Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2022.021979
Article

Tech Science Press

Autonomous Exploration Based on Multi-Criteria Decision-Making and Using D* Lite Algorithm

Novak Zagradjanin[1,*], Dragan Pamucar[2], Kosta Jovanovic[1], Nikola Knezevic[1] and Bojan Pavkovic[3]

[1]University of Belgrade, School of Electrical Engineering, Belgrade, 11120, Serbia
[2]University of Defense, Military Academy, Department of Logistics, Belgrade, 11000, Serbia
[3]Military Technical Institute, Belgrade, 11030, Serbia
*Corresponding Author: Novak Zagradjanin. Email: zagradjaninnovak@gmail.com
Received: 23 July 2021; Accepted: 06 September 2021

Abstract: An autonomous robot is often in a situation to perform tasks or missions in an initially unknown environment. A logical approach to doing this implies discovering the environment by the incremental principle defined by the applied exploration strategy. A large number of exploration strategies apply the technique of selecting the next robot position between candidate locations on the frontier between the unknown and the known parts of the environment using the function that combines different criteria. The exploration strategies based on Multi-Criteria Decision-Making (MCDM) using the standard SAW, COPRAS and TOPSIS methods are presented in the paper. Their performances are evaluated in terms of the analysis and comparison of the influence that each one of them has on the efficiency of exploration in environments with a different risk level of a "bad choice" in the selection of the next robot position. The simulation results show that, due to its characteristics related to the intention to minimize risk, the application of TOPSIS can provide a good exploration strategy in environments with a high level of considered risk. No significant difference is found in the application of the analyzed MCDM methods in the exploration of environments with a low level of considered risk. Also, the results confirm that MCDM-based exploration strategies achieve better results than strategies when only one criterion is used, regardless of the characteristics of the environment. The famous D* Lite algorithm is used for path planning.

Keywords: Exploration strategies; multi-criteria decision-making; SAW; COPRAS; TOPSIS

1 Introduction

The autonomous exploration of unknown environments is one of the most important tasks in mobile robotics. It can be defined as a set of actions taken by an autonomous mobile robot or by multiple robots in line with a strategy in order to discover unknown features in environments. Hence, exploration is the basis of numerous real-world applications of robotics, such as mapping [1,2], search and rescue [3], planetary missions [4], visual inspections [5], mining [6], and so on. For instance, in mapping the features to be explored, there are a free space and obstacles, whereas in search and rescue missions, it can be the areas where the victims of a disaster or fires are located.

Exploration can be broadly classified into two distinct approaches [7]. The first approach involves a prior knowledge of the environment, based on which off-line algorithms are used to define the exploration strategy. In this approach, the path of the robot is determined in advance (a predefined path). The second approach is applied when the environment is completely unknown or when there is insufficient information to effectively implement an off-line algorithm. In this case, exploration is much more challenging, usually implying taking incremental steps to realize it, where in each step a robot has to choose its next location, move to it and make a new observation of the environment at that particular location. The main problem of this exploration concept is the choice of the next location of the robot. The strategies implemented by this approach are often called Next-Best-View (NBV) exploration strategies. In the exploration strategies, the next location is usually selected among a number of the candidate locations usually placed on the frontiers between the explored free space and the unexplored parts of the environment, evaluating them according to certain criteria [8,9]. Yamauchi [10] was the first to propose frontier-based approach in exploration. Other techniques may imply that the choice of the next location is made by means of a random method, a human-directed method, and so forth. Each exploration strategy is aimed at ensuring that the environment is explored as quickly as possible. Here, it is important to notice the difference between the terms "exploration" and "coverage", which are often associated with each other. In the case of coverage, the environment map is completely known, and the aim is to generate the path that enables the robot to cover the whole environment with some tools (a sensor, a brush, etc.). Generally speaking, the goal of exploration and coverage can be the same—to search certain items of interest, but the exploration problem is harder to solve, because the robot first has to map an unknown environment.

In this paper, the frontier-based exploration strategies based on Multi-Criteria Decision-Making (MCDM) are presented. Having in mind that MCDM methods are the state of the art methods for the decision-making problems, the motivation of the authors is to discuss their integration with autonomous exploration. So far, MCDM has been applied in the exploration process in order to make an optimal choice of the next robot location. The general characteristics of the MCDM-based exploration strategies are the following: MCDM provides a candidate selection based on the aggregation of the different criteria (which are some-times conflicting) that affect the quality of a decision and enable an easy addition of new criteria. The advantages of the MCDM-based strategies over other approaches in exploration can be found in [7,9]. This approach to exploration, however, has not sufficiently been explored given the diversity of MCDM methods and their increasing application in various scientific disciplines. The main contributions and novelties of this paper are that it expands the MCDM base of exploration strategies with three standard MCDM methods—SAW, COPRAS and TOPSIS (which, to the knowledge of the authors, have not been used for this purpose so far). The main idea is to expand the comparison of the MCDM-based exploration strategies to the approach when only one criterion is used and analyze the influence of such different MCDM methods on the efficiency of exploration in environments with a high risk of a bad choice (primarily in terms of an information gain) in the selection of the next robot position. Bearing in mind the fact that the TOPSIS method is based on the idea that the optimal alternative should have the smallest distance (in a geometric sense) from the ideal solution and simultaneously the largest distance from the negative ideal (anti-ideal) solution, the assumption is that this method will have better results than the other considered methods in this situation.

Besides, the powerful D* Lite algorithm is used to calculate the shortest path to each candidate (one of the criteria in MCDM) in a complex environment with many obstacles and plan the robot path to the selected candidate. In addition to the shortest path from the current robot position to the candidate calculated by the planner, the criteria considered for selecting candidates are an information gain estimated for each candidate, as well as its distance from the base station (or the probability of communication).

The rest of this paper is organized as follows. Section 2 presents an overview of the related work. Section 3 deals with the detection of the frontiers, candidate selection, the D* Lite algorithm and the implemented MCDM methods. Simulation, evaluation and discussion are given in Section 4. The conclusions are presented in Section 5.

2 Related Work

The frontier-based approach is still one of the main directions of the research in autonomous exploration in robotics. In [11] the exploration planner combining the local and global approaches to exploration is presented. It is one of the methods for the detailed exploration of large unknown environments, which simultaneously tries to take advantages and minimize the disadvantages of both the local and the global approaches to exploration. An interesting frontier-based autonomous exploration algorithm for aerial robots that shows some improvements which is to be used in search and rescue missions is proposed in [12]. There, a RGB-D cameras are used to sense the environment and an OctoMap is used for expression the obstacle, unknown and free space in the environment. Then, a algorithm for clustering is used to filter the frontiers extracted from the OctoMap and the information-based cost function is applied in order to choose the optimal frontier. Finally, the feasible path is given by the A* algorithm and the safe corridor generation algorithm. In [13], a heading-informed frontier-based exploration approach is proposed and tested. Here, an additional rotation cost is added to the utility function for the selection of frontiers, which enables the agent to maintain its orientation during the exploration.

