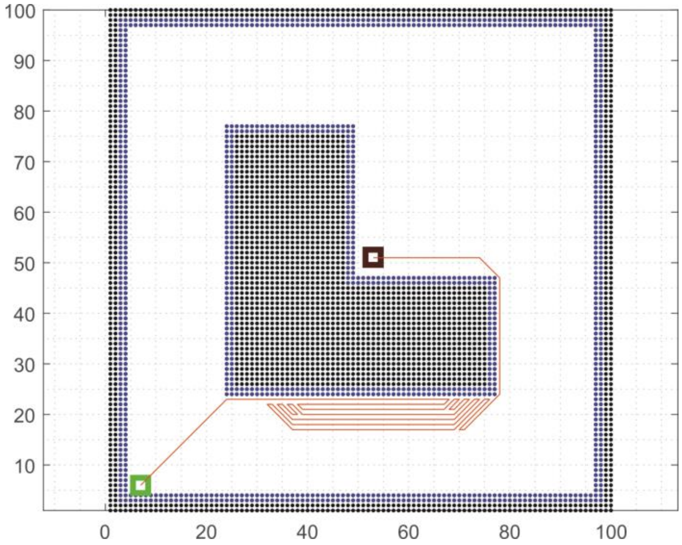


Fig. 4 First path planning

The previous process is repeated by other GBFS, applied switching start and goal; that is, generating a new path, calculated by making that in the previous path planning was considered a goal is the new start and vice versa.

Because of this commutation, a new path is obtained (as see in Fig. 5).

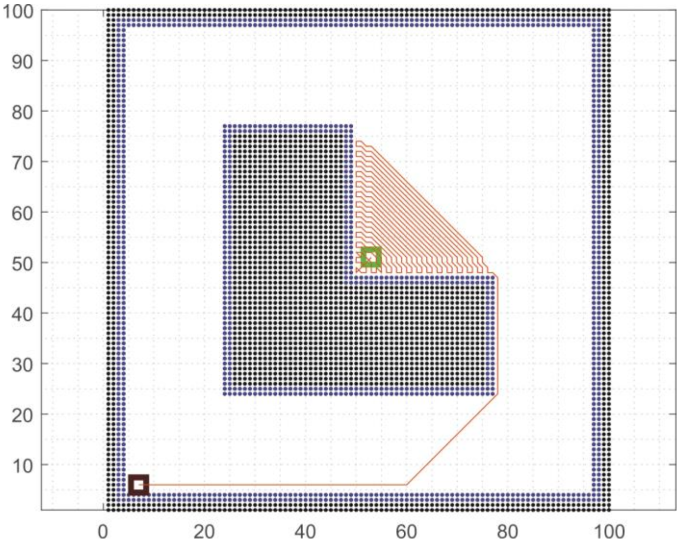


Fig. 5 Second path planning. Implementing GBFS by switching start and goal

Then, the last path (Fig. 5), can be compared and/or combined with the path obtained in previous step (Fig. 4) to obtain a new path called “Shortest estimated path” (as see in Fig. 6), this is recommended path [23].

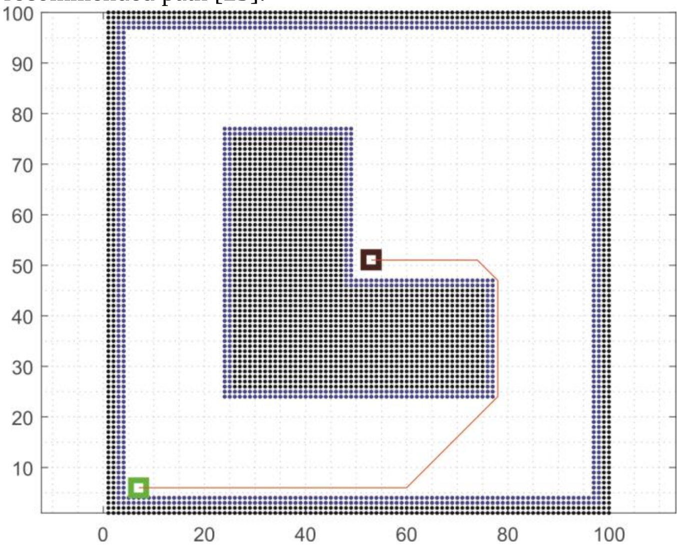


Fig. 6 Shortest estimated Path

In general, when two paths are compared, five situations can occur, for each one of these comparison situations, the actions defined in Table I are carried out to obtain a single path (one with the shortest distance):

