

Évf. 6 szám 1 (2024): Bánki Közlemények/Bánki Reports Spring (2024)

Megjelent: 2024-02-27

Technical Informatics (Műszaki Informatika)

Fuzzy Logic in Drone Control

Anja Stijepović, Una Sredovic

 PDF (English)

Útvonaltervezés heurisztikus módszerekkel –a hangyakolónia algoritmus

Gabriella Polyak, Miklós Póth

 PDF (English)

Mathematical models of tumor growth

Sara Medojević

 PDF (English)

Fuzzy Logic - Based control for autonomous vehicles in urban environments

Enisa Trubljanin

 PDF (English)

Usage of Fuzzy Logic and Machine Learning in Web Development

Zlatko Čović

 PDF (English)

Lágy számítás alkalmazása ellenőrzés területén

Polett Péczka

 PDF (English)

Nyelv

Magyar

English

Bánki Közlemények - Bánki Reports (ISSN 2560-2810)

Platform
workflow
OJS / P

Fuzzy Logic in Drone Control

Anja Stijepović⁽¹⁾, Una Sredović⁽²⁾,

⁽¹⁾ University of Donja Gorica, Faculty of applied sciences, Mathematics, Podgorica, Montenegro,
anja.stijepovic2@udg.edu.me

⁽²⁾ University of Donja Gorica, Faculty of applied sciences, Mathematics, Podgorica, Montenegro,
una.sredovic@udg.edu.me

Abstract — In this paper, we examine altitude and attitude stabilization of an unmanned aerial vehicle (UAV) using Fuzzy Logic. There are numerous types of UAVs, but the model analyzed in this paper is designed for a quadrotor micro aerial vehicles with mass no greater than 0,1 kg. A fuzzy logic controller with two inputs (errors and derivative errors) and a fuzzy output is designed in order to stabilize the roll, pitch and yaw angles and the altitude of a drone with four rotors. Since MATLAB is a well developed and commonly used platform by mathematicians and engineers, it was used for building the Fuzzy model. After the creation of appropriate membership functions and a 3x3 table of if-then rules, centroid method was used for the outputs defuzzification and the Mamdani model was tested in Simulink. In addition, a slight perturbation was applied to observe the robustness of this controller during hovering. The model showed a good and stabilized flight control scheme for a quadrotor, even if it is affected by perturbation.

Keywords: UAV, Quadrotor, Fuzzy Logic Controller, MAV

1 INTRODUCTION

Quadrotor is a drone with four rotors that are directed upwards and placed in a square formation with equal distance from its center. Many quadrotor drones have been developed in the recent years and they differ in size and purposes, such as: Mini drones, Hobby drones, Selfie drones, racing drones and proportional drones. Mini drone or Micro Air Vehicle (MAV) got attention due to its small size with mass less than 0,1 kg, but also due to its durability and autonomous ability. Quadrotor is a typical design for small unmanned aerial vehicles because of its simple structure. It is controlled by adjusting the angular velocities of the rotors which are spun by electric motors. Since the quadrotor has six degrees of freedom (three translational and three rotational) but only four control inputs (the speeds of the four motors), it is an underactuated aerodynamic vehicle whose controlling can be a difficult task. [4] As a result, the robust autonomous flight capability is required for a quadrotor to accomplish the desired tasks. In general, quadrotors are inherently unstable systems, therefore, stabilization control is required. The purpose is to maintain its stability during maneuver, therefore a suitable control strategy is needed. Various control methods have been proposed to control the quadrotor: sliding mode control, PID & LQR control, Feedback Linearization, Backstepping control. However, this paper focuses on Fuzzy Logic controller for stabilization of a micro aerodynamic vehicle. Since the fuzzy logic controller (FLC) is one of the most popular strategies for addressing nonlinear dynamic

uncertainties and a quadrotor is defined by a set of nonlinear equations, fuzzy logic is a good method for stabilizing a mini drone.