Frontier-based exploration can also successfully be applied to multi-robot systems. In this regard, various unknown environment exploration techniques using autonomous multi-robot teams are proposed. For example, in [14], the multi-robot frontier exploration uses available information about the scene in the decision process in order to obtain a higher reward of the areas of a certain type and separate robots' trajectories. In [15], a method for the coordination of the robot team for unknown scenario exploration which may include moving obstacles is presented. This method is based on the separation of the map into zones, so that each robot explores a different zone. In [16], a total of five multi-robot frontier-based exploration strategies were applied to test how the efficiency of an exploration strategy depends on the environment where it is tested in.

To evaluate the candidate r in order to select the next robot position in frontier-based exploration strategies, different criteria are proposed in the literature. The following criteria are most commonly used [8,9]:

- $L(r)$, the minimum length of a collision free path or the minimum path cost from the current robot position to r usually calculated by using a path planner,
- $A(r)$, the expected information gain obtained by simulating the robot's perception from the location r; it is calculated or estimated based on the size of the unexplored area that would be explored from that location, given the current status of the occupancy grid map and the robot's sensing range,
- $P(r)$, the probability that the robot, if it arrives at the location r, will be able to communicate with the base station and send the collected data; this probability generally directly depends on the Euclidean distance of r from the base station.

There are various approaches using one of the above-mentioned criteria or a larger number of them in the form of the function that uniquely describes each candidate r.

In [17], $L(r)$ is combined with $A(r)$ in the form of the linear function (1):

$$u(r) = A(r) - \beta L(r) \tag{1}$$

The parameter β regulates the influence of the criterion $L(r)$ *vs*. $A(r)$.

In [18], the presented approach describes the candidate r as the following exponential function (2):

$$u(r) = A(r) \cdot \exp(-\lambda L(r)) \tag{2}$$

The parameter λ is greater than zero and weights both included criteria. This strategy is named after the authors as the GBL strategy [9].

These exploration approaches are mainly used for the map building process. In this process and especially in the case of search and rescue missions, however, it is usually important to introduce a criterion related to the probability of establishing communication between the robot from the location r and the base station ($P(r)$), so that the information can be forwarded as soon as possible to be further used. The introduction of this criterion was, for the first time, explicitly considered in paper [19] in the form of the following function (3):

$$u(r) = \frac{A(r)P(r)}{L(r)} \tag{3}$$

On the other hand, different additional criteria for the selection of the next robot position are proposed in the literature. For example, in [20], the overlap (*O(r)*) of the current environment map and the part of the environment visible from r is proposed as an additional criterion. In addition to the path cost, the criteria such as the recognition of the uncertainty of a landmark, the number of the features visible from the location, the length of the visible free edges, and the number of the rotations and stops required from the robot to reach a location are considered in [1]. Criteria selection depends on the mission specifics and the exploration goals. In this context, the introduction of the criterion that (if possible) will take into account the types of facilities for each candidate location is proposed in the paper [9] (which considers exploration in search and rescue missions) in order to initially direct a mission to the residential area, where the largest number of the victims of a disaster are logically expected.

Taking into account the above-mentioned, a more recent direction of research in this area includes the implementation of different MCDM methods in MCDM-based exploration strategies. MCDM provides a broad and flexible approach to the selection of the utility function that can be used to evaluate candidates for the next observation location. In [7,8], for example, the Choquet fuzzy integral is proposed. This approach enables a researcher to take into account the relative relationship between criteria, such as redundancy and synergy, which is its main characteristic. The experimental results in those papers show that respectable results are obtained by using MCDM-based exploration strategies compared to the other exploration strategies. In [9], the proposed approach to the selection of the next location from a set of candidates within the exploration strategy uses a standard MCDM method—PROMETHEE II. Here, an attempt is made to take advantage of the characteristic of this method referring to the fact that, in addition to weights, preference functions are used as additional information for each criterion. Autonomous robot navigation strategy based on MCDM Additive Ratio ASsessment (ARAS) method is proposed in [21]. The greedy area exploration approach is suggested, while the criteria list consists of: battery consumption rate, probability to collide with other objects, probability to yaw from course, probability to drive through doors, probability to gain new information, length of the minimum collision-free path. In [22], the implementation of the on-line MCDM-based exploration strategy that exploits the inaccurate knowledge of the environment (information obtainable from a floor plan) is proposed. The results of the experiment show that the proposed approach has a better performance in different types of environments with respect to the strategy without prior information. Although the use of more accurate prior information leads to a significant improvement in performance, the use of inaccurate prior information could lead to certain advantages also, managing to reach the high percentages of the explored area travelling a shorter distance, with respect to the strategy not using any prior information. In [23], behavior-based and MCDM methods for autonomous exploration in an environment after a nuclear disaster are presented. These approaches are developed for the mobile robots that use gamma-based camera for discovering radioactive hotspots. As the limitations of the gamma-based cameras (a long acquisition time, a poor angular resolution, etc.) are the main challenges for considered autonomous exploration method (because they degrade the duration and accuracy of the exploration), algorithms are adapted so as to overcome these limitations. Based on the presented results, it was concluded that MCDM is a more effective approach for the exploration and mapping of radioactive sources than the behavior-based method.

This paper generally belongs to the above-mentioned direction of autonomous exploration research, bearing in mind the fact that this issue is not sufficiently considered in the existing literature. It expands the base of the MCDM methods implemented in exploration strategies and compares them with each other, trying to contribute to overcoming research gap in that manner.

Worthy of mention is the fact that the other important directions of the research done in the field of autonomous exploration in robotics include the study of different information gain prediction techniques in selecting the next robot location regardless of the strategy, bearing in mind important role of this criterion. In that context, [24] proposes that a neural network should be implemented in order to perform the prediction of unexplored areas during the exploration process. Assuming that, at the beginning of the process, certain environmental data are known in the form of topometric graphs, exploration efficiency can be increased as is shown in [25]. These graphs can automatically be generated based on the existing plans or maps of the environment or hand-drawn by the people familiar with the environment. The approach to exploration where the selection of the next location is made based on the direct observation of the robot and the prediction of the suitability of not-yet-visited locations by using Gaussian processes is proposed in [26]. In [27,28] the introduction of heuristics for an additional description of the next best-view location for an autonomous agent is proposed in order to enhance the unknown environment exploration process.

Finally, it should be noted that a path planner plays an essential role in autonomous exploration. In this sense, graph-based algorithms are often used as reliable planners in order to calculate the minimum path cost criterion from the robot current position to the candidate, as well as to plan the robot path to the selected position in the exploration process. The most commonly used algorithms are $A*$ [12,16,29,30], Dijkstra [11,31–33] and $D*$ [34], as well as $RRT*$ [35]. As an advanced version of the graph-based algorithm and a variation of $D*$ with a lot of practical implementations, the $D*$ Lite algorithm used in this paper is also suitable for being implemented in exploration.

3 MCDM-Based Exploration Strategies Using SAW, TOPSIS and COPRAS

The $D*$ Lite algorithm is proposed as the path planner in this paper, given its advanced planning features in complex and dynamic environments. Besides, the important roles in MCDM-based exploration strategies have a built-in technique for defining a set of candidates for the next robot position, as well as the applied MCDM method.

