

REGULAR ARTICLE

Control and management of water quality for Nile tilapia fish in net tanks based on fuzzy modeling.

Vinicius Mastelini^{1,3}, Timóteo Ramos Queiroz^{1,4}, Luís Roberto Almeida Gabriel Filho^{1,4}, Mario Mollo Neto^{1,2}

¹Graduate Program in Agribusiness and Development, School of Science and Engineering, São Paulo State University - UNESP, Tupã, SP, Brazil.

²Department of Biosystems Engineering, School of Science and Engineering, São Paulo State University - UNESP, Tupã, SP, Brazil.

³University Center of Adamantina – UNIFAI, Adamantina, SP, Brazil.

⁴Department of Management, Development and Technology, School of Science and Engineering, São Paulo State University - UNESP, Tupã, SP, Brazil.

Regular Section

Academic Editor: Celso Antonio Goulart

Statements and Declarations

Data availability

All data will be shared if requested.

Institutional Review Board Statement

Not applicable.

Conflicts of interest

The authors declare no conflict of interest.

Funding

This study was financed by the National Council for Scientific and Technological Development (CNPq) for the research productivity grants awarded (Process #313339/2019-8 (MMN) and Proc. 315228/2020-2 (LRAG))

Author contribution

VM: Conceptualization, Experimental data collection, Data custody, Data analysis, Writing the manuscript; TRQ: Data analysis, Writing the manuscript, Manuscript Review; LRAG: Data analysis, Writing the manuscript, Manuscript Review; MMN: Data analysis, Writing the manuscript, Manuscript Review, Experimental data collection, Data analysis.

Abstract

Being classified as one of the most important species for fish farming, the Nile tilapia has seen a huge increase in breeding worldwide. As in any cultural medium, dealing with the quality of the medium in which it is grown guarantees a large part of the success of the process, being of equal importance, in this case, the quality of the water. Taking advantage of existing mathematical models, humans were able to measure and design best practices in virtually all areas, pointing to its great functionality, this article used the Fuzzy logic mathematical model together with Mamdani inference to analyze water quality scenarios and their consequences, various environments, variables, capable of directly affecting fish farming. The purpose was to use the MatLab scientific software to cross these variables with the possible output scenarios, facilitating the producer's decision-making. As a result of the research, it was possible to develop an algorithm to be embedded in a mobile application in the future with fuzzy mathematical modeling, with a Mamdani inference system for management and control of water quality in Nile Tilapia fish farming. The same will be made available to these breeders, since it has a structure of rules, aiming at the delivery of scientific information that collaborates with the best cultivation practices, improving production and profitability, through decision support to fish farmers.

Keywords

Mathematical Models; Decision Making; Measure; Design; MatLab.



This article is an open access, under a Creative Commons Attribution 4.0 International License.

Introduction

Used in a vast amount of fields of human activity, many of the existing practical problems require mathematical models, and sometimes, in very different situations, however, maintaining the same approach and philosophy (Sodré, 2007).

A mathematical model is capable of being presented through a representation of a real system, it must represent a system and how changes occur in it (Bassanezi, 1988; Bertalanffy, 1973; Sodré, 2007).

A model is a simplification of the real world or some way of working with it, however, the essential characteristics of this real-world must exist in this model, in such a way that it behaves in the same or similar way (Bassanezi, 1988; Bertalanffy, 1973; Sodré, 2007).

The main objective of a model is that it allows understanding of the model itself more simply (Bassanezi, 1988; Bertalanffy, 1973; Sodré, 2007).

Fuzzy logic is based on the Fuzzy sets' theory (Zadeh, 1988). It differs from other traditional logic systems due to its characteristics and details. In this type of logic, exact reasoning corresponds to a limiting case of approximate reasoning, interpreted as a process of composing fuzzy relations. The truth value of a proposition in Fuzzy logic can be a Fuzzy subset of any partially ordered set, different from binary logic systems, where the truth value assumes only two values: true (1) or false (0). Using multivalued logic systems, the truth value can be an element of a finite set, an interval, or even a Boolean algebra. In fuzzy logic (where Fuzzy fits), the truth values are expressed linguistically (true, very true, not true, false, very false), where each term is interpreted as a Fuzzy subset of the interval (Zadeh, 1988).

* Corresponding author

E-mail address: vinicius.mastelini@unesp.br (V. Mastelini).

According to Lee (1990), fuzzy modeling and control are techniques for rigorously handling qualitative information. Making inaccuracy and uncertainty powerful enough to manipulate knowledge. Dealing with the relationship between inputs and output, aggregating various process and control parameters.

The Fuzzification module is the stage in which the system inputs are and modeled by Fuzzy sets with their respective domains. It is at this point where the inherent need for specialists in the phenomenon to be modeled is justified (Souza, 2010).

In this stage of the Fuzzy inference module, each Fuzzy proposition is translated “mathematically” through the techniques of Fuzzy logic, providing the Fuzzy output (control) to be adopted by the controller, from each Fuzzy input (Souza, 2010).

The Mamdani inference method proposes a binary relation M between x and u to mathematically model the rule base. Based on the max-min inference composition rule (Souza, 2010).

Belonging to the cichlid family, the Nile tilapia, *O. Niloticus*, originates from the Nile River basin, in East Africa and is widely disseminated in tropical and subtropical regions, such as Israel, Southeast Asia, and the American continent (Carvalho, 2006).

Considered one of the fish with the greatest potential for aquaculture, tilapia has several essential characteristics: it is precocious, fast-growing, has a large and varied food menu, has the physiological capacity to adapt to different environments and production systems, is disease-resistant, tasty meat with low fat and calories and high fillet yield, making it extremely suitable for industrialization and high commercial value (Castagnolli, 1992; Schimittou, 1995; Ono & Kubitz, 2003; Zimmermann & Fitzsimmons, 2004; Cyrino & Conte, 2006).

Raising fish, first of all, is “creating water”, so great is the interaction of this factor. To achieve success in raising fish through any system, it is essential to take into account the physical and chemical factors of the water, the main ones being: temperature; Dissolved oxygen; hydrogen potential – pH; transparency; ammonia, and nitrite (Codevasf, 2019).

Based on the context described above, this article seeks to develop an algorithm embedded in Fuzzy mathematical modeling and the Mamdani inference system for management and control of water quality in Nile Tilapia fish farming, which may later have its logic and results passed on for a specialist mobile application that will be made available to these breeders, aiming at the delivery of scientific information that can collaborate with the best cultivation practices, thus also improving their production and profitability.

Materials and methods

The first step was the definition of the input variables, based on the limits determined by the systematic review of the literature, limits that were adapted, codified, and treated, allowing better adaptation and assertiveness in the logic of the algorithm for inclusion in the tool (Mastelini & Mollo Neto 2022).

Of the many variables that influence water quality identified, five of the most relevant were selected, according to the authors (Codevasf, 2019; Coleção Senar 208, 2018; Ematur-DF, 2009), these, as well as their limits and their adapted codes are presented in the results topic. The second step was the verification of all possibilities of existing scenarios among all environment variables and their status. Multiplying the five variables with the total status that each one could assume within the scenarios; we arrive at a total of 432 possible situations ($4 \times 3 \times 4 \times 3 \times 3$). These scenarios were identified with the help of the Microsoft Office Excel tool.

For greater precision and assertiveness in this step, an algorithm was developed in the C++ programming language capable of relating the status of each variable, taking care to never repeat the same code and/or variable within the same scenario.

The report resulting from this algorithm, as well as from the spreadsheet of the Microsoft Office Excel tool, made it possible to identify the 432 possible scenarios between the relationship of the variables and their status.