2 UNMANNED AERIAL VEHICLE

Unmanned Aerial Vehicle (UAV) is defined as an aircraft without the onboard presence of a human pilot. UAVs can be remotely controlled by a human operator or they can fly autonomously. [1]

The application of UAVs is diverse, such as in military, crop monitoring, coast watch, mineral exploration, telecommunications or ground traffic control. One of the biggest reasons for the widespread use of drones is their ability to traverse and maneuver through areas that would be dangerous for humans to be in. Apart from taking on dangerous tasks, drones can also monitor areas that do not necessarily pose a risk to human workers. Still, the reliance on human workers can add a considerable margin of error and accuracy.

UAVs come in various sizes. Large UAVs may be used alone in missions while small ones may be used in formations or swarms. The latter ones are proving to be quite useful in civilian applications. Drone automation shows a potential for reducing this risk of error. But even though UAVs have become increasingly complex and efficient by using the latest advancements in technology, there is still a chance for enhancement.

2.1 History

The earliest recorded use of an unmanned aerial vehicle for warfighting occurred in July 1849. using a balloon carrier. [2] In 1915, Nikola Tesla wrote about unmanned aerial combat vehicles. The first attempt at a self-propelled drone as an aerial target was completed in 1916 by A.M. Low. It wasn't until World War I that the first pilotless torpedo was invented by the Dayton-Wright Airplane Company. During World War II both Allied and German forces used drones as an aid in missions and trainings. After the end of World War II, drone developers began using jet engines in technologies like the Australian GAF Jindivik and the Model 10001, built for the U.S. Navy by Beechcraft. Even though drones were used 150 years for military research, the first use of non-military purposes was in 2006 when the Federal Aviation Administration issued its first commercial drone permit. After 2013, drones were also available to citizens, who most often use them for commercial purposes and for recording panoramic views.

3 FUZZY LOGIC

Fuzzy logic is similar to a human being's feeling and inference process. Unlike classical control strategy, which is a point-to-point control, fuzzy control is a range-to-range control. The inputs and outputs of the controller are the same as the classical techniques, so the input is the error in the controlled variable, and the output is the control magnitude. However, the output of a fuzzy control is derived from fuzzification of both inputs and outputs using the associated membership functions. A crisp input, based on its current value, will be converted into different members of the associated membership functions. From this point of view, the output is based on its membership in different membership functions, which can be considered as a range of inputs.

Implementation of a fuzzy logic technique to real life problems requires four main steps:

1. Fuzzification – refers to the process of transforming crisp values into grades of membership using linguistic terms of fuzzy sets. The membership of each function is determined by a predefined fuzzy rule, which represents the fuzzy logic.
2. Fuzzy rule base – a component based on if-then rules. Fuzzy rule base will consider how to react with an input.
3. Inference – applies fuzzy rule base to the fuzzy outputs. After consideration of fuzzy rule base, an inference engine will provide an output and pass this output to defuzzification.
4. Defuzzification – a method to obtain mathematical data from the output of fuzzy rule base. In other words, it provides the final output of the fuzzy controller. [3]

A fuzzy control system primarily refers to the control of processes through fuzzy linguistic descriptions. More precisely, it is a mathematical system that analyzes analog input values in terms of logical variables. The advantages that the fuzzy logic has over neural networks and genetic algorithms are that the solution to a particular problem can be realized in terms that human operators can understand and their experience can be used to design the controller's if-then rules. The main benefit of fuzzy control is introducing clarity to the development, evaluation and maintenance of a control system.

A special class of control problems is control of highly nonlinear processes that are exposed to strong influences of external disturbances. Such systems are controlled by operators using their years-long experience and knowledge about static and dynamic characteristics of the system. The achieved quality of control is usually proportional to the operator's knowledge. The operator's experience is connected to monitoring relevant process variables, and depending on their states and deviations from reference values, operators decide where, how, and

how much they need to act on the process to achieve a given control goal using the if-then rules.

3.1 History

Since the release of MATLAB and its graphical interface Simulink by Mathworks, inc. as a commercial product in 1984, the model design and simulation capabilities of MATLAB have been widely used and developed across several engineering and science disciplines. MATLAB and Simulink are currently used for various engineering implementations, education, research and development. In this paper, MATLAB and Simulink are used to build a fuzzy control system which will bring the drone to a still state without large oscillations.