3.1 Frontiers and Candidate Detection

The frontier determination techniques used in frontier-based exploration may be different. One of the first approaches is described in the papers [10,36], generally referring to the environment representation in the form of an occupancy grid map. In this approach, if a cell belonging to the explored part of the environment is adjacent to a cell belonging to the unexplored part of the environment, then the cell is called a frontier edge cell. By grouping adjacent frontier edge cells, larger entities are obtained—the so-called frontier regions. Usually, a threshold is defined for the frontier region in terms of the minimum size—when that threshold is reached, it is called a frontier. If the robot moves in accordance with a strategy from one frontier to another, then it has the ability to constantly upgrade its knowledge of the environment. This process is called frontier-based exploration. The frontier determination approaches based on the above-mentioned can be found explained in more detail in the papers [7,8,37].

The main steps of frontier-based exploration are presented in Fig. 1 and can be defined as follows [38]:

1. The selection of the next observation location according to an exploration strategy;
2. The reaching of the observation location selected in previous step. This step requires the planning and following a path, that goes from the robot's current position to the chosen location;
3. The acquisition of a partial map from the observation location, using data collected by the robot's sensors;
4. The integration of the partial map within the global map.

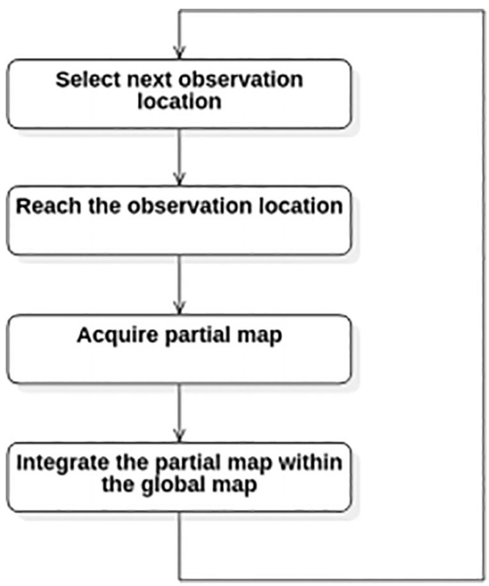


Figure 1: The main steps of the exploration process

A similar approach is applied in this paper (Fig. 2). The boundary of the robot's field of view defined by the range of its sensors can be divided into free, obstacle and frontier arcs. Free arcs are parts of the border to the explored part of the environment, whereas obstacle arcs are parts of the border to locally detected obstacles. Any arc that is neither obstacle nor free is a frontier arc and is, in fact, part of the border to the unexplored part of the environment. The approach that each frontier arc is actually one frontier can be adopted.

Allow us also to take that the middle point of each frontier represents one candidate for the next robot position, which is a common approach in the papers dealing with this topic [7,8,39]. The criteria $L(r)$, $A(r)$ and $P(r)$ described in Section 2 are used for candidate selection. The $D*$ Lite algorithm is used to calculate the $L(r)$ and plan the robot path to the selected candidate as well.

When the robot reaches the selected location, it observes the surroundings, improves the knowledge of the environment and also updates the frontier list [18,37].

3.2 D Lite Algorithm*

$D*$ Lite [40] is a broadly used incremental path planning algorithm. It is a variation of the $D*$ algorithm. The $D*$ Lite is at least as efficient as $D*$, only differing from it in the algorithmic procedure and in being simpler in that sense [41]. Both $D*$ and $D*$ Lite are an extension of the more famous $A*$ algorithm [42]. $D*$ Lite works by calculating an initial path with a minimum total cost principally in a similar way as a backward version (search is done from the goal to the start) of $A*$. If changes in the environment are identified, then $D*$ Lite executes the replanning process so as to test the existing path and, if needed, corrects it or generates a new one with the maximum possible exploitation of the calculation result from previous iterations. That is, if a graph change is detected, $D*$ Lite does not reset the calculation process (i.e., it does not return the search to the beginning as it is the case in $A*$), which makes $D*$ Lite remarkably more efficient than the basic $A*$ planner [43].

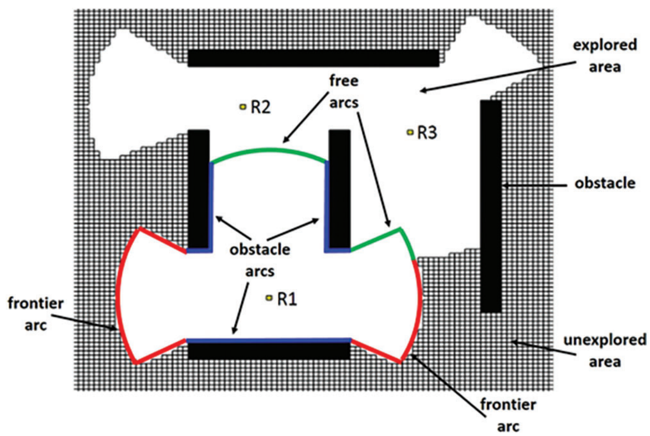


Figure 2: Free, frontier and obstacle arcs in exploration

During the path search, $D*$ Lite initializes and updates the value of the parameters that characterize the cell s [44], as follows:

- $g(s)$, the minimum cost of transition from s_{goal} to s found so far;
- $h(s)$ (or heuristic value), estimates the minimum cost of transition from s to s_{start}. Using heuristic provides that the search tree is oriented toward the most optimistic cells regarding their belonging to the most favorable path from the start cell to the goal cell (which speeds up the search itself);
- $f(s) = g(s) + h(s)$, estimates the minimum cost of transition from s_{start} via s to s_{goal};
- $rhs(s) = min_{s' \in Succ(s)}(c(s, s') + g(s'))$ or $rhs(s) = 0$ if $s = s_{goal}$, the one-step lookahead estimate of s and expresses the path cost defined by analyzing its neighbors' g parameters. In application, each cell keeps the pointer to the cell from which it provides its rhs parameter, so the agent should follow the line of pointers from its current cell in order to pursue the generated path to the goal.

The pseudocode of $D*$ Lite is presented in Tab. 1 [45] (Algorithm 1), and its main steps are analyzed below. During the operation of $D*$ Lite, the cells are classified as consistent (if $g(s) = rhs(s)$) or inconsistent (in all other cases). The cells considered as inconsistent can be overconsistent (if $g(s) > rhs(s)$) or underconsistent (if $g(s) < rhs(s)$). In a fashion similar to the other planners from the $A*$ and $D*$ families, $D*$ Lite initializes and updates a set of inconsistent cells (the *OPEN* list) so as to execute an efficient expansion of these cells and at the same time to expand a search tree. Cell expansion is based on the *key* value as in line 8, which is calculated as in line 1. Initialization of the $D*$ Lite algorithm implies determination the *goal* cell parameters so that it is classified as inconsistent and is inserted into the *OPEN* list (lines 15–17). The algorithm then generates the minimum cost path by activating the ComputePath( ) function (line 19). The agent moves along the calculated path (s_{start} is also updated). If certain change occurs in the environment during the robot's movement within the range of its sensor, the planner updates the *rhs* of all the cells affected directly by this change and inserts those cells that become inconsistent in the *OPEN* list (lines 20–23). After that, it executes the path checking-replanning process by recalling the ComputePath( ) function.

