




A Fuzzy Logic Approach and Path Algorithm for Time and Energy Management of Smart Cleaning Robots

Hakan SIMSEK^{1,*} , Faize Nur ERTURK² , Recep SEKER² 

¹*Antalya Bilim University, Department of Industrial Engineering, Antalya, Turkey*

²*Antalya Bilim University, Department of Computer Engineering, Antalya, Turkey*

Highlights

- This paper focuses on time and energy management of smart cleaning robots.
- An algorithm with clumsiness probability is proposed to keep home's dirt level as low as possible.
- A fuzzy inference system is also presented for estimating the battery life of smart cleaning robots.

Article Info

Received: 17 Dec 2021
Accepted: 15 June 2022

Keywords

Algorithms
Cleaning Robot
Dirt
Fuzzy Inference System

Abstract

Smart home technologies (SHM) or devices provide some degree of digitally connected, automated, or enhanced services to building occupants in residential areas and have been becoming increasingly popular in recent years. SHM have the potential to improve home comfort, convenience, security, and energy management. Different technologies are used to equip household parts for smarter monitoring, movement, and remote control and to allow effective harmonic interaction between them. Especially, energy management and path-planning algorithms are some of the important problems for such technologies to get optimum efficiency and benefit. Smart vacuum cleaning robot is one of the applications of such devices with various functions. These cleaning robots have limited battery power and battery sizes, thus effective cleaning is critical. Additionally, the shortest / optimal path planning is essential for the efficient operation of effective cleaning based on the battery time. In this article, two distinct algorithms, which are Search algorithm and CSP algorithm are utilized to obtain distinct optimal minimum path lengths for keeping the home's total dirt level as low as possible. Depending on various types of linguistic, abstract, or perceptual variables, these algorithms are not enough for the energy management of the battery. Therefore, the fuzzy logic-based inference system is proposed for obtaining the charge durability of battery of the cleaning robot, in addition to these algorithms. The inputs affecting the charge durability are considered as floor type, dirt level and the width of area for the fuzzy approach.

1. INTRODUCTION

Smart cleaning machine or robot vacuum cleaner or smart cleaner is a type of vacuum cleaner that's works with our human interaction depending on a certain algorithm and system. The robot vacuum cleaner market is expected to reach a compound annual growth rate of 13% until 2026. It is estimated by the International Robotics Federation (IFR) that 31 million home robots were sold between 2016 and 2019, of which 96% are vacuum and floor cleaning robots¹. In another market analysis report, the global robotic vacuum cleaner market size has been valued at USD 2.56 billion in 2019 and is estimated to grow at a CAGR of 17.7% from 2020 to 2027² (Figure 1). Many major factors, like lifestyle, ease of usage, occupational injuries, currently COVID-19 outbreak and so on, lead to increase the growth of the market worldwide. Cleaning, especially for public places, has also become much more crucial for the areas used individually and collectively because of the COVID-19 pandemic. This occurrence resulted in an upsurge in demand for cleaning supplies, cleaning tools, and cleaning robots for use in both home and industrial settings.

¹ <https://www.mordorintelligence.com/industry-reports/robot-vacuum-cleaners-market>; Robot Vacuum Cleaners Market-Growth, Trends, COVID-19 Impact and Forecasts (2021- 2026)

² <https://www.grandviewresearch.com/industry-analysis/robotic-vacuum-cleaner-market>

*Corresponding author, e-mail: hakan.simsek@antalya.edu.tr

An algorithm is a finite set of well-defined, computer-executable instructions for solving a particular class of problems or performing a calculation. Algorithms are always precise and are used as specifications for computations, data manipulation, automated reasoning, and other tasks. On the other hand, fuzzy logic is a mathematical tool that models uncertainty. It enables us to consider artificial intelligence approach like human reasoning to handle the diverse type uncertainties. In the literature, there are different studies about smart cleaning robots using related algorithms and fuzzy logic.

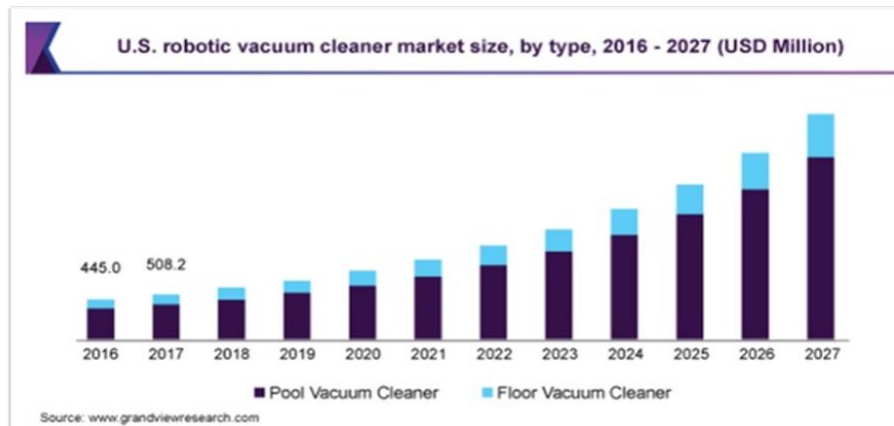


Figure 1. US robotic vacuum cleaner market size, 2016-2027 (USD million)

Galceran and Carreras [1] represented a survey study about coverage path planning methods for robotics. Viet et al. [2] investigated smart cleaning robots in obscure working areas based on the boustrophedon motions and the A* search algorithm. Vu Le et al. [3] contributed exploring the possibility to utilize evolutionary algorithms for finding the path providing the minimum energy consumption. Pham et al. [4] presented Smooth-STC Algorithm for identifying a proper route, minimizing backtracking, and maximizing coverage in any described functional environment in mobile applications. Bai and Hsueh [5] worked on a path planning learning algorithm by means of three-dimensional vector coordinates. Ciabattini et al. [6] presented a fuzzy logic approach to model household electrical consumption of some electrical appliances. Bissey et al. [7] discussed a fuzzy approach to effectively estimate short-term load consumption in individual housing. Belman-Flores et al. [8] studied a fuzzy logic system to keep the temperature of the fresh food partition in the desired level while saving energy for a domestic refrigerator. Soetedjo et al. [9] proposed a Home Energy Management System based on fuzzy approach to minimize electricity costs. Hemi et al. [10] implemented power management strategy by considering the fuzzy approach in a hybrid electric vehicle to preserve the battery from overcharging pending the braking energy accumulation. Agarwal et al. [11] presented a general review of route planning and obstacle avoidance algorithms for smart mobile robotics such as artificial potential field, rapidly exploring random trees, Fuzzy technique, A* algorithm, modified local path planning algorithm and so on. Kontogiannis et al. [12] designed and implemented a fuzzy system processing environmental data based on weather parameters to obtain minimum energy consumption for a residential building. Yang et al. [13] presented a questionnaire of route algorithms, which can be applied in aerial robots, ground robots, and underwater robots, and classified all these approaches into five frameworks. Waqar and Demetgül [14] used fuzzy logic approach for controlling the suction power of automatic vacuum cleaner. Mohanty and Parhi [15] investigated a meta-heuristic algorithm for a precisely autonomous mobile robot path planning in an obscure environment with static barriers. Mohamed et al. [16] investigated the particle flock optimization used to figure out the path planning problem for mobile robots. Juang and Chang [17] proposed an EGPSSO-designed fuzzy system and applied to mobil robot navigation in obscure places. Sasaki et al. [18] studied a path-planning algorithm considering dust for cleaning robots that prioritize and sort areas with large amounts of dirt. Ong and Azir [19] aimed to design a low-cost and smart IoT based vacuum cleaner by using mapping algorithm for cleaning homes or offices. Ravankar et al. [20] proposed an algorithm for dirt detection with an outer camera and communicate with a robot in real-world circumstances. Ciliz [21] applied a tuning methodology for controlling a household vacuum cleaner based on fuzzy logic. Vu Le et al. [22] discussed complete coverage path planning algorithm for generating an efficient navigation path for reconfigurable cleaning robots composed of tetramond. Song et al. [23] implemented a cuckoo search method considering compact and parallel