3.2 MAV modeling

Quadrotor UAV is described by certain dynamic equations. The quadrotor vehicle operates on the concept of variable torques and thrusts. The quadrotor motors are arranged symmetrically in pairs along the horizontal and vertical axes with the forward pair rotating clockwise and the horizontal pair rotating counter-clockwise. This design results in the reaction torques from the pairs of motors being exactly opposed by each other if they are all spinning at the same speed. The elimination of the rotating moment (anti-torque) allows the vehicle to maintain a constant heading while hovering. Yaw is controlled by varying the speeds of the pairs of motors to create a non-zero net counter torque. Altitude is controlled by varying the thrust from each motor by equal amounts to provide a net thrust vector and without a rotational moment. To move in the lateral directions, the relative speeds of each motor in the lateral pair are varied to create a desired lateral thrust offset. The simple design results in small platforms which are battery operated, able to perform a stable hover, and safe to use in indoor environments. [4] Figure 1 and 2 shows a form of a quadrotor.

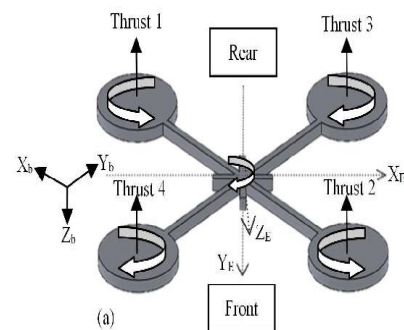


Figure 1: Quadrotor

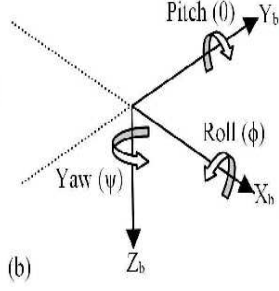


Figure 2: Quadrotor

Translational dynamics consists of three following equations that represent the acceleration of x, y and z. MAV altitude is defined by the equation of coordinate z.

$$\ddot{x} = \frac{U_1}{m} (S\psi S\phi + C\psi S\theta C\phi) \quad (1)$$

$$\ddot{y} = \frac{U_1}{m} (-C\psi S\phi + S\psi S\theta C\phi) \quad (2)$$

$$\ddot{z} = g + \frac{U_1}{m} (C\theta C\phi) \quad (3)$$

The attitude, i.e. the angular position of the quadrotor, is defined in the inertial frame with three Euler angles. Pitch angle θ determines the rotation of the quadrotor around the y-axis. Roll angle ϕ determines the rotation around the x-axis and yaw angle ψ around the z-axis. Following are the rotational equations, or angular rate of roll, pitch and yaw.

$$\ddot{\phi} = \frac{1}{I_{xx}} \left((I_{yy} - I_{zz})\dot{\psi}\dot{\theta} - (J_r\Omega_d)\dot{\theta} + lU_2 \right) \quad (4)$$

$$\ddot{\theta} = \frac{1}{I_{yy}} \left((I_{zz} - I_{xx})\dot{\psi}\dot{\phi} - (J_r\Omega_d)\dot{\phi} + lU_3 \right) \quad (5)$$

$$\ddot{\psi} = \frac{1}{I_{zz}} \left((I_{xx} - I_{yy})\dot{\theta}\dot{\phi} + U_4 \right) \quad (6)$$

In the equations (4), (5) and (6), m is the mass, g is the gravitational coefficient, l is the lateral arm length, and U_1 is the total thrust force generated by four rotors. I is the moment of inertia for the quadrotor, the diagonal matrix 3-by-3 is defined as $I = \text{diagonal}[I_{xx}, I_{yy}, I_{zz}]^T$, J_r is the rotor inertia, and Ω_d is the total rotor speed generated from the two pairs of rotors. Total torque is represented by U_2, U_3 and U_4 . [5]

3.3 Control design

Fuzzy logic is designed to control MAV during hovering and attitude stabilization. The variables z, roll (ϕ), pitch (θ) and yaw (ψ) are used as a feedback response and four dedicated Fuzzy logic controllers are designed to control each state. The errors (e) and its derivatives (de) are used as fuzzy controller inputs in order to generate required outputs for the stabilization of the MAV. In addition, an integral feed forward is applied to improve

steady state of the response signal. The controller architecture used in this paper is illustrated in Figure 3.