Table 1: $D*$ Lite pseudocode

Algorithm 1: The D* Lite algorithm (the basic version)
key(s)
01. return [min($g(s)$, $rhs(s)$) + $h(s_{start}, s)$; min($g(s)$, $rhs(s)$)];
UpdateCell(s)
02. if s was not visited before
03. $g(s) = \infty$;
04. if ($s \neq s_{goal}$) $rhs(s) = \min_{s' \in Succ(s)}(c(s, s') + g(s'))$;
05. if ($s \in OPEN$) remove s from *OPEN*;
06. if ($g(s) \neq rhs(s)$) insert s into *OPEN* with $key(s)$;
ComputePath()
07.while ($\min_{s \in OPEN}(key(s)) < key(s_{start})$ OR $rhs(s_{start}) \neq g(s_{start})$)
08.   remove the cell s with the minimum *key* from *OPEN*;
09. if ($g(s) > rhs(s)$)
10. $g(s) = rhs(s)$;
11.   for all $s' \in Pred(s)$ UpdateCell(s');
12. else
13. $g(s) = \infty$;
14.   for all $s' \in Pred(s) \cup \{s\}$ UpdateCell(s');
Main()
15. $g(s_{start}) = rhs(s_{start}) = \infty$; $g(s_{goal}) = \infty$;
16. $rhs(s_{goal}) = 0$; $OPEN = \emptyset$;
17. insert s_{goal} into *OPEN* with $key(s_{goal})$;
18. forever
19. ComputePath();
20.   wait for changes in the edge costs;
21.   for all the directed edges (u, v) with changed edge costs
22.   Update the edge cost $c(u, v)$;
23.   UpdateCell(u);

Line 18 actually denotes that, in a practical implementation, the algorithm ends its work when $s_{start} = s_{goal}$ (when the robot reaches the goal cell) [46]. If edge costs are equal to the dimensions (lengths) of the corresponding transitions, then $D*$ Lite calculates the least-cost path, which is also the shortest path in the graph representation of the environment.

3.3 Some Properties of TOPSIS, COPRAS and SAW

The Technique for Ordering Preference by Similarity to Ideal Solution (TOPSIS) method introduces the ranking index that includes distances from the ideal solution and the anti-ideal solution [47]. The ideal solution minimizes the cost type criteria and maximizes the benefit type criteria, while the opposite is only true for the anti-ideal solution. The TOPSIS method is based on the idea that, in a geometric sense, the optimal alternative should have the smallest distance from the ideal solution and simultaneously the largest distance from the anti-ideal solution. The TOPSIS method is characterized by the following advantages [48,49]: 1) it has a simple mathematical formulation; 2) it respects the ideal solution and the anti-ideal solution when defining the final criterion function; 3) it provides a well-structured analytical framework for ranking alternatives; 4) the algorithm is not complicated regardless of the number of criteria and alternatives; 5) it is one of the best methods for solving rank reversal problems. This method is especially suitable when there is an intention to avoid risk, because the maximum possible benefit and the maximum possible risk avoidance are often equally important to the decision-maker. The normalization of values in decision matrix in the TOPSIS method is carried out by using vector normalization, as in expression (4):

$$x_{ij} = \frac{r_{ij}}{\sqrt{\sum_{i=1}^{m} r_{ij}^2}} \tag{4}$$

where r_{ij} represents the value of the alternative i for the criterion j, m represents the total number of alternatives, while x_{ij} represents the normalized value of the alternative i within the criterion j. Multiplying the normalized values x_{ij} by criteria weights gives the elements v_{ij} of the weighted normalized decision matrix [50]. The ideal solution and the anti-ideal solution are defined by applying expressions (5) and (6):

$$\begin{aligned} A^* &= \left\{(\max v_{ij} \,|\, j \in G), (\min v_{ij}, j \in G'), \, i = 1, .., n\right\} \\ &= \left\{v_1^*, \, v_2^*, \, \ldots, \, v_m^*\right\} \end{aligned} \tag{5}$$

$$\begin{aligned} A^- &= \left\{(\min v_{ij} \,|\, j \in G), (\max v_{ij}, j \in G'), \, i = 1, .., n\right\} \\ &= \left\{v_1^-, \, v_2^-, \, \ldots, \, v_m^-\right\} \end{aligned} \tag{6}$$

A^* indicates the best alternative (the ideal solution), whereas A^- indicates the anti-ideal solution following the same logic [51]. Based on the defined absolute distances of the alternatives from the ideal solution and the anti-ideal solution, the relative distance of the alternatives to the ideal solution is defined and the final rank of the alternatives is determined [52]. The best alternative is that with the highest value of the criterion function.

The Compressed Proportional Assessment (COPRAS) method [53,54] has a somewhat more complex criterion function value aggregation process. The COPRAS method, however, is characterized by a simplified data normalization procedure, since data normalization performed by this method requires no criterion transformation depending on the qualification of the benefit type or the cost type, which eliminates the danger of the data structure disruption in the value normalization process, as is the case with the largest number of multi-criteria models. The COPRAS method also provides a well-structured analytical framework for ranking alternatives [54]. While the TOPSIS belongs to the so-called distance-based MCDM methods, the COPRAS method is classified as a scoring MCDM method. It uses the procedure for the stepwise ranking of alternatives and the procedure for the evaluation of alternatives in terms of their significance and utility degree. In the COPRAS method, criterion value normalization is carried out by applying additive normalization, as in expression (7):

$$x_{ij} = \frac{r_{ij}}{\sum_{i=1}^{m} r_{ij}} \tag{7}$$

where r_{ij} represents the value of the alternative i for the criterion j, m represents the total number of alternatives, while x_{ij} represents the normalized value of the alternative i within the criterion j. After the normalization of values, the aggregate values are defined within the benefit set and the cost set [55]. By applying the expression (8), the significance (influence) of each of the alternatives from the set is determined and the final rank of the alternatives is defined:

$$Q_i = S_i^+ + \frac{S_{\min}^- \sum_{i=1}^{m} S_i^-}{S_i^- \sum_{i=1}^{m} \left(\frac{S_{\min}^-}{S_i^-} \right)} = S_i^+ + \frac{\sum_{i=1}^{m} S_i^-}{S_i^- \sum_{i=1}^{m} \frac{1}{S_i^-}} \tag{8}$$

where S_i^+ and S_i^- represent the aggregated criterion values within the benefit set and the cost set, respectively.