methodologies for route problems of unmanned robots in three dimensions and proposed a new parallel communication strategy. Oprea et al. [24] focused on Smart Adaptive Switching Module architecture to effectively incorporate renewable energy sources into power systems using fuzzy logic. Yakoubi and Laskri [25] suggested an evolutionary technique to address route problem of a vacuum cleaner in a coverage zone in a room environment using Genetic Algorithms.

Almost all cleaning robots currently on the market could have some efficiency problems like charge durability, finding the most convenient path. Normally, these robots don't know if the room is clean or not, until they come to inspect it or, robots may be completely unaware of the room's sudden contamination (clumsiness). For this purpose, camera utilizing is suggested in [20], for example. In our approach, clumsiness is an important assessment scale. While there is more than one way to evaluate or measure this, we have calculated it by giving a clumsiness ratio to the areas where clumsiness may occur the most/least. Another major issue with cleaning robots is their battery power. The battery life of these cleaning robots is an important research topic.

The motto of this paper is to manage the battery life of cleaning robots using a fuzzy inference technique and to construct alternative optimal path-finding algorithms including the clumsiness probability and the Viterbi algorithm. In Fuzzy Inference System (FIS), there are three linguistic inputs as follows: floor type, pollution level of the area to be cleaned, and size of the area to be cleaned. Based on these inputs, the result of the FIS allows us to estimate the battery life of the smart cleaning robot. On the other hand, for the algorithm part, there have been several different types of optimal path planning algorithms proposed in the literature; Q-learning algorithm, collision avoidance algorithm, cuckoo search algorithm, genetic algorithms and so on. In this study, two different algorithms are used for finding best optimal path for specific cleaning robot, considering the clumsiness situation. These are the search algorithm and the CSP (Constraint Satisfaction Problems) algorithm. These two algorithms use distinct rules to try to discover the most optimal path. For this, we constructed a simple house model with four rooms and corridors.

The layout of the article is as follows. The section 2 explains the optimal path algorithms of the smart cleaning robot. In Section 3, basic information about fuzzy approach is presented for the charge durability. Finally, discussion and conclusion parts are given in Section 4.

2. FINDING OPTIMAL PATH ALGORITHM

The code includes two types of algorithms: search algorithm and CSP algorithm. The algorithm starts with current state data created manually by the user or detectable by input devices such as cameras [20]. Unlike some other studies that can be considered close to itself, this algorithm includes a method that is not related to the travel of the cleaning robot in a room, but to the evaluation of the sequence of transitions from one room to another according to the dirtiness. For example, let's assume that our house has four rooms A, B, C and D. The method outlined in this article does not work on how the cleaning robot will determine a cleaning path inside one of these rooms. The specified method decides in what order the cleaning robot is going to clean rooms A, B, C and D. In the decided room visiting path, there is only information such as which room to go to in which order to perform the cleaning action, and whether it is necessary to go to another room in the next unit time in case of going to the room to perform the cleaning action. As a result of the room permutation found with the room visiting path, the cleaning robot determines whether to stay in the room in the next step and continue cleaning in the same room, or whether to go to another room and continue the cleaning action there and perform the necessary steps to achieve this. On the other hand, if the robot decides that clumsiness is occurred in a room according to the clumsiness probabilities, the dirt level of the responsible room is going to increase by their own clumsiness dirt value. Clumsiness is occurred randomly, based on clumsiness probabilities for each room. At last, we again do the calculations for the optimal visiting path.

2.1. Search Algorithm

The searching algorithm is a plan of action that looks at specific data among a group of data by approaching them to step by step. Figure 2 shows general flow of search algorithm. The process part can differ for different types of the search algorithm. In the Search Algorithm, the main purpose is to keep the general dirt value of the house as low as possible.

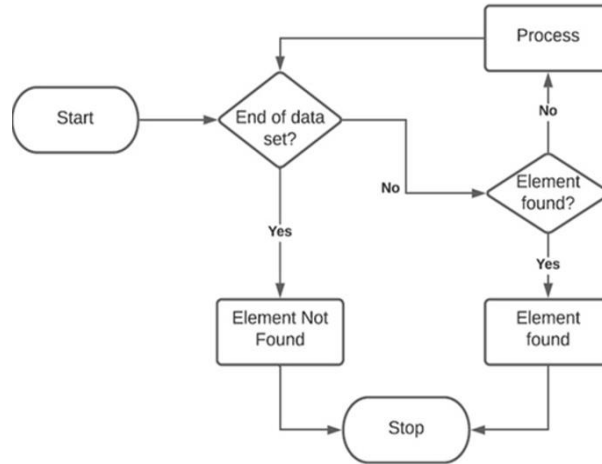


Figure 2. General flow of optimal path finding algorithm

Figure 3 shows the flowchart of the finding optimal path with search algorithm. Initial values are given by user. This values provide information for the robot about the **dirt level** (current dirt value of a room), the **max dirt level** (maximum possible dirt value of a room), the **clumsiness probability** (probability of dirt increase caused by clumsiness for a room), the **dirt increase per moment** (average dirt, which caused by dust, shedded hair to the floor, etc., **increase value** per unit time for a room, and the **clumsiness dirt value** (dirt value that caused by clumsiness for a room). By using these input values, robot finds the optimal path to keep home as much as clean.