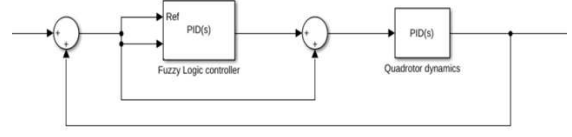


Figure 3: Simulink model

In this Mamdani fuzzy model, three membership functions are created for the error and derivative of the error. The error and derivative error are set to normalize at range $[-2, 2]$. The first input (error) has three membership functions: two Gaussian functions and one triangular function. The second input (derivative error) has three Gaussian membership functions. The linguistic variables used for error and derivative error are negative, zero and positive.

Five membership functions are created for the output to cover more areas during centroid methods of defuzzification. Three of output membership functions are triangular, and two of them are Sigmoid type functions. The range for the output is normalized at $[-10, 10]$. The linguistic variables used for the output are slow down, fast down, hover, slow up and fast up. Figures 4, 5 and 6 show the membership functions developed for the error, derivative error, and output, respectively. Figure 7 shows the output of the Mamdani model.

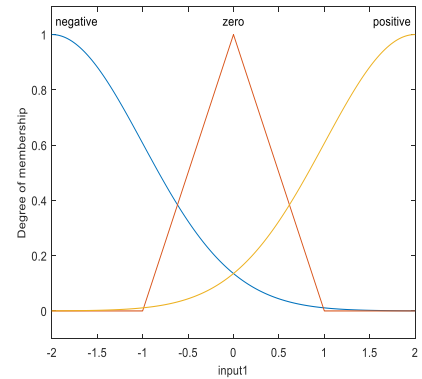


Figure 4: E input membership functions

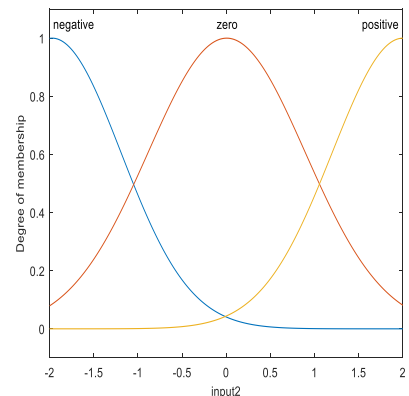


Figure 5: De input membership functions

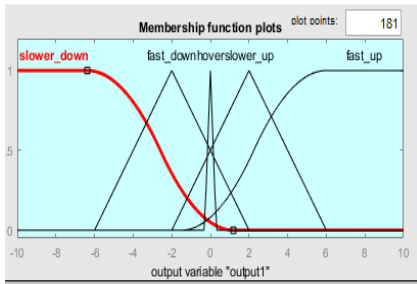


Figure 6: Output membership functions

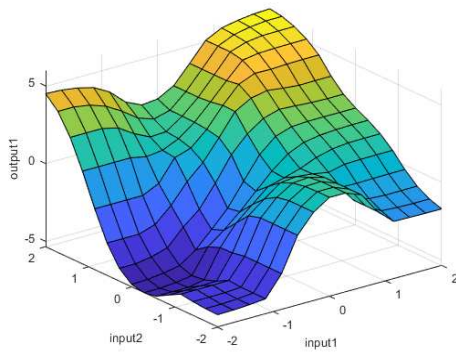


Figure 7: 3D output of the Mamdani model

A 3x3 table of if/then rules is created. For instance, if error is zero and derivative error is zero, then the drone should hover. The number in the bracket is a weightage set for the fuzzy engine to determine probability of which rule needs to be active in case there are two set of rules developed.