The third step was to compare the possible scenarios (defined with the scenario combination algorithm) with the output variables, which were created according to the water quality and the linguistic terms used for the final classification of the generated algorithm: “terrible, bad, good and great”.

Using the Microsoft Office Excel tool in the table with the scenarios resulting from the algorithm, a new field was added, called the output variable: linguistic term, which, in turn, suffered the insertion of several conditional structures (IF/ELSE). The Microsoft Office Excel tool has great programming/development power and is also widely used for this purpose.

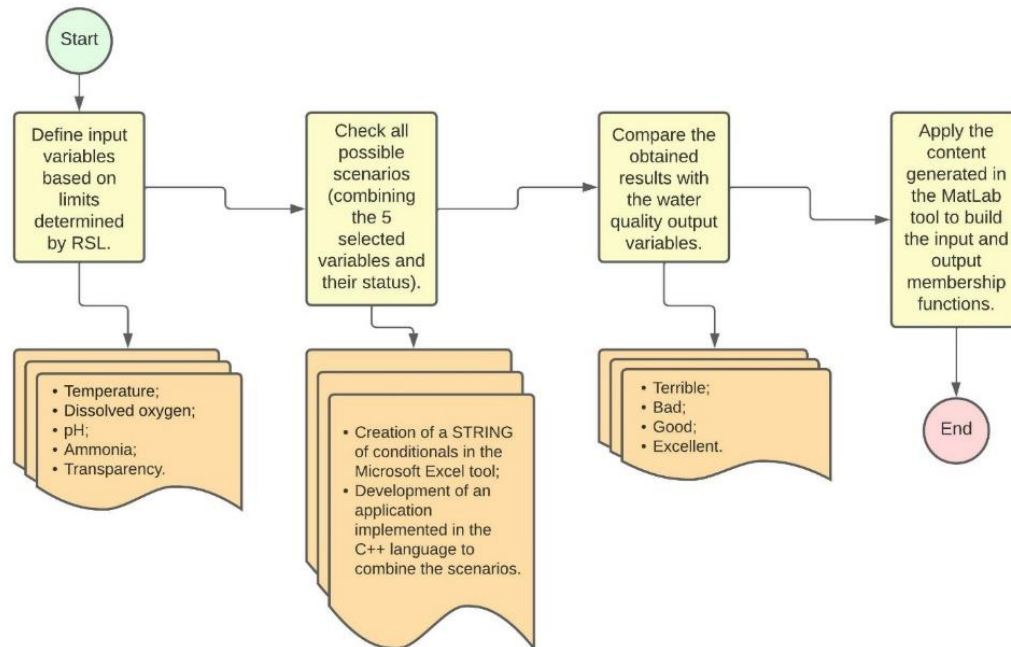
The purpose of this STRING was to combine the statuses of the scenarios previously created in the previous step with the linguistic term for water quality output variables, such as: terrible, bad, good, and excellent. For example: if scenario “X” has any of the input environment variables with the status “LETHAL”, its output variable would automatically receive the linguistic term “VERY BAD”; if scenario “Y” has a critical input environment variable, with its status “ALERT”, its output variable would automatically receive the linguistic term “BAD”; if scenario “Z” has a non-critical input environment variable, with “ALERT” status, its output variable would automatically receive the linguistic term “GOOD”; if scenario “W” has all its input environment variables with status “IDEAL”, its output variable would automatically receive the linguistic term “GREAT”. The complete STRING used in this application is available in the results topic.

To validate the results obtained with the application of conditional structures in the Microsoft Office Excel tool, a new algorithm was created, and implemented in the C++ programming language, also using programming logic and the application of conditional structures. The logic of the conditionals remained the same as applied in the Excel tool, however, they were only adapted to the programming language.

Finally, all the previously generated content in the MatLab tool was applied to build the input membership functions and construct the output membership function of water quality and

its defined classification levels. It is possible to visualize the methods described above by the flow diagram available in Figure 1.

Figure 1. Diagram representing the Flow of methods.



By default, the tool creates the project with only one input variable and one output variable (with four possible situations), which was later added with four more input variables, totaling five.

After creating the input variables in the MatLab tool, it was time to create and configure the output variable (with four different situations).

With the input and output pertinence definitions configured, the rules were then created in MatLab, the Fuzzy logic as well and the Mamdani inference.

The rules applied in the tool also follow the same ones applied in the previous step (Microsoft Excel tool and C++ algorithm).

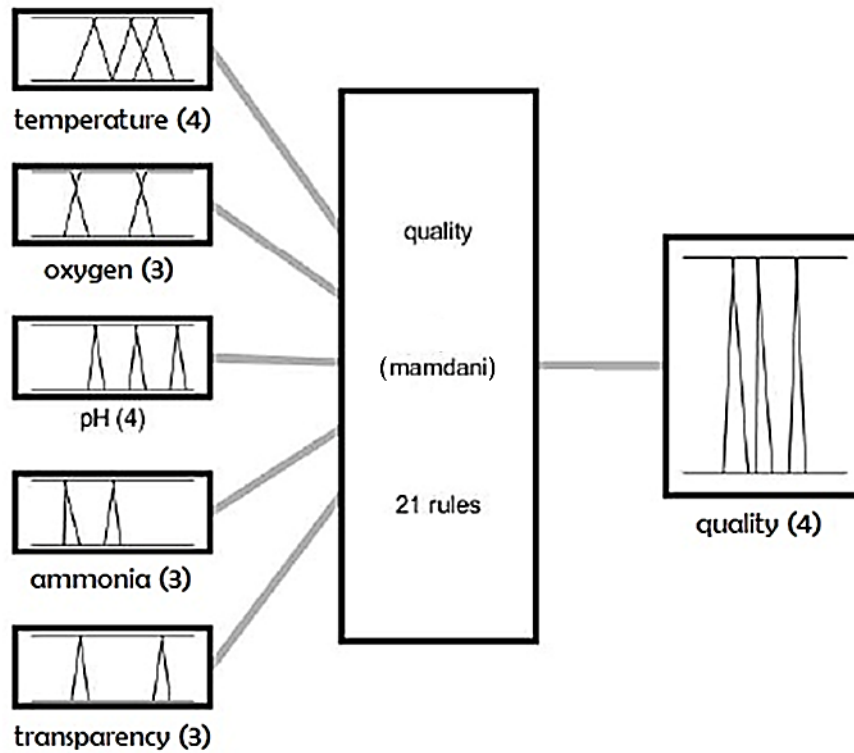
Then, in the next section, with the results obtained from triggering the rules for all the scenarios obtained from the input variables, it was possible to carry out the defuzzification by obtaining numerical values for the outputs of each of the

scenarios and constructing the respective surfaces generated by the interactions between input and output variables.

Results and discussion

As described in the topic materials and methods, in the first stage, the definition of the variables of relevance, their limit values, and expected results were adapted from the authors (Codevasf, 2019; Coleção Senar 208, 2018; Ematur-DF, 2009) and organized in tables, codes for these expected results were also defined, enabling their maintenance by algorithms within the system.

Once the steps were completed, the Fuzzy mathematical modeling scheme with Mamdani inference was completed, generating the project scheme, visible in Figure 2.



System quality: 5 inputs, 1 outputs, 21 rules

Figure 2. Fuzzy modeling scheme with Mamdani inference.

To deal with the temperature variable, four different linguistic terms or “status” were created: lethal, alert 1, ideal,

and alert 2, each with its respective code and limit value, as shown in Table 1.

Table 1. Temperature variable and its limits.