TABLE I.
SITUATIONS AND ACTIONS
TO DETERMINE THE SHORTEST ESTIMATED PATH

	CASES	ACTION
1	The two paths obtained are exactly same.	**a) SEP selection:** The first path is simply chosen as the final path and this is the SEP. Although the second path could also have been chosen, if a different decision criteria were considered (fewer number of node, less control effort or lower energy consumption).
2	The two paths obtained are different from each other, their traveled distances are equal and the never intersect.	
3	The two paths obtained are different from each other, their traveled distances are different and they do not intersect.	**a) SEP selection:** The path with the shortest traveled distance is chosen as the SEP.
4	The two paths obtained are different from each other, their traveled distances are equal and the cross in come sections.	**a) Crossing points location:** Crossing points are located (C1, C2, …, Ci, Cj, …, Cn). **b) Section definition:** A section is defined as a pair of paths between crossing points Ci and Cj. **c) Comparison of paths:** The paths traveled distances by agents are compared in each section and the shortest one is selected. **d) Combination of paths**: The concatenation of previously selected paths with the shortest distances results in the definitive path called the SEP.
5	The two paths obtained are different from each other, their distances traveled are different and they intersect in some sections.	

As see, by technique used, the shortest estimated path let to obtain the shortest path between two paths obtained previously. Table II, shows an example of fields used as a comparative table with blanks (empty), used to compare traveled distances in each experiment.

TABLE II.
EMPTY TABLE USED TO COMPARE TRAVELED DISTANCES.

SEP	P1	P2	%P1-SEP	%P2-SEP

Where:

SEP: Distance of the Shortest Estimated Path in cell units
SG: Distance of Start to Goal in cell units
GS: Distance of Goal to Star in cell units
In addition, Table II includes percentages that relate the increase of SG and GS respect to SEP as shown in (3) and (4).

Where:

%P1-SEP: Percent increase of path planner 1 respect to SEP
%P2-SEP: Percent increase of path planner 2 respect to SEP

It is important to indicate that previous notation and analysis will be used in this entire document.

III. EXPERIMENTAL RESULTS

The purpose of the experiment was to probe the benefits of this novel technique to obtain the shortest estimated path using two path planners GBFS working collaboratively throughout a very simple strategy with three steps:

1. Implementing GBFS defining start and goal.

2. Implementing GBFS switching start and goal.

3. Obtaining the Shortest Estimated Path (SEP) by comparison and combination (if is possible) of two paths obtained previously.

It is important to indicate that in each case, in the experiments appears the traveled distance with a number in cell units, additionally, always for the shortest estimated path the "Start" and the "Goal" are the same that for the first implementation of GBFS.

A. Summary of analyzed cases

As see in Fig. 7, Fig.8 and Fig. 9 the experiments confirm that two Agents A1 and A2 with GBFS path planners working cooperatively make it possible to obtain the shortest estimated path.

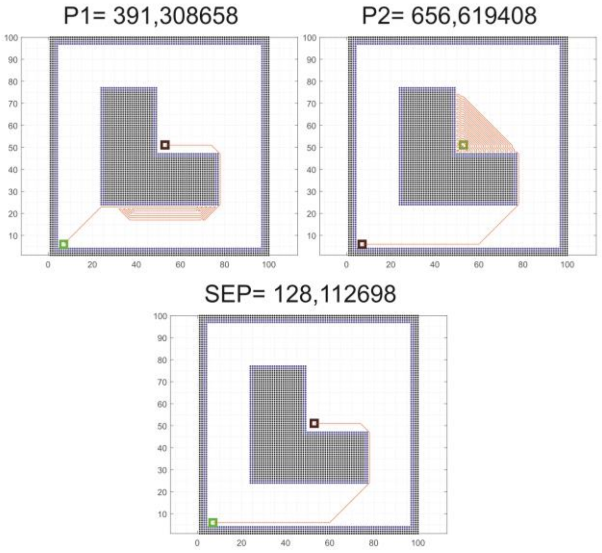


Fig. 7 Case 1 of analyzed cases.