The Simple Additive Weighting (SAW) method [47,56] belongs also to the group of the scoring multi-criteria methodologies. Its main advantage reflects in the simple aggregation function. For each alternative a cumulative characteristic is calculated, representing the sum of the weighted normalized values by all the criteria. The alternative that corresponds to the largest value calculated in this way is the best solution. In addition to the fact that SAW provides a simple procedure for ranking alternatives, the results obtained by its application generally do not deviate from the results obtained by more advanced methods. In sum, the following advantages of the SAW method can be underlined: 1) data normalization is not mandatory; 2) it has a simple mathematical apparatus; 3) it has a well-structured analytical framework for ranking alternatives; 4) the algorithm is not complicated regardless of the number of criteria and alternatives. One of the main disadvantages of this method is that it can only be applied directly if all the criteria are maximizing, while the minimizing criteria must first be converted to maximizing ones. After defining the weight coefficients of criteria [57,58], the ranking of alternatives is performed by applying the aggregation function (9):

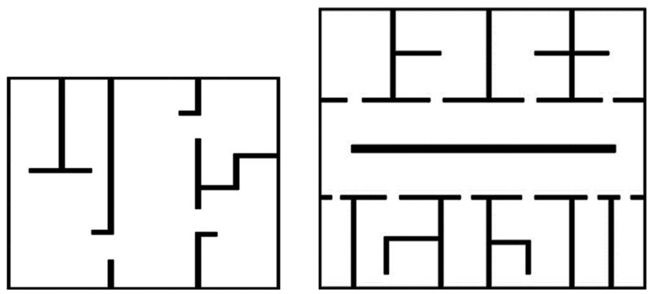
$$Q_i = \sum_{j=1}^{n} x_{ij} \cdot w_j \tag{9}$$

where w_j represents the weight coefficient of the criterion j, n represents the total number of the criteria, while x_{ij} represents the normalized (simple linear normalization) value of the alternative i within the criterion j.

In addition to the previously mentioned characteristics of used MCMD techniques, below are details that additionally motivated the authors to use the TOPSIS, COPRAS and SAW methods in this study. The SAW is probably the best known and most commonly used MCDM method, because of its simplicity, but also because it provides quality solutions [47]. The COPRAS method is one of newer methods which is increasingly used in literature because of its several advantages (less computational time, very simple and transparent, etc.) over other MCDM methods [50]. COPRAS is essentially an evolution of SAW so it also provides quality solutions. TOPSIS on the other hand has the specific property of choosing the optimal alternative not only on the basis of considering the ideal, but also the anti-ideal alternative. It provides the choice of the alternative that is both closest to the ideal and farthest from the anti-ideal alternative and therefore tends to minimize risk in decision making. These specificities were the motivation to experiment with the general implementation of SAW, COPRAS and TOPSIS in MCDM-based exploration strategies, as well as to additionally research the potential advantage of applying TOPSIS for exploration in complex environments with a high level of risk of a bad choice in selection of the next robot position. However, the authors in future research plan to experiment with other MCDM techniques, in order to define the optimal tools for application in autonomous exploration strategies in different specific situations.

4 Simulation and Results

To evaluate the implementation of different MCDM methods in exploration strategies, the autonomous exploration of the unknown complex environments was simulated in Matlab. In particular, an exploring robot was engaged in the environment A and the environment B, as is shown in Fig. 3.



(a) (b)

Figure 3: The environments A and B used for the test. (a) The environment A (100 × 100) (b) The environment B (150 × 150)

These two environments have some different characteristics. The environment B contains the corridors and a lot of rooms of a similar size, whereas the main specificity of the environment A is that it contains corridor without obstacles and one room significantly larger than the others. The common specificity of the environments reflects in the fact that they both have a high risk of a bad choice in the selection of the next robot position (primarily in terms of an information gain), noting that the level of this risk is different.

Exploration effectiveness also depends on the robot starting location to some extent. The results presented in [7] show that, if there are fewer obstacles in an environment, then the influence of the choice of the starting location on the environment exploration efficiency is smaller. Therefore, a total of four different starting locations were processed for each given environment (one from the central area on each side of the environment), different initial exploration conditions being tested. At each observation location, the robot performed a 360° scan of the environment, within the maximum sensor range (R = 15). The exploration ended when 90% of the environment was covered. The remaining 10% was mainly composed of the room corners related to the less significant features of the environment, not significantly affecting the comparison of the strategies [7,9].

The weights of criteria $L(r)$ and $A(r)$ were changed for each MCDM method in both environments with an increment of 0.1 (the weight of the criterion $P(r)$ was constant −0.1), keeping in mind that the sum of all weights is equal to one, in order to find the combination that would give the best results. Defining a higher weight for the criterion $A(r)$ compared to the criterion $L(r)$ caused the robot to strive to choose the location that gave it a greater benefit in terms of the size of the expected information gain, although the path it had to pass in order to go to that location was longer than that to the competitive locations [8]. Detailed considerations regarding the choice of criteria weights can be found in [8,9]. All the MCDM methods tested in this paper had the best result with the combination of the weights 0.6, 0.3, 0.1 (MCDM-based Strategy 1) in the environment A, and with the combination of the weights 0.7, 0.2, 0.1 (MCDM-based Strategy 2) in the environment B, for the criteria $L(r)$, $A(r)$ and $P(r)$, respectively. In order to test different approaches, in addition to MCDM-based strategies we tested Dist_Min and GBL strategy. Dist_Min selects the candidate that minimizes criterion $L(r)$, while GBL selects the candidate that maximizes the value of expression (2), where $\lambda = 0.2$ (the same value reported in the original paper [18]). The performance of the exploration for the starting location (in the Cartesian coordinate system) $x = 50$, $y = 99$ and MCDM-based Strategy 1 (by using TOPSIS, COPRAS and SAW), Dist_Min and GBL in the environment A, as well as for the starting location $x = 70$, $y = 149$ and MCDM-based Strategy 2 (by using TOPSIS, COPRAS and SAW), Dist_Min and GBL in the environment B are given in Fig. 4.

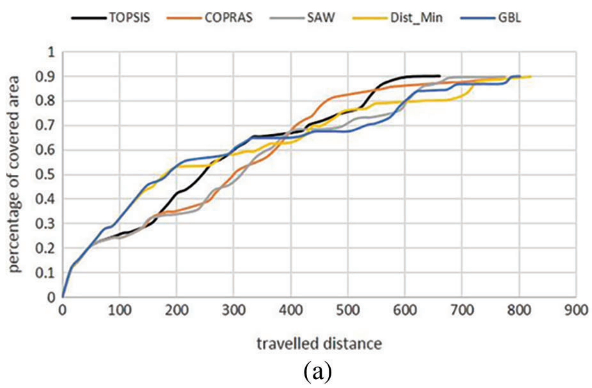


Figure 4: The performance of the tested exploration strategies. (a) The environment A (b) The environment B

The exploration results in terms of the average travelled distance for the four above-mentioned starting locations as per environment are given in Tab. 2.

Table 2: The average exploration results in terms of the travelled distances for the environments A and B

Exploration strategies		The travelled distance	
		Environment A	Environment B
MCDM-based	TOPSIS	785.83	1679.26
	COPRAS	884.80	2176.37
	SAW	808.42	2012.74
Dist_Min		896.59	2426.94
GBL		889.87	2223.81

The risk level of a bad choice in the selection of the next robot position relates to the complexity of the environment, primarily in terms of the present obstacles, their shape and dimensions, and so on. The mentioned risk is particularly significantly higher in the case when the environment contains the obstacles whose dimensions are larger than the robot's sensor range. In the opposite case, the level of considered risk is lower. Accordingly, the MCDM-based exploration Strategy 1 by using SAW, COPRAS and TOPSIS, as well as Dist_Min and GBL were additionaly tested in a simple environment with a low risk of a bad choice in the selection of the next robot position (Fig. 5).