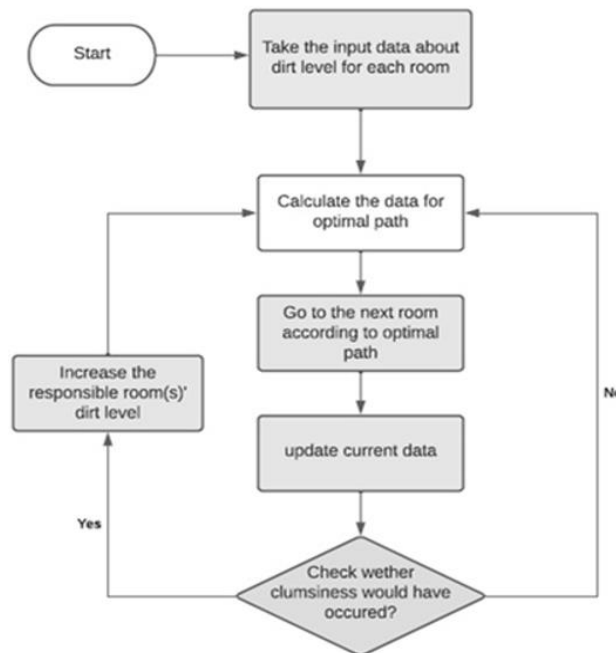


Figure 3. Flowchart of search algorithm

In the Search Algorithm, the idea behind the Viterbi Algorithm, [26], is used to avoid the calculation of unnecessary optimal path options and, thanks to this method, we have provided the fast calculation

technique. Therefore, we could find the optimal paths that are focusing on a future time calculations faster. The Viterbi algorithm is a graphical approach used in NLP (Natural Language Processing) technologies. A generalization of the Viterbi algorithm, called the max-sum (or max-product) algorithm, can be used to discover the most likely assignment of all or some subset of hidden variables in many graphical models. This algorithm can be applied to proposed approach by considering the nodes as rooms to be visited in order, shown in Figure 4³. On the other hand, nodes that are not selected as path can be considered as the other path sequence that will not create an optimal transition.

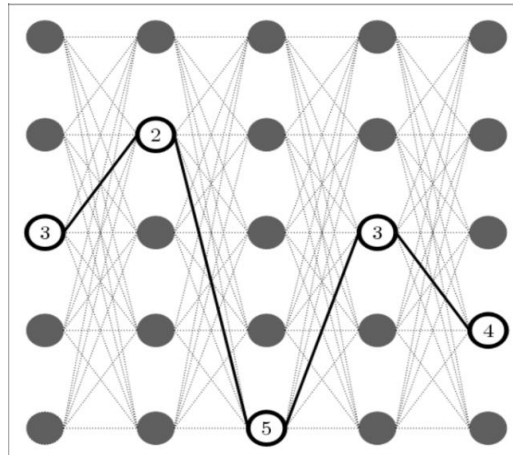


Figure 4. Viterbi Algorithm

Every room has an increasing dirt value per moment, meaning the dirt will increase for each room per unit time. This frame can be installed manually by user's instruction. This increase is determined as the sum of the function that calculate the normal dirt increase per unit time, $S(x_i, x'_i)$, and the function that controls the dirt increase due to the clumsiness probability of the existing room, $P(x_i, x_i^{**}, x_i^*)$. This is given by the equation

$$\sum_i^L [S(x_i, x'_i) + P(x_i, x_i^{**}, x_i^*)]$$

where L is known as list of rooms, which has the clumsiness probability of each room. The variables are defined as

- x_i : current dirt of the room(s),
- x'_i : standard increase in unit time,
- x_i'' : maximum dirt level of the room
- x_i^* : probability of being clumsy,
- x_i^{**} : value to be added in cafe of clumsiness.

The functions are given as

$$S(x_i, x'_i) = x_i + x'_i$$

and

$$P(x_i, x_i^{**}, x_i^*) = \begin{cases} 0, & \text{if the probability of } x_i^* \text{ is not occurred} \\ x_i + x_i^*, & \text{if the probability of } x_i^* \text{ is occurred.} \end{cases}$$

³ <https://commons.wikimedia.org/wiki/File:Hmm-Viterbi-algorithm-med.png>

$P(x_i, x_i^{**}, x_i^*)$ explains that if clumsiness occurs, then the function adds to the dirt of the responsible room. If there is no clumsiness, it returns zero, so it does not increase the dirt of the room. Also, in the case $x_i > x_i''$, the dirt value of the room is set to the value of x_i'' .

2.2. CSP (Constraint Satisfaction Problem) Algorithm

Constraint satisfaction problems (CSPs) are mathematical queries defined as a set of objects whose

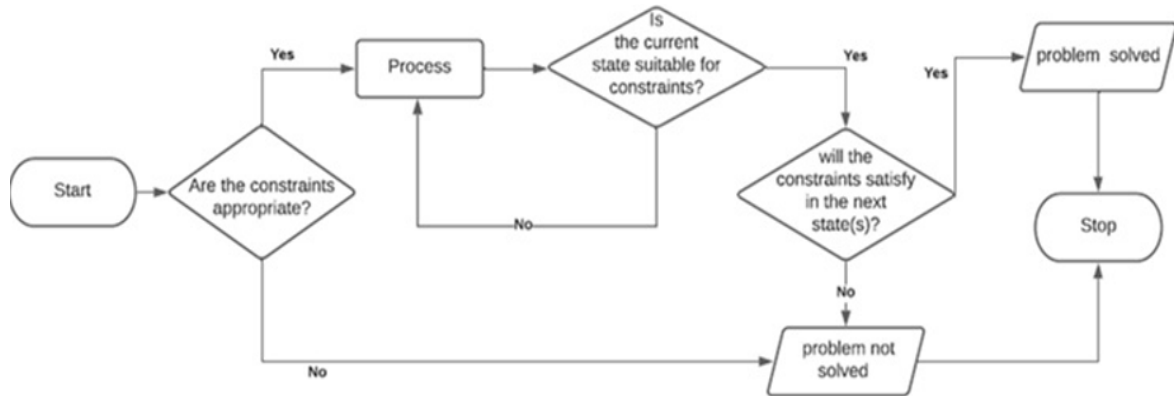


Figure 5. General flow of CSP algorithms

state must satisfy several constraints or limitations. Many issues in artificial intelligent can be modeled as CSP. Hence, improvements of effective solution techniques for CSPs are a notable area of research. You can see the general flow of CSP algorithm in Figure 5.

In this algorithm, we have some limitation (constraint) about the maximum dirty level for some or each room and, we also have a threshold level for each room. The threshold shows us what we want for maximum dirt level (maximum possible dirt value of a room), and the limitation tells us how dirty each room can be. If the dirt level in at least one of the rooms exceeds its threshold, the cleaning robot assumes this situation as a problem. To solve this problem, our cleaning robot tries to find the path that provides the current problem satisfied by finding the room visiting path with the minimum number of rooms needed to be cleaned. If none of the room's dirt exceed its own threshold, our cleaning robot selects the next room as the room that exceed its dirt limit for the longest time.