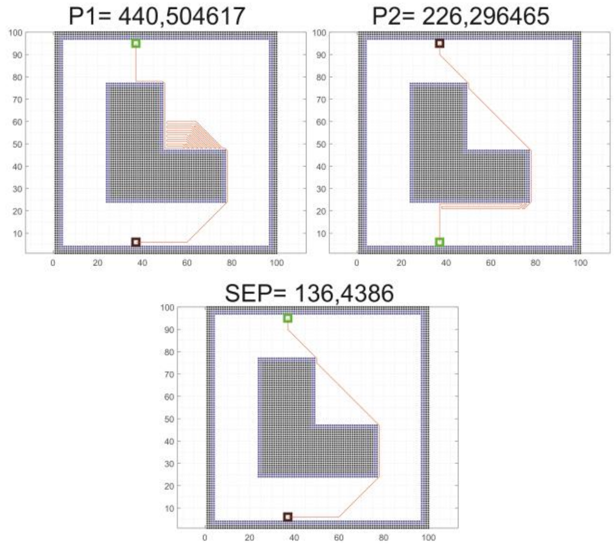


Fig. 8 Case 2 of analyzed cases.

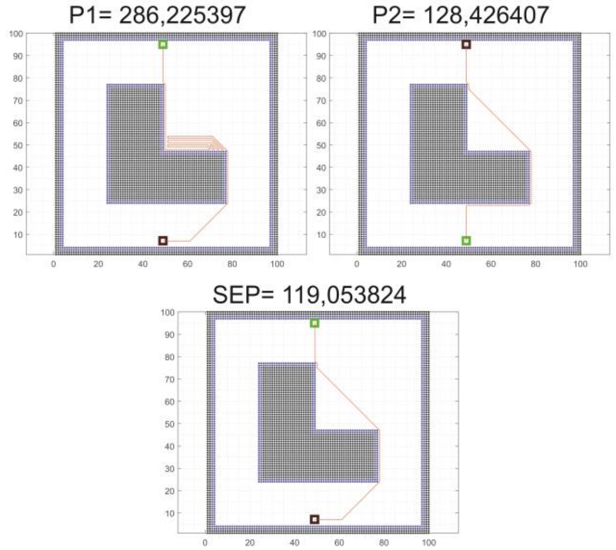


Fig. 9 Case 3 of analyzed cases.

TABLE III.
COMPARISON OF TRAVELED DISTANCES

CASE	SEP (cell)	P1 (cell)	P2 (cell)	%P1-SEP	%P2-SEP
1	128,112	391,308	656,619	205,44	412,53
2	136,438	440,504	226,296	222,86	65,86
3	119,053	286,225	128,426	140,42	7,87

When comparing the particular results obtained from the simulations carried out to get the shortest travel distance, the following general results can be drawn from the present study:

1. If the two paths obtained are the same, the first path is simply chosen as the best path; in this situation the best-estimated distance is the same that the previous paths. In a real system, the paths could also be compared according to other criteria such as low energy consumption, grass direction, or considering movement restrictions [24]-[28], among others (for example, if the robots are moving in a garden or soccer field). According with the last:

a. SG=GS=SEP, %SG-SEP=%GS-SEP=0.

2. If the two paths obtained are different from each other, different in travel distance and do not intersect, the path with the shorter travel distance is chosen as the best estimated path. According with the last:

a. If SG<GS then: SG=SEP, %GS-SEP>0, %SG-SEP=0.

b. If SG>GS then: GS=SEP, %SG-SEP>0, %GS-SEP=0.

3. If the two paths obtained are different from each other, equal in travel distance and do not intersected, the first path is simply chosen as the final path and therefore as the best estimated distance, in this situation the second path could also be chosen using another decision criterion (in a real system the paths could also be compared according to another criteria such as: low control effort, low energy consumption, path smoothness, among others). According with the last:

a. SG=GS=SEP, %SG-SEP=%GS-SEP=0.

4. If the two paths obtained are different from each other, equal in travel distance and crossed in some sections, after determining crossings between the paths, a combination of the paths is made using the points in common. This combination is obtained by comparing distances between sections and selecting the shortest distance between crossing points, at the end, the sections of paths are taken and "added" to obtain the definitive path, in this case the shortest estimated distance is the same that the previous paths. According with the last:

a. SG=GS=SEP, %SG-SEP=%GS-SEP=0.

5. If the two paths obtained are different from each other, their distances travel are different and intersect in some sections. Determining crossings between the paths, a combination of the paths is made using the points in common. This combination is obtained by comparing distances between sections and selecting the shortest distance between crossing points, at the end; the sections of paths are taken and "added" to obtain the definitive path with the shortest estimated distance. According with the last:

a. If SG>SEP, %SG-SEP>0.

b. If GS>SEP, %GS-SEP>0.