The exploration results in terms of the average travelled distance for the four different starting locations (one from the central area on each side of the environment C) for this environment are given in Tab. 3.

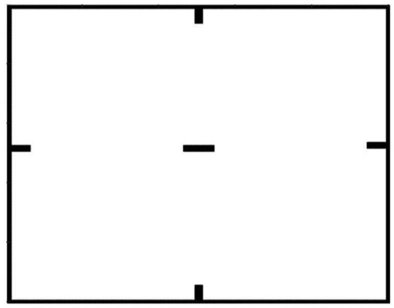


Figure 5: The environment C (100 × 100) used for the test

Table 3: The average exploration results in terms of the travelled distances for the environment C

Exploration strategies		The travelled distance
		Environment C
MCDM-based	TOPSIS	434.48
	COPRAS	423.86
	SAW	444.59
Dist_Min		579.52
GBL		476.66

The exploration results of the MCDM-based strategy using the TOPSIS method and the corresponding robot paths generated by the *D** Lite algorithm for the starting locations considered in Fig. 4 in the environments A ($x = 50$, $y = 99$) and B ($x = 70$, $y = 149$) are shown in Figs. 6 and 7, respectively.

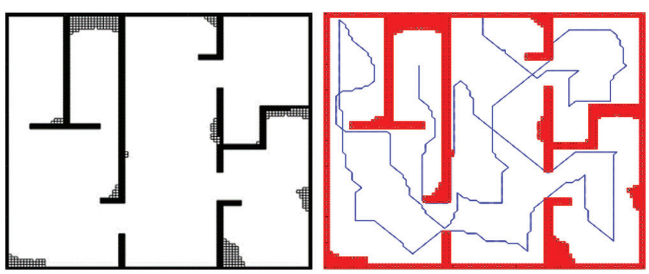


Figure 6: The covered area and the path generated by the *D** Lite algorithm in the environment A with the implemented MCDM-based exploration Strategy 1 (by using TOPSIS)

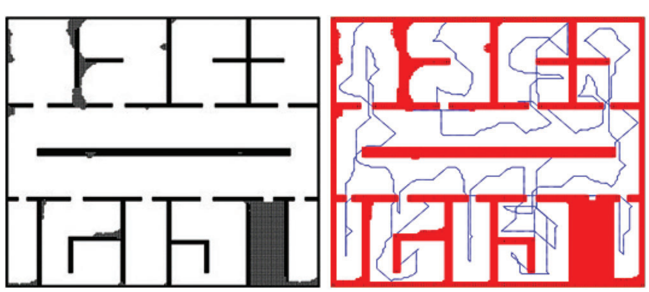


Figure 7: The covered area and the path generated by the *D** Lite algorithm in the environment B with the implemented MCDM-based exploration Strategy 2 (by using TOPSIS)

5 Discussion

The results presented in Tab. 2 show that it is reasonable to apply a MCDM-based strategy in order to improve the efficiency of exploration, rather than the other tested techniques. Comparing the three applied MCDM methods, the TOPSIS has the best exploration results in both environments, which means that TOPSIS leads the robot to travel the shortest distance in order to explore 90% of the area (a distance shorter 11.22%, 2.83%, 12.39% and 11.74% than that by using COPRAS, SAW, Dist_Min and GBL respectively in the environment A, and a distance shorter 22.84%, 16.57%, 30.81% and 24.49% than that by using COPRAS, SAW, Dist_Min and GBL respectively in the environment B). These results have a theoretical basis, taking into account the common specificity of the environments A and B (a high risk of a bad choice in the selection of the next robot position) and the characteristic of the TOPSIS method related to the intention to minimize the risk in decision-making as well. On the other hand, SAW and COPRAS are the classical aggregation methods that do not consider the "distances" between the ideal solution and the anti-ideal solution, but are also broadly applied in MCDM irrespective of that fact. The results presented in Tab. 3 show that, in the scenarios with a low risk of a bad choice in the selection of the next robot position, all the three analyzed MCDM methods can be expected to achieve similar or comparable exploration results.

Analyzed from the aspect of combinations of criteria weights, the MCDM-based Strategy 1 was more successful than the MCDM-based Strategy 2 in the environment A because it gave more importance to the criterion of the expected information gain, bearing in mind the fact that it can bring significant exploration benefits. This approach comes to the fore in such situations like the environment A, which contains a larger open spaces without obstacles. On the other hand, this approach can lead to poorer results compared to the approach with the lower weights of the criterion $A(r)$ if the initially assumed open space is in fact blocked [8]. Therefore, if the environment is such that a significantly greater benefit according to the expected information gain cannot be expected with certainty, then the "aggressive" strategy forcing the robot to travel greater distances in order to explore more space does not give better results than the less "aggressive" strategy that gives less importance to this criterion, simultaneously increasing the significance of the travelled distance.

The main feature of the MCDM-based exploration strategies, that is confirmed in this paper, implies that these strategies allow flexibility in selecting/adding the criteria that drive the exploration process depending on a specific situation and the objectives of the mission [7,8]. Besides, MCDM-based exploration strategies enable flexibility in defining criteria weights, taking into account the information about the environment available at the beginning of the exploration, as well as the data collated during that process (bearing in mind the fact that, in a real situation, it is difficult to define in advance the combination of weights that will provide the best exploration results). In other words, the MCDM-based strategy allows the application of and change in various behaviors which the exploration will be carried out. If there is a set of defined criteria, changing the value of their weights during the exploration process [8] may lead to transition from one behavior to another.

6 Conclusion

The application of MCDM-based exploration strategies by using the SAW, COPRAS and TOPSIS methods is presented in this paper, together with the analysis of their performances in the complex environments with a different level of the risk of a bad choice in the selection of the next robot position. The best exploration results in the environments with a high risk of a bad choice in the selection of the next robot position were obtained by using TOPSIS. These results are theoretically based, bearing in mind the fact that the TOPSIS method enables the approach in which the maximum possible benefit and the maximum possible risk avoidance are treated as equally important. In the environments with a low level of a considered risk, all the analyzed MCDM methods achieved either similar or comparable exploration results. In both cases, the MCDM-based exploration strategies achieve better results than other tested techniques. The D* Lite algorithm is used to calculate the shortest path to each candidate (one of the criteria in MCDM), as well as to plan the robot path to the selected candidate.

A limitation of this concept may be the impossibility of processing uncertain input information. This limitation will be considered in the future research that will include the application of fuzzy theory, rough theory or neutrosophic theory in order to process the uncertainties that represent criterion values. In order to improve the exploration process, especially in search and rescue missions, the integration of probabilistic models to increase the reliability of path planning [59] will be considered. Potential future research will also relate to the automation of the criteria weights adjustment process in MCDM-based exploration strategies, based on the experience and learning algorithms, as well as to include soft set theory in MCDM [60].