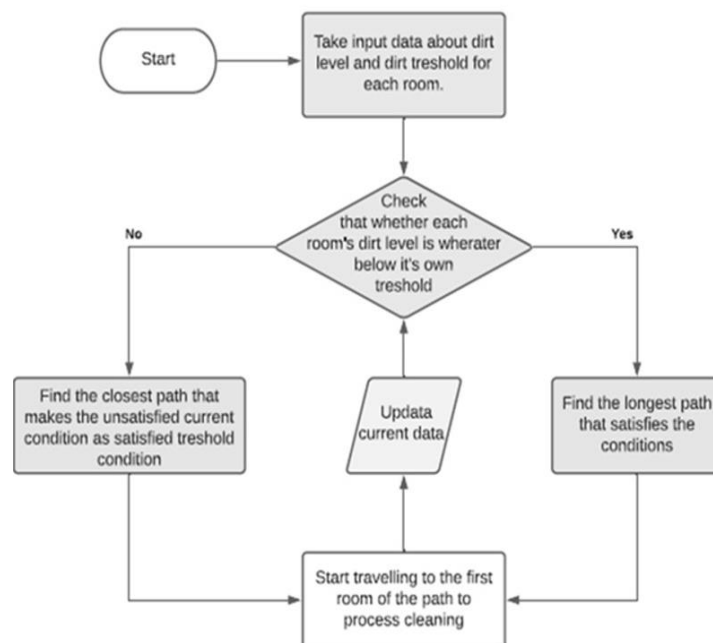


Figure 6. Flowchart of optimal path with CSP algorithm

Figure 6 shows the flowchart of the finding optimal path with CSP algorithm. Initial values given by user provide information for the robot about dirt level (current dirt value of a room), max dirt level (maximum possible dirt value of a room), clumsiness probability (probability of dirt increase caused by clumsiness for a room), dirt increase per moment (average dirt, which caused by dust, shedded hair to the floor, etc.), increase value per unit time for a room and also room constraints about dirt level (The given constraints says which room can get how much dirty). By using these input values, robot finds the optimal path to keep limitation given rooms' dirt level under their own dirt threshold. The related mathematical formula is as follows,

$$\sum_i^L C(x_i, x_i^{***})$$

where

$$C(x_i, x_i^{***}) = \begin{cases} 0, & \text{if } x_i \leq x_i^{***} \\ 1, & \text{if } x_i > x_i^{***} \end{cases}$$

x_i^{***} : room's dirt threshold,

x_i : current dirt of the room(s).

If the return of C (check) function values is not zero, at least one of the room's threshold conditions is not satisfied.

2.3. Experiment and Results

The following two code parts are our initial values given by manually. First part of the code shows data for search algorithm and second one shows for CSP algorithm initial values, respectively

```
Faize = Room ("Faize", 20, 0.25, 50, 7, 7, 15)
Kitchen = Room ("Kitchen", 10, 0.50, 70, 10, 15, 10)
livingRoom = Room ("Living Room", 10, 0.35, 70, 6, 8, 10)
recep = Room ("Recep", 7, 0.20, 40, 3, 5, 15)
myList = [faize,kitchen,livingRoom,recep]
```

```
faize2 = Room ("Faize", 20, 0.25, 50, 5, 7, 15)
kitchen2 = Room ("Kitchen", 10, 0.50, 70, 10, 15, 10)
livingRoom2 = Room ("Living Room", 10, 0.35, 70, 6, 8, 10)
recep2 = Room ("Recep", 7, 0.20, 40, 3, 5, 15)
myList2 = [faize2, kitchen2, livingRoom2, recep2].
```

Parameters of the room function represent the name of the room (x_i), standart increase in unit time (x_i'), maximum dirt level of the room (x_i''), probability of being clumsy (x_i^*), value to be added in case of clumsiness (x_i^{**}), the room's dirt threshold (x_i^{***}), for CSP algorithm, respectively. The data for the search algorithm is shown visually in Figure 7. For example, necessary initial values of the Room1 are given as follows

```
current dirt of the room is ( $x_i$ ) = 20,
standard increase in unit time is ( $x_i'$ ) = 0.25,
maximum dirt level of the room ( $x_i''$ ) = 50,
probability of being clumsy ( $x_i^*$ ) = 7,
value to be added in case of clumsiness ( $x_i^{**}$ ) = 7
and the room's dirt threshold ( $x_i^{***}$ ) = 15.
```



Figure 7. Data for Search Algorithm with Bird's Eye view of a House

The cleaning robot using this algorithm works on minimizing the total dirt level with appropriate timing. For example, let the cleaning machine start from room1 with this initial data and suppose there is a battery that can advance 4 steps. In other words, we assume that the machine will visit 4 different rooms. At the end of this process, what the machine wants to achieve is to provide the minimum level of total dirt. In other words, the machine chooses the path that is at the minimum of the total dirt for the cleaning process. The robot continues to stay in the starting room in line with the values we give and will spend its first step in this room. For the next step, the machine can go to the kitchen and after the cleaning process here, it can go to the living room and then come back to the kitchen. At the same time, this process takes also into account the clumsiness occurring randomly depending on the percentage of clumsiness.

```
for i in range(5):
    pathFindSearch(search,10,myList)
    pathFindCSP(csp,2,myList2)
```

Figure 8. Part of the Code

The part of the code shown in Figure 8 tells us how much the iteration is and can be changed manually. For instance, if we delete the number 5 here and write 9 instead for the "range" parenthesis, this number will be our new iteration number and the process continues for 9 steps. The second parameter of the functions represents the number of room that our room visiting path has as a length. Meaning that, the most optimal visiting path for cleaning action of ten rooms will be given if the integer 10 is given in the second parameter of **pathFindSearch** or **pathFindCSP** functions. Third parameter of the given functions

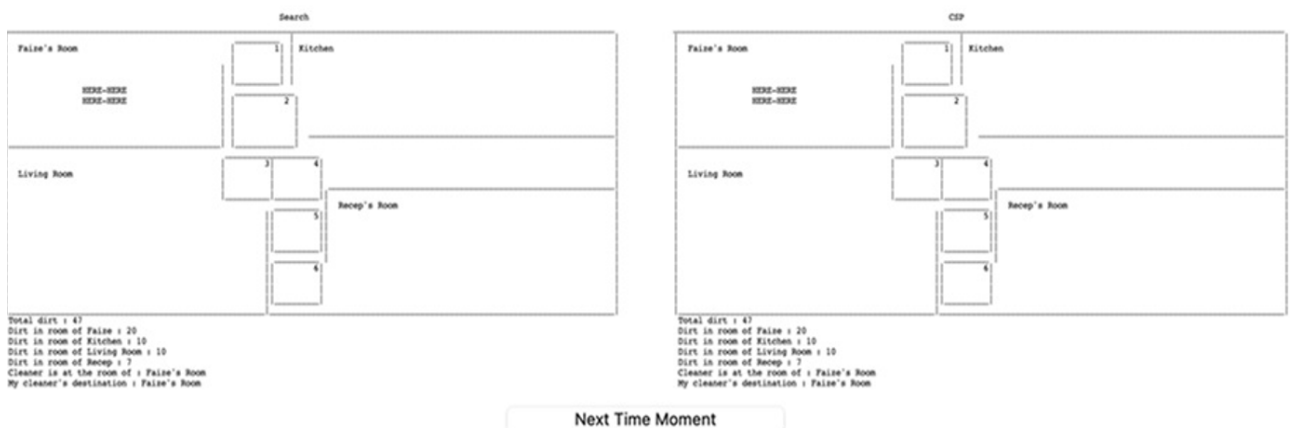


Figure 9. The Image of the Structure of Home with Code Parameters

represent the room lists. The reason for two different variables given in the function is to be able to see the differences of the Search and CSP comparison. More accurate room visiting path is provided by the algorithms if the second parameter's value incremented. Given screenshots in Figure 9 show the manner of work of Search (left side) and CSP (right side) method.