IV. CONCLUSIONS

In this paper, a method based in a cooperative work among two agents searching the shortest estimated path between two points is proposed. In this technique, both agents use modifications of the GBFS algorithm. Three cases are presented to evaluate the performance of proposed path planning method in comparison with the individual behavior of each agent with his own GBFS algorithm.

The findings of the experiments illustrate how two agents with path planners GBFS working cooperatively allow find the shortest estimated path. An implication of this study is the possibility of use another path planners to probe the proposed technique.

REFERENCES

[1] Kumar Das, Pradipta, S. N. Patro, C. N. Panda, and Bunil Balabantaray. "D* lite algorithm based path planning of mobile robot in static Environment." Int. J. Comput. Commun. Technol.(IJCCT) 2 (2011): 32-36.

[2] Vargas, Heriberto Casarrubias. "Generación de trayectorias para un robot móvil Khepera II usando técnicas de aprendizaje automático." (2007).

[3] Ospina, Alejandro González, C. Garzón, and H. Baldomiro. "Diseño, implementación y aplicación de una estrategia de búsqueda preferente por amplitud, para uso multidireccional sobre sistemas distribuidos o de procesamiento en paralelo, usando un simulador de escenarios, construido para el trazado de rutas en robótica móvil." (2011): 1-4.

[4] Koenig, Sven, and Maxim Likhachev. "D* Lite." Aaai/iaai 15 (2002).

[5] Koenig, Sven, Maxim Likhachev, and David Furcy. "Lifelong planning A*." Artificial Intelligence 155, no. 1-2 (2004): 93-146.

[6] Parker, Lynne E. "Distributed intelligence: Overview of the field and its application in multi-robot systems." Journal of Physical Agents 2, no. 1 (2008): 5-14.

[7] L. E. HAJJAMI, E. M. MELLOULI and M. BERRADA, "Neural Network Based Sliding Mode Lateral Control For Autonomous Vehicle," 2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET), Meknes, Morocco, 2020, pp. 1-6, DOI: 10.1109/IRASET48871.2020.9092055.

[8] Latombe, Jean-Claude. Robot motion planning. Vol. 124. Springer Science & Business Media, 2012. ISBN 978-1-4615-4022-9

[9] Garrido, Santiago, Luis Moreno, Dolores Blanco, and Marisa L. Munoz. "Sensor-based global planning for mobile robot navigation." Robotica 25, no. 2 (2007): 189-199. DOI: 10.1017/S0263574707003384

[10] Bruce, James, and Manuela Veloso. "Real-time randomized path planning for robot navigation." In IEEE/RSJ international conference on intelligent robots and systems, vol. 3, pp. 2383-2388. IEEE, 2002. DOI: 10.1109/IRDS.2002.1041624

[11] Murray, Richard M., and Sosale Shankara Sastry. "Nonholonomic motion planning: Steering using sinusoids." IEEE transactions on Automatic Control 38, no. 5 (1993): 700-716. DOI: 10.1109/9.277235.

[12] Fernández, Martín. "Algoritmos de búsqueda heurística en tiempo real. Aplicación a la navegación en los juegos de video." (2005).

[13] Cuchango, Helbert Eduardo Espitia, and Jorge Sofrony Esmeral. "Algoritmo para planear trayectorias de robots móviles, empleando campos potenciales y enjambres de partículas activas brownianas." Ciencia e Ingeniería Neogranadina 22, no. 2 (2012): 3. DOI:10.18359/rcin.242

[14] Garrido, Santiago, Luis Moreno, D. Blanco, and Fernando Martin. "Exploratory navigation based on Voronoi transform and fast marching." In 2007 IEEE International Symposium on Intelligent Signal Processing, pp. 1-6. IEEE, 2007. DOI: 10.1109/WISP.2007.4447541

[15] LaValle, Steven M. "Rapidly-exploring random trees: A new tool for path planning." (1998).

[16] Stentz, Anthony. Optimal and efficient path planning for unknown and dynamic environments. No. CMU-RI-TR-93-20. CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 1993.

[17] Goyal, Jitin Kumar, and K. S. Nagla. "A new approach of path planning for mobile robots." In 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 863-867. IEEE, 2014. DOI:10.1109/ICACCI.2014.6968200

[18] Duchoň, František, Andrej Babinec, Martin Kajan, Peter Beňo, Martin Florek, Tomáš Fico, and Ladislav Jurišica. "Path planning with modified a star algorithm for a mobile robot." Procedia Engineering 96 (2014): 59-69. DOI:10.1016/j.proeng.2014.12.098

[19] Russell, Stuart J., and Peter Norvig. Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited, 2016.