Acknowledgement: The authors of this paper would like to express their gratitude to the Ministry of Education, Science and Technological Development of the Republic of Serbia for their support of Contract No. 451-03-68/2020-14/200325.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] B. Tovar, L. Munoz-Gomez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy *et al.,* "Planning exploration strategies for simultaneous localization and mapping," *Robotics and Autonomous Systems*, vol. 54, no. 4, pp. 314–331, 2006.

[2] F. Amigoni, V. Caglioti and U. Galtarossa, "A mobile robot mapping system with an information-based exploration strategy," in *Proc. of the Int. Conf. on Informatics in Control, Automation and Robotics*, Setubal, Portugal, pp. 71–78, 2004.

[3] S. Kohlbrecher, J. Meyer, T. Graber, K. Petersen, O. von Stryk *et al.,* "Hector open source modules for autonomous mapping and navigation with rescue robots," in *RoboCup: Robot World Cup XVII, ser. Lecture Notes in Artificial Intelligence (LNAI)*. Berlin: Springer, pp. 624–631, 2013.

[4] K. Otsu, A.-A. Agha-Mohammadi and M. Paton, "Where to look? Predictive perception with applications to planetary exploration," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 635–642, 2018.

[5] A. Ahrary, A. A. Nassiraei and M. Ishikawa, "A study of an autonomous mobile robot for a sewer inspection system," *Artificial Life and Robotics*, vol. 11, no. 1, pp. 23–27, 2007.

[6] T. Neumann, A. Ferrein, S. Kallweit and I. Scholl, "Towards a mobile mapping robot for underground mines," in *Proc. of the RobMech*, Cape Town, RSA, pp. 279–284, 2014.

[7] N. Basilico and F. Amigoni, "Exploration strategies based on multi-criteria decision making for an autonomous mobile robot," in *Proc. of the 4th European Conf. on Mobile Robots*, Mlini/Dubrovnik, Croatia, pp. 259–264, 2009.

[8] N. Basilico and F. Amigoni, "Exploration strategies based on multi-criteria decision making for search and rescue autonomous robots," in *Proc. of the 10th Int. Conf. on Autonomous Agents and Multiagent Systems*, Taipei, Taiwan, pp. 99–106, 2011.

[9] P. Taillandier and S. Stinckwich, "Using the PROMETHEE multi-criteria decision making method to define new exploration strategies for rescue robots," in *Proc. of the IEEE Int. Sym. on Safety, Security, and Rescue Robotics*, Kyoto, Japan, pp. 321–326, 2011.

[10] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. of the IEEE Int. Sym. on Computational Intelligence in Robotics and Automation*, Monterey, California, USA, pp. 146–151, 1997.

[11] M. Selin, M. Tiger, D. Duberg, F. Heintz and P. Jensfelt, "Efficient autonomous exploration planning of large scale 3D-environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1699–1706, 2019.

[12] L. Lu, C. Redondo and P. Campoy, "Optimal frontier-based autonomous exploration in unconstructed environment using RGB-D sensor," *Sensors*, vol. 20, no. 22, pp. 1–16, 2020.

[13] W. Gao, M. Booker, A. Adiwahono, M. Yuan, J. Wang *et al.,* "An improved frontier-based approach for autonomous exploration," in *15th IEEE Int. Conf. on Control, Automation, Robotics and Vision (ICARCV)*, Singapore, pp. 292–297, 2018.

[14] G. Li, W. Chou and F. Yin, "Multi-robot coordinated exploration of indoor environments using semantic information," *Science China Information Sciences*, vol. 61, no. 7, pp. 1–8, 2017.

[15] J. J. Lopez-Perez, U. H. Hernandez-Belmonte, J. P. Ramirez-Paredes, M. A. Contreras-Cruz and V. Ayala-Ramirez, "Distributed multirobot exploration based on scene partitioning and frontier selection," *Mathematical Problems in Engineering*, vol. 2018, pp. 1–17, 2018.

[16] A. Vellucci, "Multi-robot frontier-based exploration strategies for mapping unknown environments," M.S. thesis. Politecnico di Torino, Torino, Italy, 2019.

[17] C. Stachniss and W. Burgard, "Exploring unknown environments with mobile robots using coverage maps," in *Proc. of the 18th Int. Joint Conf. on Artificial Intelligence*, Acapulco, Mexico, pp. 1127–1132, 2003.

[18] H. H. Gonzales-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *International Journal of Robotics Research*, vol. 21, no. 10, pp. 829–848, 2002.

[19] A. Visser and B. A. Slamet, "Including communication success in the estimation of information gain for multi-robot exploration," in *Proc. of the 6th Int. Sym. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops*, Berlin, Germany, pp. 680–687, 2008.

[20] F. Amigoni and A. Gallo, "A multi-objective exploration strategy for mobile robots," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Barcelona, Spain, pp. 3850–3855, 2005.

[21] R. Semenas and R. Bausys, "Autonomous navigation in the robots' local space by multi criteria decision making," *Open Conf. of Electrical, Electronic and Information Sciences*, Vilnius, Lithuania, pp. 1–6, 2018.

[22] M. Luperto, D. Fusi, N. Alberto Borghese and F. Amigoni, "Exploiting inaccurate a priori knowledge in robot exploration," in *Proc. of the 18th Int. Conf. on Autonomous Agents and MultiAgent Systems*, Montreal, Quebec, Canada, pp. 2102–2104, 2019.

[23] H. Ardiny, S. Witwicki and F. Mondada, "Autonomous exploration for radioactive hotspots localization taking account of sensor limitations," *Sensors*, vol. 19, no. 20, pp. 292, 2019.

[24] R. Shrestha, F.-P. Tian, W. Feng, P. Tan and R. Vaughan, "Learned map prediction for enhanced mobile robot exploration," in *Proc. of the Int. Conf. on Robotics and Automation*, Montreal, Quebec, Canada, pp. 1197–1204, 2019.

[25] S. Oßwald, M. Bennewitz, W. Burgard and C. Stachniss, "Speeding-up robot exploration by exploiting background information," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 716–723, 2016.

[26] A. Viseras Ruiz and C. Olariu, "A general algorithm for exploration with gaussian processes in complex, unknown environments," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Seattle, Washington, USA, pp. 3388–3393, 2015.

[27] D. Holz, N. Basilico, F. Amigoni and S. Behnke, "A comparative evaluation of exploration strategies and heuristics to improve them," in *Proc. of the European Conf. on Mobile Robotics*, Oerebro, Sweden, 2011.

[28] R. Graves and S. Chakraborty, "A linear objective function-based heuristic for robotic exploration of unknown polygonal environments," *Frontiers in Robotics and AI*, vol. 5, pp. 325, 2018.

[29] C. Wang, W. Chi, Y. Sun and M. Q.-H. Meng, "Autonomous robotic exploration by incremental road map construction," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 4, pp. 1720–1731, 2019.

[30] J. Williams, S. Jiang, M. O'Brien, G. Wagner, E. Hernandez *et al.,* "Online 3D frontier-based UGV and UAV exploration using direct point cloud visibility," in *IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Karlsruhe, Germany, pp. 263–270, 2020.

[31] C. Gomez, A. C. Hernandez and R. Barber, “Topological frontier-based exploration and map-building using semantic information,” *Sensors*, vol. 19, no. 20, pp. 4595, 2019.