When the experiment is started, the cleaning robot starts its journey to determine the next room that needs to be cleaned and go to this room, as a result of the evaluations made by the two different methods. When the experiment progressed for a while, it was observed that the Search and CSP methods determined different routes. While the Search method evaluated the house as a whole, the CSP method evaluated each room separately. So much so that, in the CSP method, when we equated the maximum pollution level, which is called threshold, that we want a room to be to the maximum pollution level that the room itself can have, our cleaning robot did not stop by to clean this room and acted as if such a room did not exist at home. In this way, it has been observed that the rooms that will not be used within a certain period or that do not need to be cleaned can only be cleaned at desired times with the CSP method.

Clumsiness Calculation: Evaluation of the probability of occurrence of clumsiness is carried out separately for each room by reference to the clumsiness probability values determined for each room. In this calculation, the corresponding parameter of the object created for the room is taken; then, a random value is generated. If the value taken from the parameter value of the object is less than the randomly generated value, it is assumed that the clumsiness has occurred. The value range of the randomly generated value is determined between 0-100 if the parameter is an integer. If the value range of the randomly generated value is in a float type, the value is multiplied with the value of 10, until the randomly generated value is equal to the integer version of itself, such as 5.0 is equal to the 5. At the same time, the top limit of the range, which is 100, is also multiplied with the value of 10. With the completion of these multiplication operations, the value range of the randomly generated number is found to evaluate the probability of clumsiness according to the same process.

3. PROPOSED FUZZY INFERENCE SYSTEM

Fuzzy Logic is a field of science that includes artificial intelligence approaches like human reasoning to handle diverse types of ambiguity. Fuzzy Inference Systems provide an output by operating the values of the vague inputs of a problem that can be associated with linguistic parameters. Fuzzy inference system is a combination of **a fuzzification interface** that transforms the crisp inputs into fuzzy values with linguistic parameters; **a rule base** including fuzzy IF-THEN rules; **a database** describing the membership functions of the linguistic parameters used in the IF-THEN rules; **a decision-making unit** accomplishing the consequence operations on the IF-THEN rules; and eventually, **a defuzzification interface** that converts the fuzzy values of the consequence into a crisp output (see Figure 10). Further information on this topic can be found in many studies of literature, [27-29].

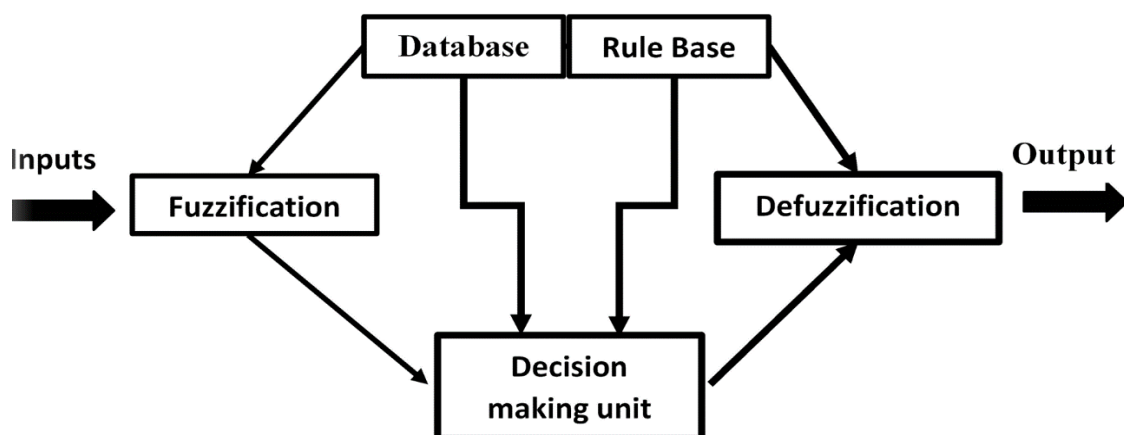


Figure 10. Fuzzy Inference System (FIS)

A fuzzy set \tilde{A} in a universe of discourse X is defined by a membership function

$$\mu_{\tilde{A}}: X \rightarrow [0,1]$$

such that it associates a real number in the range $[0,1]$ with each x element in X . The value $\mu_{\tilde{A}}(x) \in [0,1]$ gives the membership degree of the element x for the fuzzy set \tilde{A} .

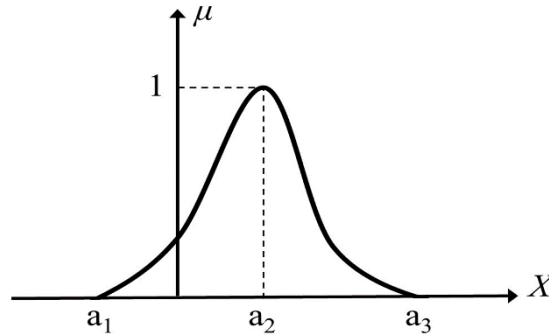


Figure 11. Fuzzy number

Fuzzy numbers are important fuzzy sets satisfying the following conditions (Figure 11):

- Convex fuzzy set (if $\mu_{\tilde{A}}(\gamma x_1 + (1 - \gamma)x_2) \geq \min(\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2))$ for all $\gamma \in (0,1]$ and $x_1, x_2 \in X$, then \tilde{A} is a convex fuzzy set),
- Normalized fuzzy set (if $Core(\tilde{A}) = \{x \in X: \mu_{\tilde{A}}(x) = 1\} \neq \emptyset$, then \tilde{A} is normalized),
- Its membership function is piecewise continuous,
- It is defined in the real number.

There are different classes of fuzzy numbers such as Triangular, Trapezoidal, Gaussian, S-shaped, G-bell, Z-shaped, Sigmoidal fuzzy numbers, which appear in real-world problems. They are used in **fuzzification interface** of a fuzzy system; that is, inputs that can be expressed with linguistic variables are described by fuzzy numbers in a fuzzy environment.