CODE	DESCRIPTION	LIMIT VALUE
101	Lethal	≤ 15
102	Alert 1	$\geq 16 \ \&\& \ < 25$
103	Ideal	$\geq 25 \ \&\& \ < 31$
104	Alert 2	≥ 31

Source: Adapted by the Authors of Codevasf (2019); Coleção Senar 208 (2018) and Ematur-DF (2009).

These results in Table 1 allowed us to construct the input membership functions for the temperature variable, which is shown in Figure 3.

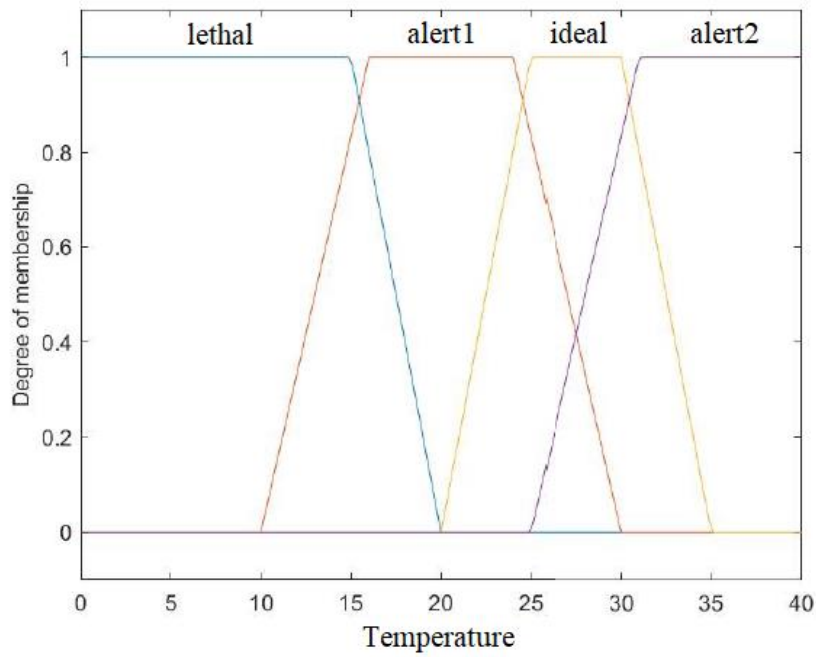


Figure 3. MatLab temperature input variable.

As for the dissolved oxygen variable, three different linguistic terms or “status” were created: lethal, ideal, and alert, each with its respective code and limit value, as shown in Table 2.

Table 2. Dissolved oxygen variable and its limits.

CODE	DESCRIPTION	LIMIT VALUE
201	Lethal	< 3
202	Ideal	>= 3 && < 7
203	Alert	>= 7

Source: Adapted by the Authors from Codevasf (2019); Coleção Senar 208 (2018) and Ematur-DF (2009).

These results from Table 2 allowed the construction of the input membership functions for the dissolved oxygen variable, which is shown in Figure 4.

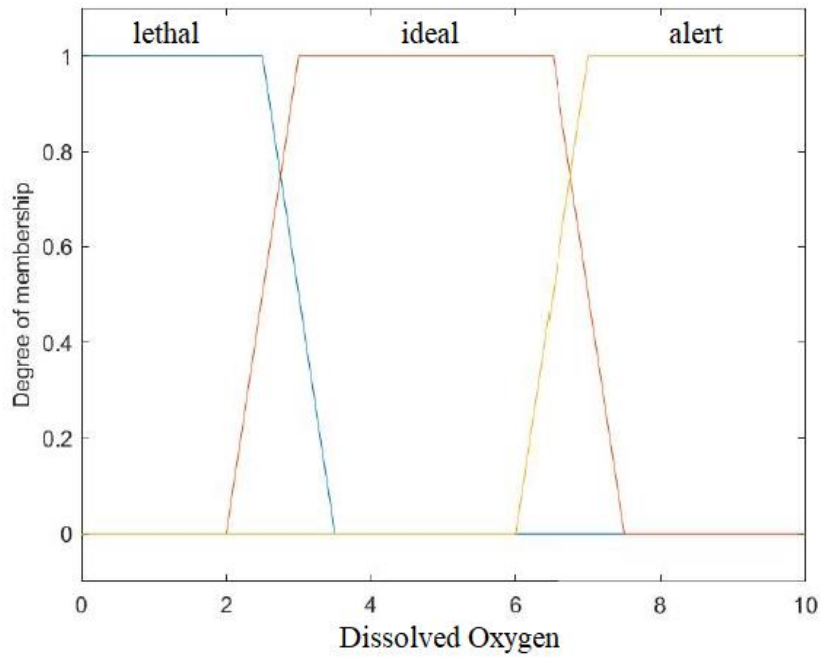


Figure 4. MatLab dissolved oxygen input variable.

The pH variable had the creation of four different linguistic terms or “status”: Lethal 1, alert 1, ideal, and lethal 2, each with its respective code and limit value, as shown in Table 3.

These results from Table 3 allowed the construction of the input membership functions for the pH variable, which is shown in Figure 5.

Table 3. Variable pH and its limits.

CODE	DESCRIPTION	LIMIT VALUE
301	Lethal 1	< 4
302	Alert 1	>= 4 && < 6.5
303	Ideal	>= 6.5 && < 9
304	Lethal 2	>= 9

Source: Adapted by the Authors from Codevasf (2019); Coleção Senar 208 (2018) and Ematur-DF (2009).

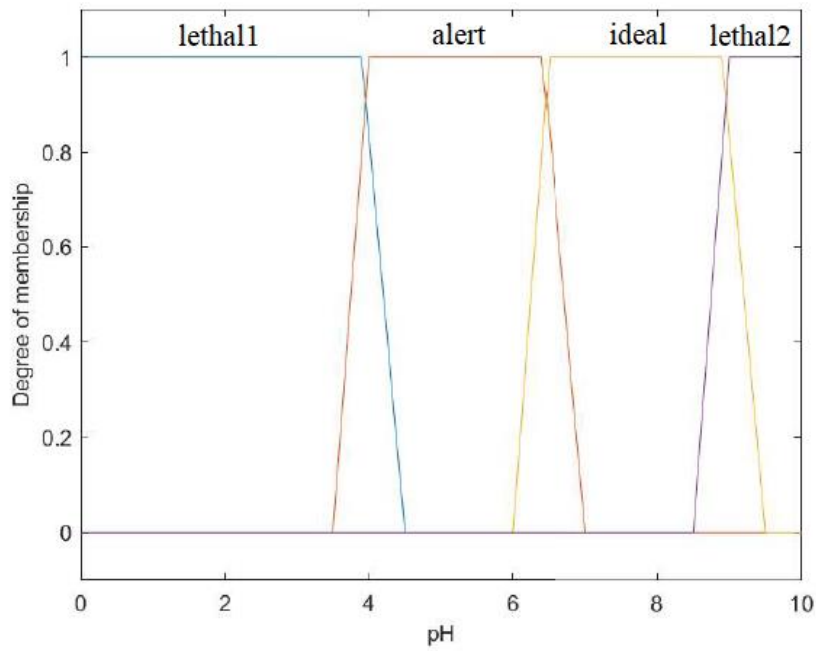


Figure 5. MatLab pH input variable.

The statuses created for the ammonia variable were three linguistic terms or “status”: alert, ideal, and lethal, each with its respective code and limit value, as shown in Table 4.

Table 4. Variable ammonia and its limits.

CODE	DESCRIPTION	LIMIT VALUE
401	Alert	≤ 0.2
402	Ideal	$> 0.2 \ \&\& \ \leq 0.5$
403	Lethal	> 0.5

Source: Adapted by the Author from Codevasf (2019); Coleção Senar 208 (2018) and Ematur-DF (2009).