Table 1: Table of rules

e \ de	NEGATIVE	ZERO	POSITIVE
NEGATIVE	Fast down (0.5) Slower down (0.5)	Slower down (1)	Hover (0.5) Fast down (0.5)
ZERO	Slower down (1)	Hover (1)	Slower up (1)
POSITIVE	Hover (0.5) Fast up (0.5)	Slower up (1)	Fast up (1)

4 SIMULATION RESULTS

To verify the reliability and robustness of the Fuzzy controller, the quadrotor model is simulated in MATLAB Simulink. A few tests were conducted in order to evaluate the performance of the designed fuzzy controller. Integral absolute error (IAE) is chosen as the index performance.

Initially, the quadrotor is set at [0, 0, 1, 0, 1, 0, 1] and [-2, 0, 0, 0] for z , ϕ , θ , and ψ , respectively. The gains were manually tuned to obtain satisfactory results and settling times. The simulation results presented in Figures 8 and 9 demonstrate the satisfactory performance of the proposed

Fuzzy Logic Control system. In the simulation, for the altitude control, the settling time to reach 2 m is about 4 s.

In the attitude control, the settling time is about 1 ms, 2 ms, and 0,64 ms for roll, pitch, and yaw, respectively. The integral absolute error (IAE) for the case without perturbation for the altitude z is 4,6 and about 0,07 for the attitude. Meanwhile, for the case with perturbation as shown in Figures, the IAE increases to 5,4 with a small error for the altitude but it remains the same for the attitude. These results demonstrate a good performance of the Fuzzy Logic controller for a small-scale quadrotor.

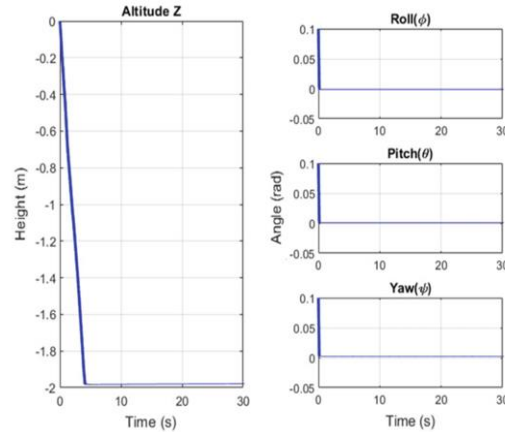


Figure 8: Altitude and attitude control using the Fuzzy controller

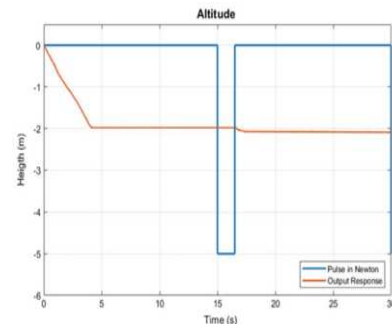


Figure 9: Altitude with slight perturbation using the Fuzzy controller

5 CONCLUSION

Fuzzy Logic has many useful applications in transportation, electronics, business and finance, defense, medicine, etc. In this paper, we examined the use of Fuzzy Logic in drone control. A Fuzzy Logic controller with 3x3 membership functions was proposed for the stabilization of a quadrotor micro aerial vehicle during hovering and movement. The simulation of the proposed Fuzzy Logic Control system was conducted in a MATLAB Simulink environment, where the results demonstrated an acceptable performance without or with perturbation applied on the altitude (z) and attitude (roll, pitch and yaw angles) of the quadrotor. Fuzzy Logic proved to be a good method for drone stabilization, however, better results can be achieved by developing different types of Fuzzy models or using other membership functions in the proposed model. Even though today's aerial vehicles are well developed and stabilized, there are still ways to improve using new technologies.

REFERENCES

- [1] Hu, J. Lanzon, A.: "An innovative tri-rotor drone and associated distributed aerial drone swarm control" (2018) *Robotics and Autonomous Systems* 103. 162-174
- [2] R. D. Layman: "Naval Aviation in the First World War: Its Impact and Influence" (1996), 56
- [3] Theerasak Sangyam, Pined Laohapiengsak, Wonlop Chongcharoen, Itthisek Nikhamhang: "Path Tracking of UAV Using Self-Tuning PID Controller Based on Fuzzy Logic" (2010)
- [4] Teppo Luukkonen: "Modelling and control of quadcopter" (2011)
- [5] Aminurrashid Noordin, Mohd Ariffanan Mohd Basri, Zaharuddin Mohamed: "Fuzzy Logic Control for Quadrotor Micro-aerial Vehicle Altitude and Attitude Stabilization (2022)