The basis of the FIS is considered to be the extraction of the output fuzzy values from the input fuzzy parameters according to a set of logic inference rules in linguistic terms, determined from the knowledge base based on an expert's opinion. A fuzzy **rule base** occurs a set of IF-THEN rules. All other components are used to run these rules in realistic and efficient manner. A fuzzy IF-THEN rule usually is of the form

R : If x_1 is \tilde{A}_1 AND (OR) x_2 is \tilde{A}_2 THEN y is \tilde{B}

where \tilde{A}_1, \tilde{A}_2 and \tilde{B} are linguistic variables determined by fuzzy numbers on input and output parameters, respectively. Logical operators AND, OR are given as the fuzzy intersection (AND), and fuzzy union (OR), respectively and described as below:

$$\tilde{A}_1 \text{ AND } \tilde{A}_2: \min\{\mu_{\tilde{A}_1}, \mu_{\tilde{A}_2}\}$$

$$\tilde{A}_1 \text{ OR } \tilde{A}_2: \max\{\mu_{\tilde{A}_1}, \mu_{\tilde{A}_2}\}.$$

After determining the rule base, we apply the process by which the fuzzy numbers representing the outputs of each rule are combined into a **single fuzzy number** (set), which is known as **aggregation** of rules. Aggregation could be done by different operators, such as Max, Sum or Probor. The Max operator is generally chosen when compensation between input variables is desirable. The Max operator is determined by:

$$\mu_{output} = \max\{\mu_{rule1}, \mu_{rule2}, \dots, \mu_{rulek}\}.$$

The last step is the defuzzification process, which converts the fuzzy output set obtained after the aggregation step into an exact number. Defuzzification techniques used in fuzzy inference systems are usually Center of area or centroid (CoA), Bisector of area, Small of maxima (SoM), Mean of maxima (MoM), and Largest of maxima (LoM). All these mathematical operations described above can be completed by using the MATLAB Fuzzy Logic Designer.

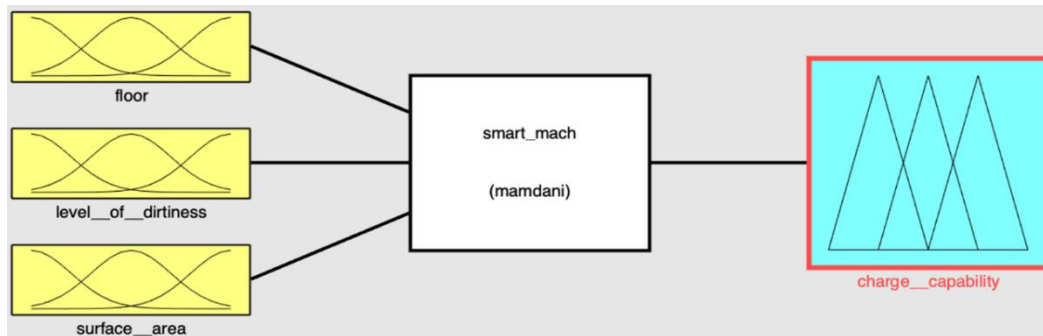


Figure 12. Inputs-Output structure of the FIS

We aim to construct a Mamdani type FIS, which is the most common in practice and in the literature, for the charge durability of the cleaning robot. We implement the **max-min inference** method for composition and minimum operator for implication process. Triangular fuzzy numbers are used for the inputs and output, and the aggregation is chosen as the Max operator. The Centroid is taken as the defuzzification method for the proposed FIS. There are four linguistic variables in the system, three of which are inputs, and one is output. The inputs are given as **Floor Type**, **Level of Dirtiness**, **Surface Area** (60-140 m²) and the output is the **charge durability** shown in Figure 12. The Table 1 shows the linguistic variables of parameters and the corresponding membership functions of the FIS. While the linguistic variables of the floor type are determined as Ceramic, Concrete-Marble, PVC, Wooden-Parquet and Carpet, the dirt level and the surface area are changing between 0-100 and 60-140 m², respectively.

Table 1. Fuzzy Classification of Input and Output Variables

	Fuzzy Membership Functions			
	Input-1	Input-2	Input-3	Output
	Floor Type	Dirt Level	Surface Area	Charge Durability
Range	[0 100]	[0 100]	[60 140]	[0 100]
Linguistic variable 1	Ceramic [0 0 25]	Clean [0 0 25]	60-85 m ² [60 60 85]	Very Short Durability [0 0 20]
Linguistic variable 2	Concrete-Marble [0 25 50]	Almost Clean [0 25 50]	70-110 m ² [60 85 110]	Short Durability [10 30 50]
Linguistic variable 3	PVC [25 50 75]	Not Clean [25 50 75]	85-135 m ² [85 110 135]	Average Durability [30 50 70]
Linguistic variable 4	Wooden-Parquet [50 75 100]	Dirty [50 75 100]	110-140 m ² [110 140 140]	Long Durability [50 70 90]
Linguistic variable 5	Carpet [75 100 100]	Very Dirty [75 100 100]	-	Very Long Durability [80 100 100]

In total, 100 rules are generated for the charge capability, a few of which are shown in Figure 14. In addition, a diagram for the IF-THEN rules started from CERAMIC for the floor type can be seen in Figure 13. The

centroid operation is taken as the defuzzification operation. The impacts of the inputs on charge durability as a rule surface performed in MATLAB are shown in Figure 15 in 3D form.

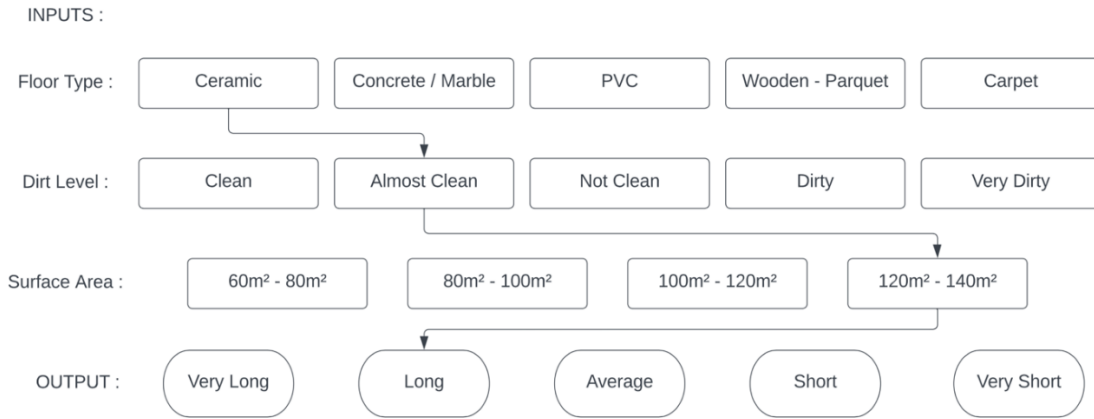
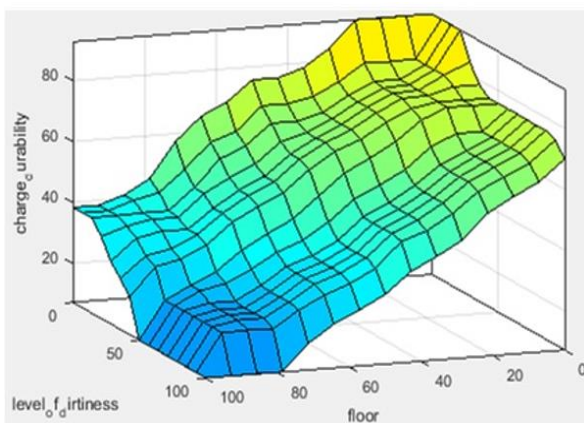