These results from Table 4 allowed the construction of the entry membership functions for the ammonia variable, which is shown in Figure 6.

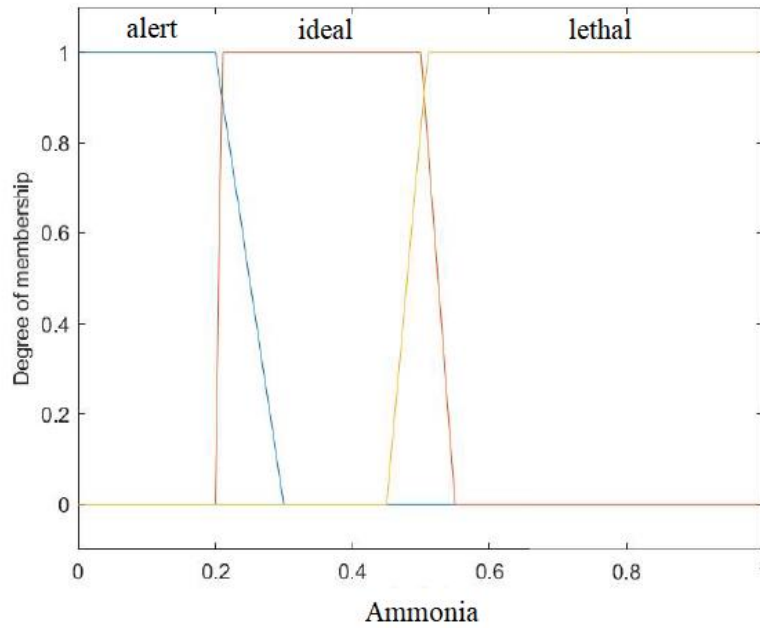


Figure 6. MatLab ammonia input variable.

To deal with the transparency variable, three different linguistic terms or “status” were created: high, average, and low, each with their respective codes and threshold values, as shown in Table 5.

Table 5. Variable transparency and its limits.

CODE	DESCRIPTION	LIMIT VALUE
501	High	$\geq 0 \ \&\& \ < 61$
502	Average	$\geq 61 \ \&\& \ < 161$
503	Low	> 161

Source: Adapted by the Author from Codevasf (2019); Coleção Senar 208 (2018) and Ematur-DF (2009).

These results from Table 5 allowed the construction of the input membership functions for the transparency variable, which is shown in Figure 7.

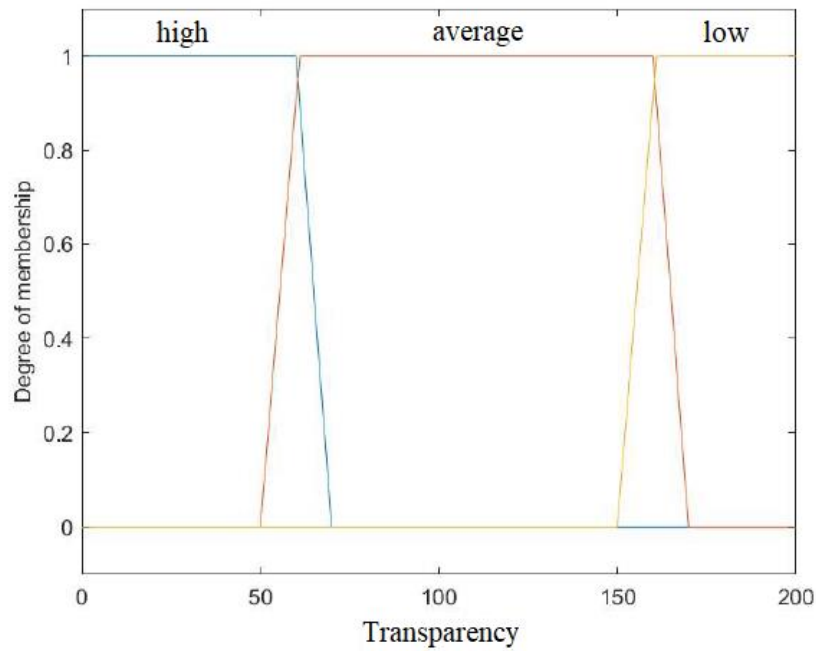


Figure 7. MatLab transparency input variable.

Next, the output membership function was modeled (Figure 8), which reflects the experts' indications, with responses for each of the projected scenarios found in the literature (Codevasf (2019); Coleção Senar 208 (2018) and Ematur -DF (2009)).

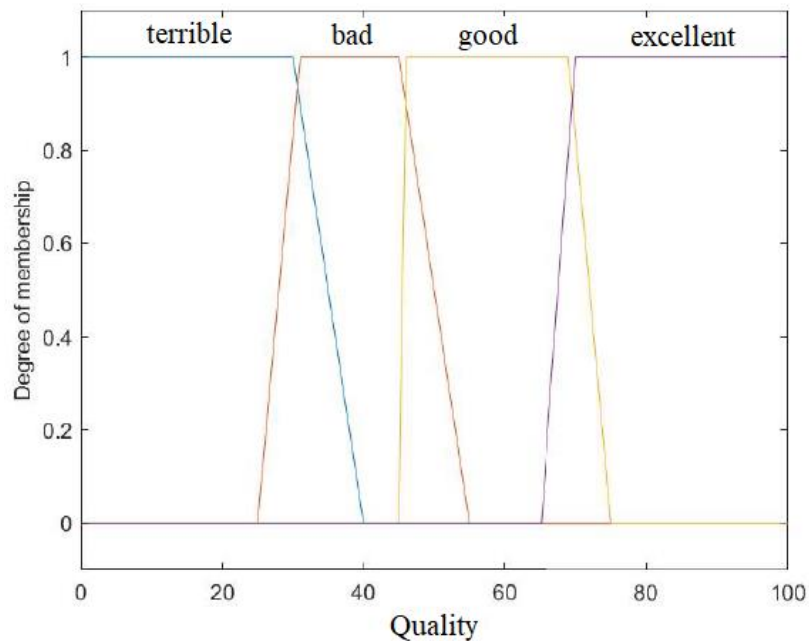


Figure 8. MatLab output (Quality) variable.

Source: Adapted by the Author from Codevasf (2019); Coleção Senar 208 (2018) and Ematur-DF (2009).

In the identification stage of possible scenarios arising from the relationship between variables and status, an algorithm developed in C++ programming language was used, visualized in the Algorithm 1.

Algorithm 1. Identification of scenarios from the relationship between variables and status.

```

1.  int main()
2.  {
3.      int num1, num2, num3, num4, num5;
4.
5.      for (int a = 101; a <= 104; a++)
6.      {
7.          num1 = a;
8.
9.          for (int b = 201; b <= 203; b++)
10.         {
11.             num2 = b;
12.
13.             for (int c = 301; c <= 304; c++)
14.             {
15.                 num3 = c;
16.
17.                 for (int d = 401; d <= 403; d++)
18.                 {
19.                     num4 = d;
20.
21.                     for (int e = 501; e <= 503; e++)
22.                     {
23.                         num5 = e;
24.
25.                         cout <<num1 <<" " <<num2 <<" " <<num3 <<" " <<num4 <<" " <<num5 << "\n";
26.                     }
27.                 }
28.             }
29.         }
30.     }
31. }

```

The resulting scenarios obtained from the execution of this algorithm show the combination of 432 possible scenarios among the environment variables and their status. The next step was the definition of the output variables with their respective linguistic terms of water quality: terrible, bad, good, and great, which were initially organized in the Microsoft Office Excel tool, with the insertion of a conditional logic STRING, capable of to verify the status of each variable of each scenario and correlate it with its proper linguistic term. The STRING can be seen below (String 1).