[32] K. Alexis, “Resilient autonomous exploration and mapping of underground mines using aerial robots,” in *Proc. of the 19th Int. Conf. on Advanced Robotics (ICAR)*, Belo Horizonte, Brazil, pp. 1–8, 2019.

[33] R. Mata, “Persistent autonomous exploration, mapping and localization,” M.S. thesis. Massachusetts Institute of Technology, Massachusetts, USA, 2017.

[34] S. Liu, S. Li, L. Pang, J. Hu, H. Chen *et al.,* “Autonomous exploration and map construction of a mobile robot based on the TGHM algorithm,” *Sensors*, vol. 20, no. 22, pp. 490, 2020.

[35] T. Cieslewski, E. Kaufmann and D. Scaramuzza, “Rapid exploration with multi-rotors: A frontier selection method for high speed flight,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Vancouver, British Columbia, Canada, pp. 2135–2142, 2017.

[36] B. Yamauchi, A. Schultz, W. Adams and K. Graves, “Integrating map learning, localization and planning in a mobile robot,” in *Proc. of the IEEE Int. Sym. on Intelligent Control*, Gaithersburg, Maryland, USA, pp. 331–336, 1998.

[37] A. Franchi, L. Freda, G. Oriolo and M. Vendittelli, “A decentralized strategy for cooperative robot exploration,” in *Proc. of the 1st Int. Conf. on Robot Communication and Coordination*, Athens, Greece, pp. 1–8, 2007.

[38] M. Kulich, T. Juchelka and L. Preucil, “Comparison of exploration strategies for multi-robot search,” *Acta Polytechnica*, vol. 55, no. 3, pp. 162–168, 2015.

[39] W. Burgard, M. Moors, C. Stachniss and F. Schneider, “Coordinated multi-robot exploration,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–386, 2005.

[40] S. Koenig and M. Likhachev, “D* Lite,” in *Proc. of the Eighteenth National Conf. on Artificial Intelligence*, Edmonton, Canada, pp. 476–483, 2002.

[41] S. Koenig and M. Likhachev, “Fast replanning for navigation in unknown terrain,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354–363, 2005.

[42] D. H. Kim, N. T. Hai and S. K. Jeong, “A guide to selecting path planning algorithm for automated guided vehicle (AGV),” in *AETA 2017-Recent Advances in Electrical Engineering and Related Sciences: Theory and Application*, Ho Chi Minh City, Vietnam, pp. 587–596, 2017.

[43] M. Jann, S. Anavatti and S. Biswas, “Path planning for multi-vehicle autonomous swarms in dynamic environment,” in *Ninth Int. Conf. on Advanced Computational Intelligence (ICACI)*, Doha, Qatar, pp. 48–53, 2017.

[44] N. Zagradjanin, A. Rodic, D. Pamucar and B. Pavkovic, “Cloud-based multi-robot path planning in complex and crowded environment using fuzzy logic and online learning,” *Information Technology and Control*, vol. 50, no. 2, pp. 357–374, 2021.

[45] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz and S. Thrun, “Anytime dynamic A*: An anytime, replanning algorithm,” in *Proc. of the Int. Conf. on Automated Planning and Scheduling*, Monterey, USA, pp. 262–271, 2005.

[46] N. Zagradjanin, D. Pamucar and K. Jovanovic, “Cloud-based multi-robot path planning in complex and crowded environment with multi-criteria decision making using full consistency method,” *Symmetry*, vol. 11, no. 10, pp. 1–15, 2019.

[47] C. L. Hwang and K. Yoon, “Methods for multiple attribute decision making,” in *Lecture Notes in Economics and Mathematical Systems—Multilple Attribute Decision Making*. Vol. 186. Berlin, Germany: Springer-Verlag, pp. 58–191, 1981.

[48] I. Petrovic and M. Kankaras, “A hybridized IT2FS-DEMATEL-AHP-TOPSIS multicriteria decision making approach: Case study of selection and evaluation of criteria for determination of air traffic control radar position,” *Decision Making: Applications in Management and Engineering*, vol. 3, no. 1, pp. 146–164, 2020.

[49] S. H. Zolfani, M. Yazdani, D. Pamucar and P. Zarate, “A VIKOR and TOPSIS focused reanalysis of the MADM methods based on logarithmic normalization,” *Facta Universitatis Series: Mechanical Engineering*, vol. 18, no. 3, pp. 341–355, 2020.

[50] K. R. Ramakrishnan and S. Chakraborty, “A cloud TOPSIS model for green supplier selection,” *Facta Universitatis Series: Mechanical Engineering*, vol. 18, no. 3, pp. 375–397, 2020.

[51] T. Cakar and B. Çavuş, "Supplier selection process in dairy industry using fuzzy TOPSIS method," *Operational Research in Engineering Sciences: Theory and Applications*, vol. 4, no. 1, pp. 82–98, 2021.

[52] M. Zizovic, D. Pamucar, B. Miljkovic and A. Karan, "Multiple-criteria evaluation model for medical professionals assigned to temporary SARS-CoV-2 hospitals," *Decision Making: Applications in Management and Engineering*, vol. 4, no. 1, pp. 153–173, 2021.

[53] E. K. Zavadskas, A. Kaklauskas, A. Banaitis and N. Kvederyte, "Housing credit access model: The case for Lithuania," *European Journal of Operational Research*, vol. 155, no. 2, pp. 335–352, 2004.

[54] D. Pamucar and L. Savin, "Multiple-criteria model for optimal off road vehicle selection for passenger transportation: BWM-COPRAS model," *Military Technical Courier*, vol. 68, no. 1, pp. 28–64, 2020.

[55] T. Milosevic, D. Pamucar and P. Chatterjee, "A model for selection of a route for the transport of hazardous materials using fuzzy logic system," *Military Technical Courier*, vol. 69, no. 2, pp. 355–390, 2021.

[56] Z. Ali, T. Mahmood, K. Ullah and Q. Khan, "Einstein geometric aggregation operators using a novel complex interval-valued pythagorean fuzzy setting with application in green supplier chain management," *Reports in Mechanical Engineering*, vol. 2, no. 1, pp. 105–134, 2021.

[57] M. R. Gharib, "Comparison of robust optimal QFT controller with TFC and MFC controller in a multi-input multi-output system," *Reports in Mechanical Engineering*, vol. 1, no. 1, pp. 151–161, 2020.

[58] S. Kayapinar Kaya, "Evaluation of the effect of COVID-19 on countries' sustainable development level: A comparative MCDM framework," *Operational Research in Engineering Sciences: Theory and Applications*, vol. 3, no. 3, pp. 101–122, 2020.

[59] H. Gao, W. Huang and X. Yang, "Applying probabilistic model checking to path planning in an intelligent transportation system using mobility trajectories and their statistical data," *Intelligent Automation & Soft Computing*, vol. 25, no. 3, pp. 547–559, 2019.

[60] N.Ç. Polat, G. Yaylali and B. Tanay, "A method for decision making problems by using graph representation of soft set relations," *Intelligent Automation & Soft Computing*, vol. 25, no. 2, pp. 305–311, 2019.