[20] Cho, Keum-Bae, and Seong-Yun Cho. "The concept of collision-free motion planning using a dynamic collision map." *International Journal of Advanced Robotic Systems* 11, no. 9 (2014): 145. DOI:10.5772/58707

[21] Spangelo, Inge, and Olav Egeland. "Trajectory planning and collision avoidance for underwater vehicles using optimal control." IEEE Journal of Oceanic Engineering 19, no. 4 (1994): 502-511. DOI:10.1109/48.338386

[22] Lozano-Pérez, Tomás, and Michael A. Wesley. "An algorithm for planning collision-free paths among polyhedral obstacles." Communications of the ACM 22, no. 10 (1979): 560-570. DOI:10.1145/359156.359164.

[23] Sai Varsha Konakalla, “A Star Algorithm” unpublished.

[24] López García, Diego Antonio. "Nuevas aportaciones en algoritmos de planificación para la ejecución de maniobras en robots autónomos no holónomos". 2011.

[25] López, D., F. Gómez-Bravo, F. Cuesta, and A. Ollero. "Planificación de trayectorias con el algoritmo RRT. Aplicación a robots no holónomos." Revista Iberoamericana de Automática e Informática Industrial 3, no. 3 (2010): 56-67.

[26] García, Diego A. López, Fernando Gómez Bravo, and Marcos del Toro Peral. "Comparativa entre planificadores de trayectorias para su uso combinado en la generación de maniobras". 2007.

[27] Laumond, Jean-Paul, Paul E. Jacobs, Michel Taix, and Richard M. Murray. "A motion planner for nonholonomic mobile robots." IEEE Transactions on robotics and automation 10, no. 5 (1994): 577-593.

[28] Fortune, Steven, and Gordon Wilfong. "Planning constrained motion." Annals of Mathematics and Artificial Intelligence 3, no. 1 (1991): 21-82.

José Andrés Chaves Osorio. (M’18) was born in Pasto, Nariño, Colombia in 1974. He received the degree in electrical engineering and the M.S. degree in Physical instrumentation from Universidad Tecnológica de Pereira (UTP), in Risaralda, Colombia, 1999 and 2010 respectively. He is currently pursuing the Ph.D. degree in engineering at Universidad Nacional de Colombia (UNAL), in Manizales, Caldas, Colombia.

Since 2007, he has been the Director of research group Robótica Aplicada. From 2003 to 2013, he was an adjunct professor and since august 2013, he has been an Assistant Professor in Basic Sciences and Engineering Faculties in UTP. He is the author of four books and more than 30 articles. His research interests include Robotics, artificial intelligence, and Internet of Things (IoT), applied physics, education in sciences and technology.

Prof. Chaves is a member of Asociación de Egresados de la Universidad Tecnológica de Pereira (ASEUTP) and has been a member of other associations like Asociación Colombiana de Ingenieros (ACIEM).
ORCID: https://orcid.org/0000-0002-9434-9224.

Juan Bernardo Gómez-Mendoza. (M’09). Mr. Gómez-Mendoza was born in Flandes, Colombia, in 1980. He received the degree in Electronics Engineering, the MEng. degree in Mathematics and a PhD degree in Engineering from Universidad Nacional de Colombia (UN), 2002, 2005 and 2012, respectively. He works as Professor for the Department of Electrical and Electronic Engineering at the same university since 2005. His current research interests include image processing, machine vision, machine learning and robotics.

Edward Andrés González Ríos. (M'19) was born in Pereira, Risaralda, Colombia in 1992. He received the degree in electronics engineering and the M.S. degree in Physical instrumentation from Universidad Tecnológica de Pereira (UTP), in Risaralda, Colombia, 2015 and 2019 respectively.

From 2016 to 2018 he was adjunct professor in Faculty of Engineering in UTP. Currently he is a professor of robotics and electronics in Industrial automation center in Servicio Nacional de Aprendizaje (SENA), Manizales, Caldas, Colombia. His research interests include Robotics, artificial intelligence and Internet of Things (IoT).

Prof. González is a member of Asociación de Egresados de la Universidad Tecnológica de Pereira (ASEUTP) and has been a member of other associations like Asociación Colombiana de Ingenieros (ACIEM).

ORCID: https://orcid.org/0000-0002-5500-5816.