String 1. Maintenance of output variables and their linguistic terms.=SE(OU(B4=101;C4=201;D4=301;D4=304;E4=403);"Terrible";SE(OU(B4=102;B4=104;E(OU(C4=203)));"Bad";SE(E(B4=103;C4=202;E(OU(D4=302;D4=304;E(OU(E4=401;E(OU(F4=502;F4=503)))))));"Good";SE(E(B4=103;C4=202;D4=303;E4=402;F4=501;F4=501);"Excellent"))))

Where:

• IF (temperature, oxygen, pH, or ammonia equal to lethal 1 or lethal 2) = "Poor Quality"

• IF (temperature equals alert 1 or alert 2 and oxygen equals alert) = "Poor Quality"

• IF (temperature and oxygen equal to ideal and pH and ammonia different from lethal and transparency equal to high or medium or low) = "Good Quality"

• IF (temperature and oxygen and pH and ammonia equal to ideal and transparency equal to high) = "Excellent Quality"

After the application of the conditional logic STRING carried out in the Microsoft Office Excel tool, the same conditional logic was applied in a new algorithm developed in

C++ language, to validate the output variables and their linguistic terms, this algorithm can be seen below, in the Algorithm 2.

Algorithm 2. Maintenance of output variables and their linguistic terms.

```

1.  int main()
2.  {
3.      int counter=0, counter1, counter2, counter3, counter4, counter5, terrible=0, bad=0, good=0, excellent=0, other=0;
4.      int codTemp, codOxi, codPh, codAmon, codTransp, WaterQuality;
5.
6.      for (counter1 = 101; counter1 < 105; counter1++)
7.      {
8.          for (counter2 = 201; counter2 < 204; counter2++)
9.          {
10.             for (counter3 = 301; counter3 < 305; counter3++)
11.             {
12.                 for (counter4 = 401; counter4 < 404; counter4++)
13.                 {
14.                     for (counter5 = 501; counter5 < 504; counter5++)
15.                     {
16.                         codTemp = counter1;
17.                         codOxi = counter2;
18.                         codPh = counter3;
19.                         codAmon = counter4;
20.                         codTransp = counter5;
21.                         counter++;
22.                         WaterQuality = -1;
23.
24.                         //case terrible
25.                         if ((codTemp == 101) || (codOxi == 201) || (codPh == 301) || (codPh
26. == 304) || (codAmon == 403))
27.                         {
28.                             WaterQuality = 0;
29.                         }
30.
31.                         //case bad
32.                         else if ((codTemp == 102 || codTemp == 104) && (codOxi == 202 ||
33. codOxi == 203) && (codPh == 302 || codPh == 303) && (codAmon != 403) && (codTransp == 501 || codTransp == 502 ||
34. codTransp == 503))
35.                         {
36.                             WaterQuality = 1;
37.                         }

```

```

36. //case good
37. else if ((codTemp == 103 && codOxi == 202) && (codPh == 302 ||
codPh == 303) && (codAmon != 403) && (codTransp == 501 || codTransp == 502 || codTransp == 503))
38. {
39.     WaterQuality = 2;
40. }
41.
42. //case bad (2° option)
43. else if ((codTemp == 103 && codOxi == 203))
44. {
45.     WaterQuality = 1;
46. }
47.
48. //case excellent
49. if ((codTemp == 103) && (codOxi == 202) && (codPh == 303) &&
(codAmon == 402) && (codTransp == 501))
50. {
51.     WaterQuality = 3;
52. }
53.
54. cout << "\n" << codTemp << " - " << codOxi << " - " << codPh << "
- " << codAmon << " - " << codTransp << " = ";
55.
56. switch (WaterQuality)
57. {
58.     case 0:
59.         terrible++;
60.         cout << "Terrible!";
61.         break;
62.     case 1:
63.         bad++;
64.         cout << "Bad!";
65.         break;
66.     case 2:
67.         good++;
68.         cout << "Good!";
69.         break;
70.     case 3:
71.         excellent++;
72.         cout << "Excellent!";
73.         break;
74.     case -1:

```

```

75.                                     other++;
76.                                     cout << "----- Other! -----";
77.                                     break;
78.                                     }
79.                                 }
80.                                 }
81.                             }
82.                             }
83.                         }
84.
85.         cout << "\n\nAmount of comparisons performed: " << counter;
86.         cout << "\nAmount Terrible: " << terrible;
87.         cout << "\nAmount Bad: " << bad;
88.         cout << "\nAmount Good: " << good;
89.         cout << "\nAmount Excellent: " << excellent;
90.         cout << "\nAmount Other: " << other << "\n\n";
91.
92.         system("pause");
93.     }

```

With the application of the conditional structure on the results of the water quality scenarios, both applied STRINGS (Excel and algorithm), brought the same output values, defined below:

- 360 scenarios with bad quality status;
- 60 scenarios with bad quality status;
- 11 scenarios with good quality status;
- 1 scenario with optimal quality status.

Finally, after inserting the previous procedures in the MatLab tool and defining the rules, available below, it was possible to perform the analysis.

- If lethal temperature, or lethal oxygen, or lethal pH 1, or lethal ammonia
 - o Bad quality;
- If lethal temperature, or lethal oxygen, or lethal pH 2, or lethal ammonia
 - o Bad quality;
- If ideal temperature, and ideal oxygen, and ideal pH, and ideal ammonia, and high transparency
 - o Great quality;
- If ideal temperature, and ideal oxygen, and non-lethal pH 1, and non-lethal ammonia, and high, or medium, or low transparency

- o Good quality;
 - If ideal temperature, and ideal oxygen, and non-lethal pH 2, and non-lethal ammonia, and high, or medium, or low transparency
- o Good quality;
 - If temperature alert 1, and oxygen alert, and pH non-lethal 1, and ammonia non-lethal, and transparency high, or medium, or low
- o Bad quality;
 - If temperature alert 1, and oxygen alert, and non-lethal pH 2, and non-lethal ammonia, and high, or medium, or low transparency
- o Bad quality;
 - If temperature alert 2, and oxygen alert, and non-lethal pH 1, and non-lethal ammonia, and high, or medium, or low transparency
- o Bad quality;
 - If temperature alert 2, and oxygen alert, and non-lethal pH 2, and non-lethal ammonia, and high, or medium, or low transparency
- o Bad quality;

With the schema assembled and finalized, it was possible to visualize the 3D surface of the input variables and their correlations, as shown in Figures 9 and 10.