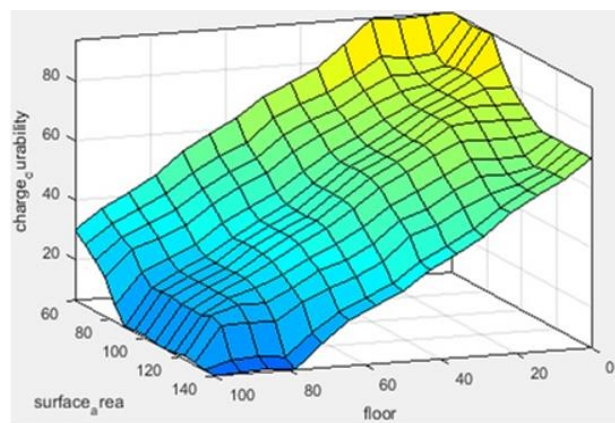
Figure 13. If floor type is CERAMIC and the dirt level is ALMOST CLEAN and surface area is 120m²-140m² then the charge durability is LONG

1. If (floor is ceramic) and (level_of_dirtiness is clean) and (surface_area is 60m²-80m²) then (charge_capability is very_Long_durability) (1)
2. If (floor is ceramic) and (level_of_dirtiness is clean) and (surface_area is 80m²-100m²) then (charge_capability is very_Long_durability) (1)
3. If (floor is ceramic) and (level_of_dirtiness is clean) and (surface_area is 100m²-120m²) then (charge_capability is very_Long_durability) (1)
4. If (floor is ceramic) and (level_of_dirtiness is clean) and (surface_area is 120m²-140m²) then (charge_capability is very_Long_durability) (1)
5. If (floor is ceramic) and (level_of_dirtiness is almost_clean) and (surface_area is 60m²-80m²) then (charge_capability is very_Long_durability) (1)
6. If (floor is ceramic) and (level_of_dirtiness is almost_clean) and (surface_area is 80m²-100m²) then (charge_capability is very_Long_durability) (1)
7. If (floor is ceramic) and (level_of_dirtiness is almost_clean) and (surface_area is 100m²-120m²) then (charge_capability is very_Long_durability) (1)
8. If (floor is ceramic) and (level_of_dirtiness is almost_clean) and (surface_area is 120m²-140m²) then (charge_capability is long_durability) (1)
9. If (floor is ceramic) and (level_of_dirtiness is not_clean) and (surface_area is 60m²-80m²) then (charge_capability is very_Long_durability) (1)
10. If (floor is ceramic) and (level_of_dirtiness is not_clean) and (surface_area is 80m²-100m²) then (charge_capability is very_Long_durability) (1)
11. If (floor is ceramic) and (level_of_dirtiness is not_clean) and (surface_area is 100m²-120m²) then (charge_capability is long_durability) (1)
12. If (floor is ceramic) and (level_of_dirtiness is not_clean) and (surface_area is 120m²-140m²) then (charge_capability is long_durability) (1)
13. If (floor is ceramic) and (level_of_dirtiness is dirty) and (surface_area is 60m²-80m²) then (charge_capability is very_Long_durability) (1)
14. If (floor is ceramic) and (level_of_dirtiness is dirty) and (surface_area is 80m²-100m²) then (charge_capability is very_Long_durability) (1)
15. If (floor is ceramic) and (level_of_dirtiness is dirty) and (surface_area is 100m²-120m²) then (charge_capability is long_durability) (1)
16. If (floor is ceramic) and (level_of_dirtiness is dirty) and (surface_area is 120m²-140m²) then (charge_capability is Average_durability) (1)
17. If (floor is ceramic) and (level_of_dirtiness is very_dirty) and (surface_area is 60m²-80m²) then (charge_capability is very_Long_durability) (1)
18. If (floor is ceramic) and (level_of_dirtiness is very_dirty) and (surface_area is 80m²-100m²) then (charge_capability is long_durability) (1)
19. If (floor is ceramic) and (level_of_dirtiness is very_dirty) and (surface_area is 100m²-120m²) then (charge_capability is long_durability) (1)
20. If (floor is ceramic) and (level_of_dirtiness is very_dirty) and (surface_area is 120m²-140m²) then (charge_capability is Average_durability) (1)

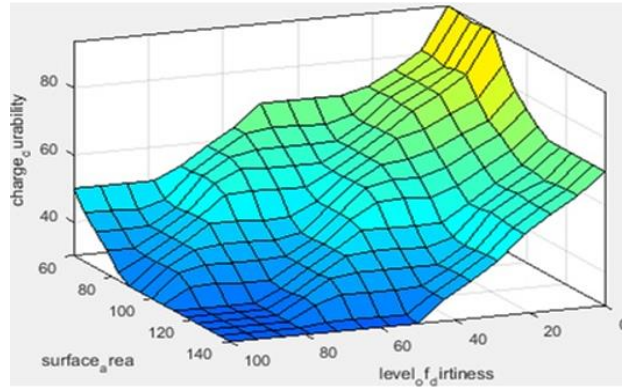
Figure 14. Rule base of the FIS in MATLAB



(a)



(b)



(c)

Figure 15. 3-D Surface Views of the Proposed FIS; a) Dirt Level and Floor Type, b) Surface Area and Floor Type, c) Surface Area and Dirt Level

According to the input values, we compute the output of the FIS that is the charge durability for the cleaning robot. For example, we have three possibilities shown in Table 2. The results show that the charge capacity of the cleaning robot becomes %78.88 in case of Sample-1, %35.4 in case of Sample-2 and, %7.1 in case of Sample-3. Namely, while the vacuum cleaner has a very low charge for Sample-3, the charge capacity of the robot is good for Sample-1. As a result, the proposed system has a reasonable potential and usefulness, and its performance is justified.

Table 2. Application of the Proposed FIS

Inputs	FIS for Charge Capability			Output
	Floor Type	Dirt Level	Surface Area	Charge Durability
Sample-1	23	35	85	78.88
Sample-2	55	70	100	35.4
Sample-3	85	80	120	7.1

4. CONCLUSION

This paper presents search and CSP algorithms for path planning considering the clumsiness situation and explores a FIS design approach to increase the battery life management capability for smart cleaning robots. By giving the initial values for the parameters of the algorithms, the robot finds the optimal path to keep home as much as clean. On the other hand, floor type, dirt level and surface area are used as inputs for the FIS of the charge durability and 100 IF-THEN rules in the rule base provide an inference about the charge situation of the robot.

The advantage of the proposed algorithm is that the cleaning robot can know the dirtiest room to clean when a clumsiness occurs. Also, the proposed fuzzy approach provides an estimation about the charge durability of the cleaning robot before it starts cleaning. One of the distinct properties of the proposed algorithm is the use of the Viterbi algorithm to avoid unnecessary memory usage and provide a fast computational technique. The proposed algorithm is more efficient than video camera location path planning or laser measurement path planning in terms of cost parameter, [5, 30]. On the other hand, the proposed fuzzy component has 100 IF-THEN rules, and the number of input-output linguistic variables increase the sensitivity of the model to obtain more efficient and appropriate results. Such a fuzzy approach practically does not appear in the literature. In addition, an ANFIS (Adaptive Neuro Fuzzy Inference System) learning method can be developed with the same input-output linguistic variables by collecting data day by day through the proposed fuzzy approach to include learning capability on the charging capacity side for the robot. Both approaches can be upgraded and improved easily. This paper simply optimizes the current vacuum cleaners by making them more efficient depending on their environments.