Útvonaltervezés heurisztikus módszerekkel – a hangyakolónia algoritmus

Route planning with heuristic methods - the ant colony algorithm

Polyák Gabriella*, Póth Miklós**

* Szabadkai Műszaki Szakfőiskola, Szabadka, Szerbia

** Szabadkai Műszaki Szakfőiskola, Szabadka, Szerbia

*poljak.gabriela@vts.su.ac.rs

**pmiki@vts.su.ac.rs

Összefoglalás — A munka célja egy útvonaltervezési probléma megoldása heurisztikus módszerrel. A problémát lehet az utazó ügynök problémához hasonítani, ugyanis útvonaltervezésről, illetve útvonal optimalizációról van szó. A munka alapját a Formula-1-es versenynaptár nagydíjhelyszíneinek földrajzi koordinátái képezik, azonban bármilyen más, nyilvánosan elérhető adathalmazon is futtatni lehet a programokat. A munkában a Microsoft Excel egyik bővítménye, az Excel Solver lett használva, illetve egy MATLAB-ban íródott program, ami a hangyakolónia algoritmust alkalmazza. A hangyakolónia algoritmus azon a természetbeni megfigyelésen alapul, hogy minden hangya nyomot hagy maga után, egy bizonyos feromon nevű vegyi anyagot, és minél több hangya követi ugyanazt az utat, annál több a lesz a feromon, és ez minden következő hangyának "pozitív információ" az adott út helyességéről. A munkában az Excelben és a MATLAB-ban kapott eredmények kerülnek összehasonlításra és összegzésre.

Kulcsszavak: útvonaltervezés, Excel Solver, hangyakolónia algoritmus, Formula-1

Abstract — The aim of this paper is to solve a route planning problem using a heuristic method. The problem can be compared to the traveling salesman problem, because it is about route planning and route optimization. The basis of the work is the geographical coordinates of the grand prix locations of the Formula 1 race calendar, but the programs can also be run on any other publicly available data set. An extension of Microsoft Excel, Excel Solver, was used in the paper, as well as a program written in MATLAB, which uses the ant colony algorithm. The ant colony algorithm is based on the natural observation that each ant leaves a trail, a certain chemical substance called a pheromone, and the more ants follow the same path, the more pheromone will be, and this is "positive information" for each subsequent ant. about the correctness of a given road. In this work, the results obtained in Excel and MATLAB are compared and summarized.

Keywords: route planning, Excel Solver, ant colony algorithm, Formula 1

1 BEVEZETÉS

A munka alapelvei az utazó ügynök probléma elvein alapulnak, mivel útvonaltervezésről, illetve annak optimalizációjáról van szó. Az utazó ügynök probléma esetén az ügynöknek n számú várost kell bejárnia, hogy az utazási költségek minimálisak legyenek. Ennek a problémának két változata van. Az első változatban az ügynöknek vissza kell térnie a kiindulási helyzetbe, így a probléma megoldása a Hamilton-kör. A második változatban az utazó ügynök kezdő és záró városa eltérő, így ebben az esetben a megoldás Hamilton-út [1].

A közelmúlt világméretű problémái, mint a koronavírus-járvány és Kelet-Európa bonyolult geopolitikai helyzete, illetve a jelenlegi energiaválság, jelentős hatással voltak és vannak az emberek mindennapi életének minden szegmensére, így a sportra, pontosabban az autó- és motorsportra is. Az autó- és motorsport sorozatok csúcskategóriái folyamatosan szigorúbb energiafogyasztási korlátozásokat vezetnek be, hogy ösztönözzék a nagy hatékonyságú hajtáslánc-technológia fejlesztését. A Formula-1-ben a verseny során korlátozott az üzemanyag fogyasztás, valamint korlátozott a hibrid rendszer villamosenergia felhasználása [2].