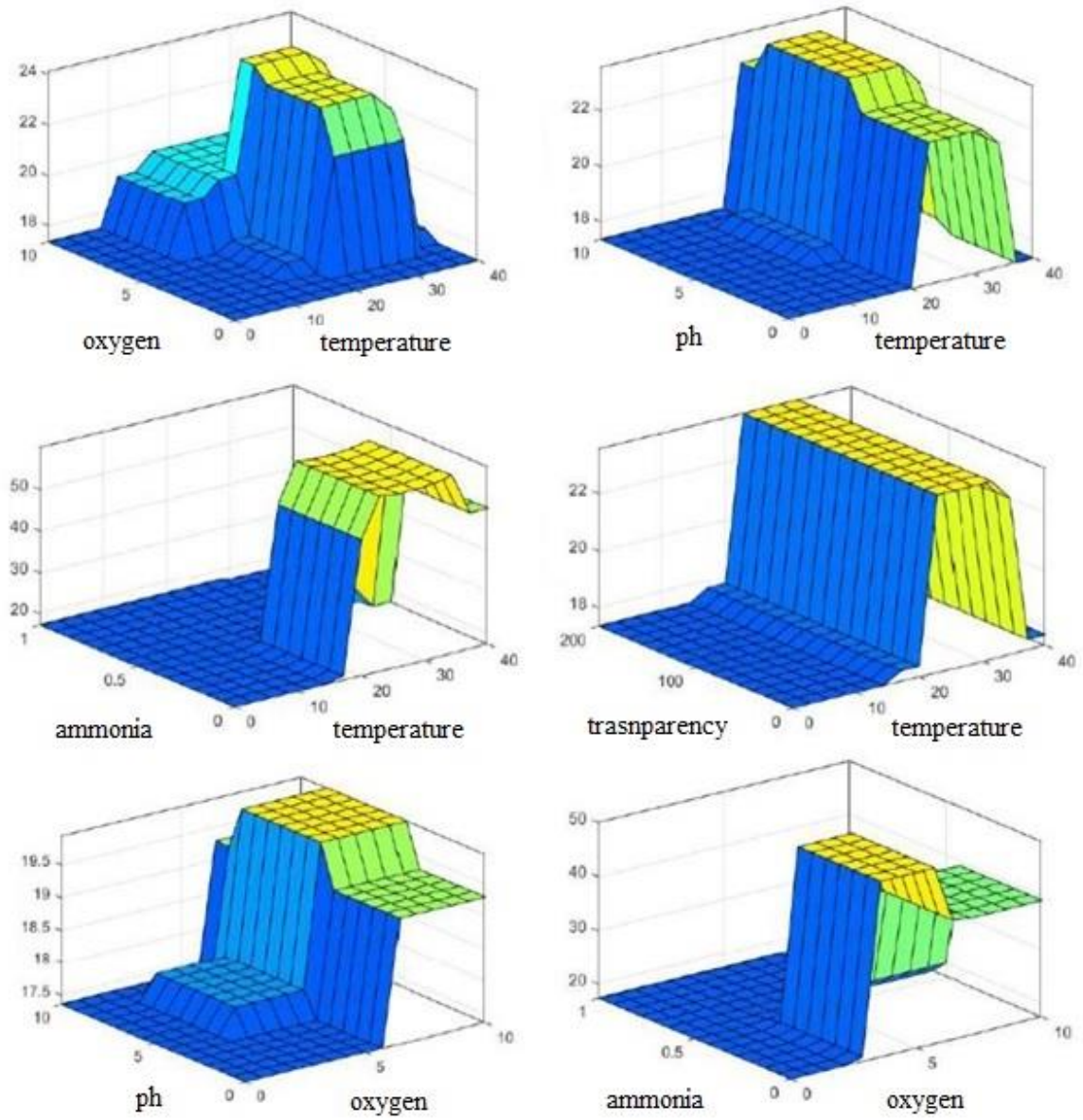


Figure 9. 3D Surfaces – part 01.

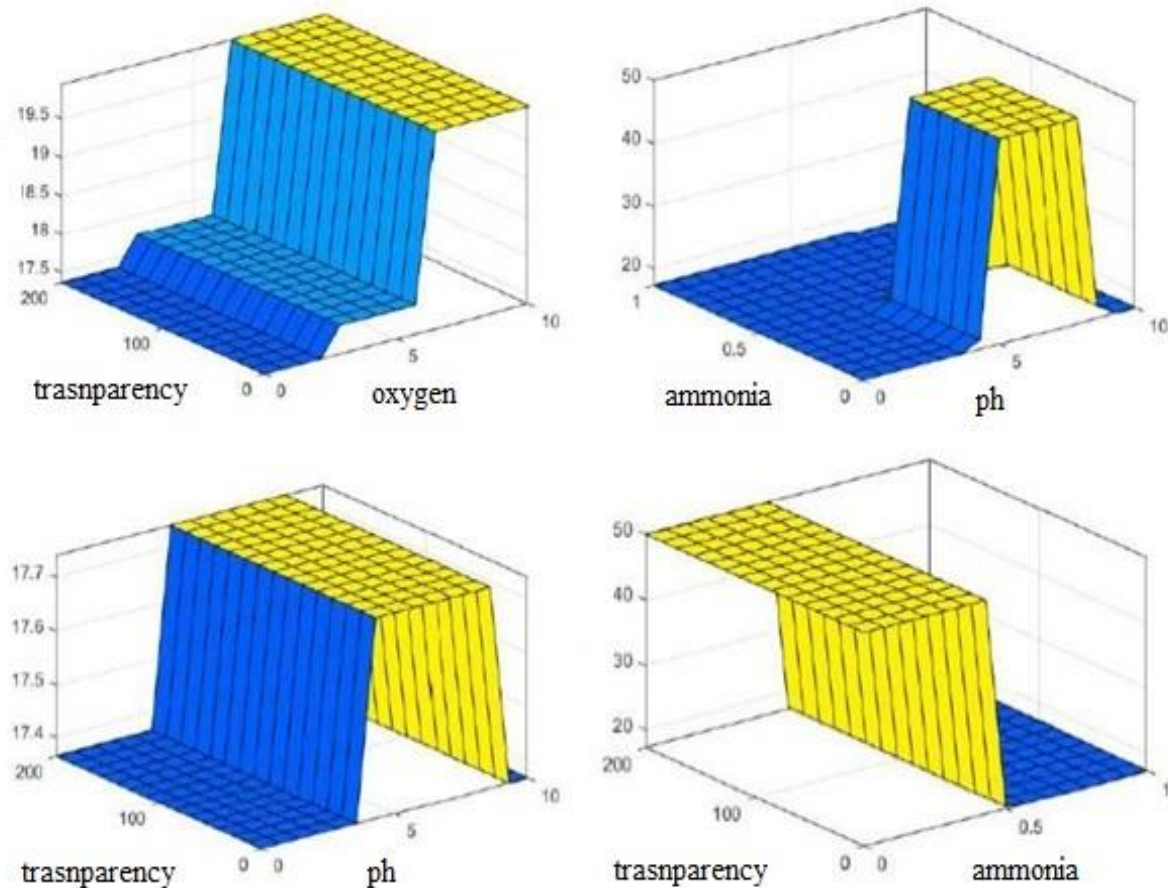


Figure 10. 3D surfaces – part 02.

It can be seen in the graphs of the response surfaces of the generated algorithm, Figures 9 and 10, that whenever any confrontation with ammonia appears, the quality axis rises to 50 points, such as the confrontations of the variables temperature x ammonia, oxygen x ammonia, pH x ammonia and ammonia x transparency, indicating that it is very important to strictly control this variable, as it can be deadly in high concentrations.

In the case of comparisons between the variables temperature x oxygen, temperature x PH, and temperature x transparency, we can see that the quality ranges from surfaces to a level of 22 to 24 points, which indicates that temperature is also an important variable to be taken into account. considered in the strictest control, but it depends on climatic conditions and is more difficult to control in the reservoir, which presents very slow thermal drift, requiring very large investments for mitigation, which leads to trying to mitigate, for example, the case of pH variable, oxygen Variable, and transparency variable, more economically viable to control through the use of pH control chemicals, oxygenators and raising the reservoir water exchange flow to eliminate transparency.

It is observed that in the case of the variables oxygen x pH and oxygen x transparency., the emphasis goes to oxygen,

which is a factor to be controlled by employing aerators since its range of interference with quality is around 19, 5 to 20 points.

Finally, the variables pH x transparency., toured the quality in the range of 17.5 to 18 points, leading us to interpret that both the pH and the transparency have less weight in the classification of water quality, but they cannot cease to be controlled, with the suggestion being to chemically control the pH and increase the water exchange flow in the reservoir to eliminate transparency issues.