CONFLICTS OF INTEREST

No conflict of interest was declared by the authors.

REFERENCES

- [1] Galceran, E., Carreras, M., "A survey on coverage path planning for robotics", *Robotics and Autonomous Systems*, 61: 1258-1276, (2013).
- [2] Viet, H. H., Dang, V-H., Laskar, N. U., "Chung T, BA*: an online complete coverage algorithm for cleaning robots", *Applied Intelligence*, 39: 217-235, (2013).
- [3] Le, A. V., Ku P. C., Tun, T. T., Nhan, N. H. K., Yuyao, S. Y., Mohan, R. E., "Realization energy optimization of complete path planning in differential drive based self-reconfigurable floor cleaning robot", *Energies*, 12: 1136, (2019).
- [4] Pham, H. V., Moore, P., Truong, D. X., "Proposed smooth-STC algorithm for enhanced coverage path planning performance in mobile robot applications", *Robotics*, 8(2): 44, (2019).
- [5] Bai, Y. W., Hsueh, M. F., "Using an adaptive iterative learning algorithm for planning of the path of an autonomous robotic vacuum cleaner", *The 1st IEEE Global Conference on Consumer Electronics*, 401-405, (2012). DOI: <https://doi.org/10.1109/GCCE.2012.6379640>
- [6] Ciabattoni, L., Grisostomi, M., Ippoliti, G., Longhi, S., "A fuzzy logic tool for household electrical consumption modeling", *39th Annual Conference of the IEEE Industrial Electronics Society*, 8022-8027, (2013). DOI: <https://doi.org/10.1109/IECON.2013.6700474>
- [7] Bissey, S., Jacques, A., Bunetel, J. B. L., "The fuzzy logic method to efficiently optimize electricity consumption in individual housing", *Energies*, 10(11): 1701, (2017).
- [8] Belman-Flores, J. M., Ledesma, S., Rodríguez-Valderrama, D. A., Hernández-Fusilier, A., "Energy optimization of a domestic refrigerator controlled by a fuzzy logic system using the status of the door", *International Journal of Refrigeration*, 104: 1-8, (2019).
- [9] Soetedjo, A., Nakhoda, Y. I., Saleh, C., "Embedded fuzzy logic controller and wireless communication for home energy management systems", *Electronics*, 7(9): 189, (2018).
- [10] Hemi H., Ghouili J., Cheriti, A., "A real time fuzzy logic power management strategy for fuel cell vehicle", *Energy Conversion and Management*, 80: 63-70, (2014).
- [11] Agarwal, D., Bharti, P. S., "A review on comparative analysis of path planning and collision avoidance algorithms", *International Scholarly and Scientific Research & Innovation*, 12(6), (2018).
- [12] Kontogiannis, D., Bargiotas D., Daskalopulu, A., "Fuzzy control system for smart energy management in residential buildings based on environmental data", *Energies*, 14(3): 752, (2021).
- [13] Yang, L., Qi, J., Song, D., Xiao, J., Han, J., Xia, Y., "Survey of robot 3D path planning algorithms", *Journal of Control Science and Engineering*, 1: 1-22, (2016).
- [14] Waqar, T., Demetgül, M., "Fuzzy logic controlled automatic vacuum cleaner", *Journal of Engineering and Technology Research*, 2(2): 93-100, (2014).
- [15] Mohanty, P. K., Parhi, D. R., "Optimal path planning for a mobile robot using cuckoo search algorithm", *Journal of Experimental & Theoretical Artificial Intelligence*, 28(1-2): 35-52, (2016).

- [16] Mohamed, A. Z., Lee, S. H., Hsu, H. Y., Nath, N., "A faster path planner using accelerated particle swarm optimization", *Artificial Life and Robotics*, 17: 233-240, (2012).
- [17] Juang, C. F., Chang, Y. C., "Evolutionary-group-based particle-swarm-optimized fuzzy controller with application to mobile-robot navigation in unknown environments", *IEEE Transactions on Fuzzy Systems*, 19: 379-391, (2011).
- [18] Sasaki, T., Enriquez, G., Miwa, T., Hashimoto, S., "Adaptive path planning for cleaning robots considering dust distribution", *Journal of Robotics and Mechatronics*, 30(1): 5-14, (2018).
- [19] Azir, K. N. F. K., Ong, R. J., "Low-cost autonomous robot cleaner using mapping algorithm based on Internet of Things (IoT)", *IOP Conference Series: Materials Science and Engineering*, 767(1): 012071, (2020).
- [20] Ravankar, A., Ravankar, A. A., Watanabe M., Hoshino Y., "An efficient algorithm for cleaning robots using vision sensors", *Proceedings*, 42(1): 45, (2020).
- [21] Ciliz, M. K., "Rule base reduction for knowledge-based fuzzy controllers with application to a vacuum cleaner", *Expert Systems with Applications*, 28: 175-184, (2005).
- [22] Le, A. V., Nhan, N. H. K., Mohan, R. E., "Evolutionary algorithm-based complete coverage path planning for tetriamond tiling robots", *Sensors*, 20(2): 445, (2020).
- [23] Song, P. C., Pan, J. S., Chu, S. C., "A parallel compact cuckoo search algorithm for three-dimensional path planning", *Applied Soft Computing Journal*, 94, 106443, (2020).
- [24] Oprea, S. V., Bâra, A., Preda, S., Tor, O. B., "A smart adaptive switching module architecture using fuzzy logic for an efficient integration of renewable energy sources: A case study of a RES System located in Hulubești, Romania", *Sustainability*, 12(15): 6084, (2020).
- [25] Yakoubi, M. A., Laskri, M. T., "The path planning of cleaner robot for coverage region using Genetic Algorithms", *Journal of Innovation in Digital Ecosystems* 3: 37-43, (2016).
- [26] Pulgord, G., "The Viterbi Algorithm", *IEE Seminar on Target Tracking: Algorithms and Applications*, (2006). DOI: <https://doi.org/10.1049/ic:20060556>
- [27] Mamdani, E. H., Assilian, S., "An experiment in linguistic synthesis with a fuzzy logic controller", *International Journal of Man-Machine Studies*, 7: 1-13, (1975).
- [28] Zimmermann, H. J., "Fuzzy Set Theory and Its Applications", Third Edition. Kluwer Academic Publishers, Boston/Dordrecht/London, (1996).
- [29] Ross, T. J., "Fuzzy Logic with Engineering Applications", Third Edition. John Wiley & Sons, Ltd., (2010).
- [30] Chen, Y. L., Wang S. C., Luo, Y. H., Lu, L. T., "Dynamic coordinate method for home return of cleaner robot", *Proceedings of the 2011 IEEE International Conference on Electric Information and Control Engineering (ICEICE)*, 5874-5877, (2011).