It is still possible to highlight the importance attributed by the arrangement of the rules to the pH variable because it has two lethal statuses, followed later by the temperature variable, the oxygen Variable, and the ammonia variable, with one lethal status each, and finally, transparency, which does not have lethal status, which is considered the one that least impacts water quality in the constructed algorithm.

With all the parameterization of the input and output variables, with the surfaces and rules assembled, Figures 11, 12, 13, and 14 show the basis of Fuzzy modeling and Mamdani inference in action, in their respective results: poor quality, bad quality, good quality, and great quality.

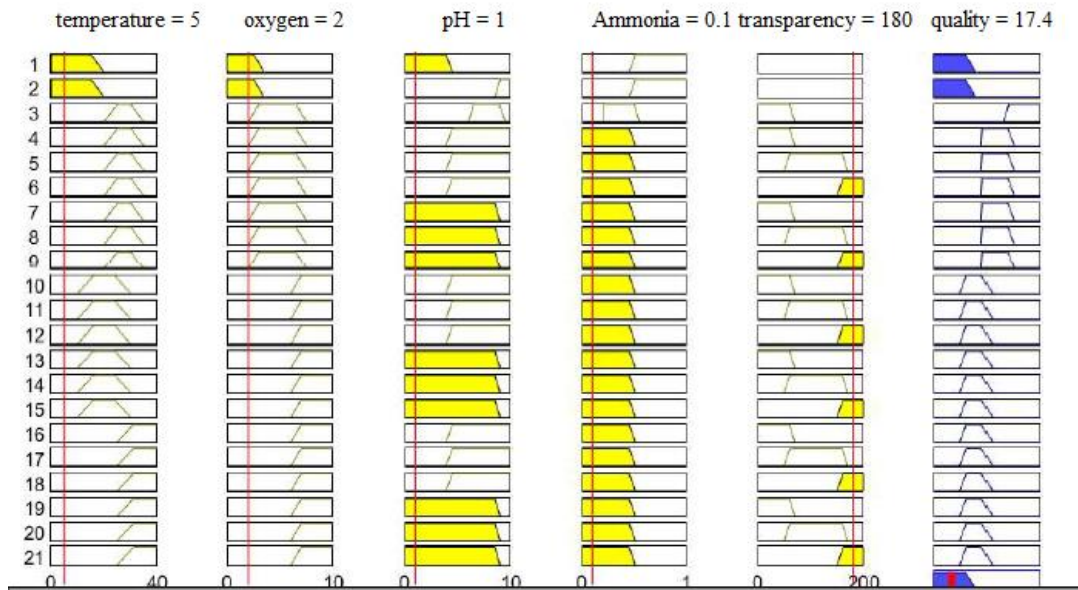


Figure 11. Rules surface – poor water quality.

As seen in the quality column of Figure 11, the score of 17.4 responds to the linguistic term “poor water quality”.

Figure 12 shows, in the quality column, the score of 39.1, which response to the linguistic term “poor water quality”.

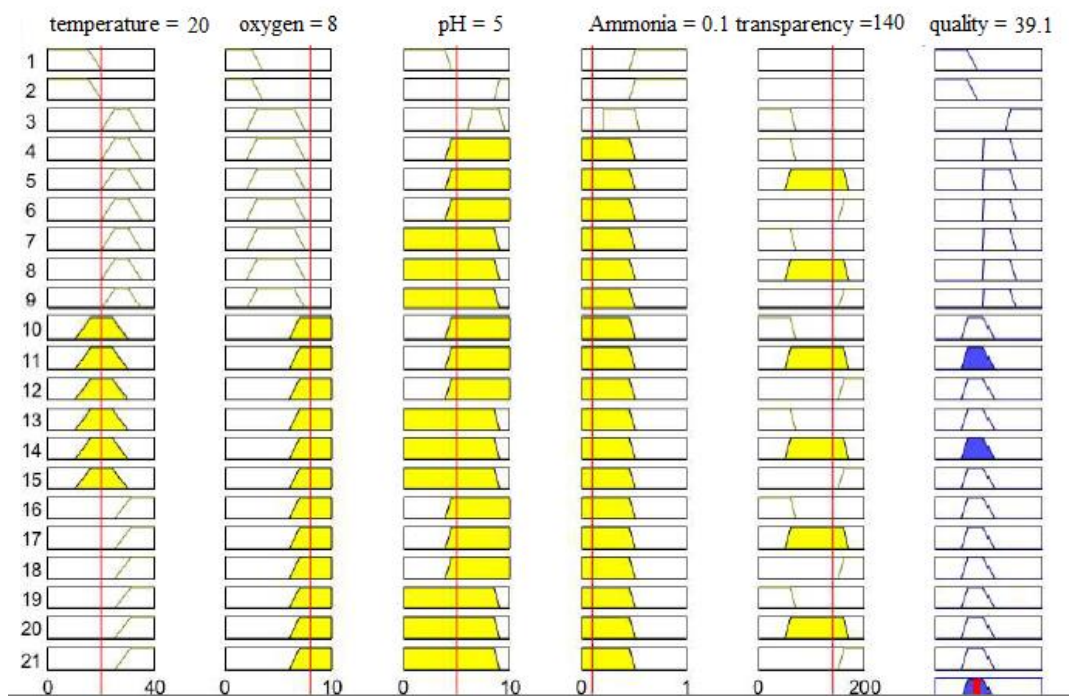


Figure 12. Rule surface – poor water quality.

As seen in Figure 13, quality column, the score of 58.8 responds to the linguistic term “good water quality”.

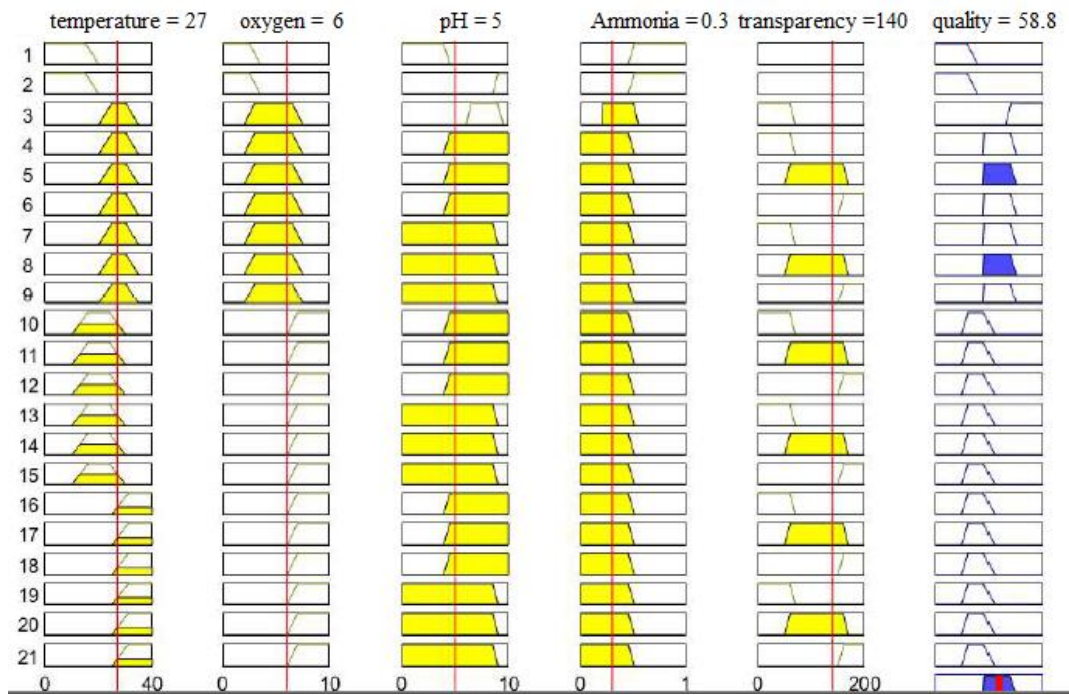


Figure 13. Rule surface – good water quality.

It is possible to observe in Figure 14 that, in the column destined to quality, the value 73.0 represents the linguistic term “great water quality”.

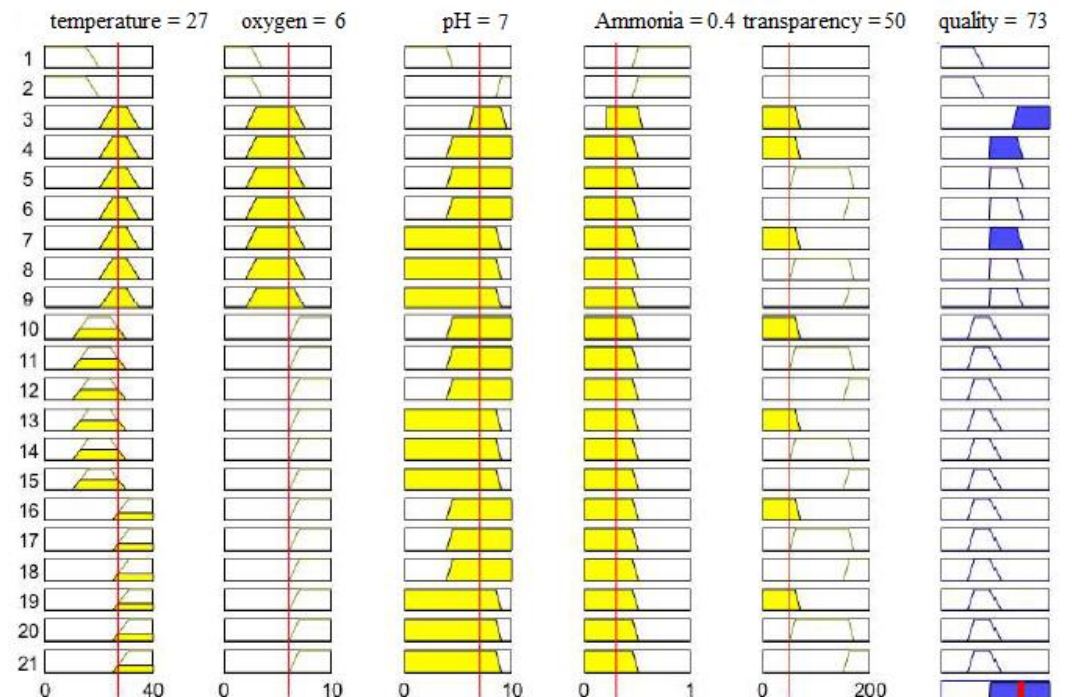


Figure 14. Rules surface – optimal water quality.

With the completion of the Fuzzy algorithm, it was possible to observe that water quality, as well as its many variables and different statuses, can generate several different scenarios, each with its characteristics, which, among those analyzed in this article, showed that, within of 432 possible scenarios, only one of them was considered optimal for fish farming of the tropical fish species Nile Tilapia.

Conclusions

With the results obtained, it was possible, in this article, to develop an algorithm to be embedded in a mobile application in the future with fuzzy mathematical modeling, with a Mamdani inference system for the management and control of water quality in Nile Tilapia fish farming, later, having its logic and results passed on to a specialist mobile application that will be made available to these creators, due to its portability to the programming language, since it has a structure of rules, aiming at the delivery of scientific information that can collaborate with the best practices of cultivation, thus also improving their production and profitability, through decision support for fish farmers.

Acknowledgments

The authors wish to acknowledge the Postgraduate Program in Agribusiness and Development (PGAD) of the School of Sciences and Engineering of São Paulo State University (UNESP), the Brazilian National Council for Scientific and Technological Development (CNPq), for support (Process #313339/2019-8 (MMN)).

References

- Bassanezi, R. C. & Ferreira Jr., W. C. (1988). Equações Diferenciais com aplicações, Editora HarbraLtda., São Paulo.
- Bertalanffy, L. (1973). Teoria Geral dos Sistemas, Editora Vozes, Petrópolis.
- Carvalho, E.D. (2006). Avaliação dos impactos da piscicultura em tanques-rede nas represas dos grandes tributários do alto Paraná (Tietê e Paranapanema): o pescado, a ictiofauna agregada e as condições limnológicas. Relatório Científico (FAPESP). Botucatu, SP. 46p.
- Castagnolli, N. (1992). Criação de peixes de água doce. Jaboticabal: FUNEP. 189p.
- Codevasf. (2019). Manual de Criação de Peixes em Tanques-Rede. Ministério do Desenvolvimento Regional. 3º ed. Brasília – DF.
- Coleção Senar 208. (2018). Piscicultura: Criação de Tilápias em Tanques-Rede. Serviço Nacional de Aprendizagem Rural. Brasília – DF.
- Cyrino, J.E. & Conte, L. (2006). Tilapicultura em Gaiolas: produção e economia. In: José Eurico Possebon Cyrino e Elisabeth Criscuolo Urbinati (Eds.). AquaCiência: Tópicos Especiais em Biologia Aquática e Aqüicultura. Jaboticabal: Sociedade Brasileira de Aqüicultura e Biologia Aquática, cap.12, p.151-171.
- Emater-DF. (2009). Criação de Tilápias. Governo do Distrito Federal. 2º ed. Brasília – DF.
- Hughes, S.G. (1993). All-vegetable protein feeds. Feed International, v.14, p.55-60.
- Lee, C.C (1990). Fuzzy Logic in Control Systems: Fuzzy Logic Controller, parts I and II. IEEE Trans. on Systems, Man, and Cybernetics, vol. 20, pp 404-435.
- Mastelini, V., & Mollo Neto, M. (2022). INDICADORES DE QUALIDADE DA ÁGUA PARA CRIAÇÃO DE TILÁPIAS-DO-NILO EM TANQUE-REDE: UMA REVISÃO DAS PRÁTICAS DE ANÁLISES DE CRIAÇÃO (2010 – 2021). RECIMA21 - Revista Científica Multidisciplinar - ISSN 2675-6218, 3(12), e3122363. <https://doi.org/10.47820/recima21.v3i12.2363>
- Ono, E. A. & Kubitza, F. (2003). Cultivo de peixes em tanques-rede. 3ªed. Jundiá: Eduardo A. Ono, 112p.
- Schmittou, H.R. (2003). Produção de peixes em alta densidade em tanques-rede de pequeno volume. Tradução de Eduardo Ono. ASA - Associação Americana de Soja. Editado por Silvio Romero Coelho, Mogiana Alimentos S.A., 1995, 78p. ONO, E. A.; KUBITZA, F. Cultivo de peixes em tanques-rede. 3ªed. Jundiá: Eduardo A. Ono, 112p.
- Sodré, U. (2007). Modelos Matemáticos. UEL, Londrina/PR.

Souza, O. N. (2010). Introdução à Teoria dos Conjuntos Fuzzy. Universidade Estadual de Londrina - Centro de Ciências Exatas - Departamento de Matemática, Londrina/PR.

Zadeh, L. (1988) - Fuzzy Logic - IEEE Computer, April, pp. 83-92.

Zimmermann, S. & Fitzsimmons, K. (2004). Tilapicultura intensiva. In: José Eurico Possebon Cyrino, Elisabeth Criscuolo Urbinati, Débora Machado Fracalosi, Newton Castagnolli (Editores), Tópicos especiais em piscicultura de água doce tropical intensiva, São Paulo: TecArt, Cap.9, p. 239-266.