Az üzemanyag fogyasztás és az üvegház hatású gázok kibocsátásának csökkentése érdekében rendkívül fontos lenne a Formula-1-es versenynaptár optimalizálása. Ebben az esetben a versenyautók, a felszerelések, az alkatrészek és a csapatok éves szinten sokkal kevesebbet utaznának, ez a megtakarítás pedig több ezer kilométerben mérhető. Az üvegház hatású gázok kibocsátása és az üzemanyag fogyasztás csökkenése mellett jelentős anyagi és időmegtakarítás is elérhető.

2 IRODALMI ÁTTEKINTÉS

Az utazó ügynök probléma (TSP) az egyik legszélesebb körben tanulmányozott kombinatorikus optimalizálási probléma. Állítása megtévesztően egyszerű, mégis továbbra is az operációkutatás egyik legnagyobb kihívást jelentő problémája. A TSP leggyakoribb gyakorlati értelmezése az, hogy egy utazó ügynök a legrövidebb utat keresi n ügyfélen vagy városon keresztül. Ez az alapvető probléma számos járműirányítási alkalmazás alapját képezi [3].

Amikor a gyakorlatban optimalizálni kell egy utazó ügynök problémát, gyorsan számos nehézségbe előtérbe kerülhet. Előfordulhat, hogy nincs kéznél algoritmus egy példa optimális megoldására, és hogy az idő vagy a tudás nem teszi lehetővé egy ilyen algoritmus fejlesztését és megvalósítását. A problémák egyszerűen túl nagyok lehetnek, és ezért meghaladják még a legjobb algoritmusok képességeit is az optimális megoldások megtalálásához. Másrészt az is előfordulhat, hogy a számításához szükséges idő nem elegendő ahhoz, hogy egy algoritmus elérje az optimális megoldást. Mindezekben az esetekben határozottan szükség van olyan közelítő (heurisztikus) algoritmusokra, amelyek jó minőségű megoldásokat határoznak meg, és az adott korlátok mellett a legjobb elérhető eredményeket adják [4].

Számos heurisztikus technikát alkalmaztak a probléma hatékony megoldásának megtalálására, mint például a hangya kolónia algoritmusok, szimulált hűtés, részecske-raj alapú optimalizálás és genetikai algoritmusok [5]. De ahogy a városok száma növekszik, a megoldás megtalálásához szükséges számítások nehézkessé válnak. A számítási nehézségek ellenére olyan módszereket lehet használni, mint a genetikai algoritmusok és a tabu keresés, amelyek közel optimális megoldást nyújthatnak több ezer városra is [6].

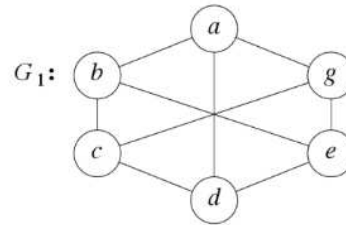
A hangya kolónia algoritmus az utazó ügynök probléma megoldásának egyik metaheurisztikus módszere. A hangya kolónia algoritmus hasonlít a valódi hangyák viselkedésére. Miután élelmet találtak, visszatérnek a kolóniába, miközben feromonnyomokat hagynak maguk után. Ha más hangyák találnak ilyen utat, valószínűleg nem véletlenszerűen választanak más utat, hanem inkább követik a nyomot. Nagyobb a valószínűsége annak, hogy a hangyák a magasabb feromonkoncentrációjú nyomvonalat választják. A feromonnyom lehetővé teszi a hangyák számára, hogy visszatérjenek az élelmiszerforráshoz, és fordítva. A nyomvonalat más hangyák is használják a felfedezett táplálékforrás megtalálására. Ahogy egyre több hangya választja ki a követendő utat, az ösvényen lévő feromon felhalmozódik, vonzóbbá téve azt más hangyák számára is [6].

3 ELMÉLETI MEGALAPOZÁS

Ebben a fejezetben a munka alapelemei, az útvonaltervezési módszerek és kritériumok kerülnek bemutatásra. A munkában a Microsoft Excel egyik bővítménye, az Excel Solver lett alkalmazva, illetve a MATLAB-ban egy hangyakolónia algoritmust alkalmazó program.

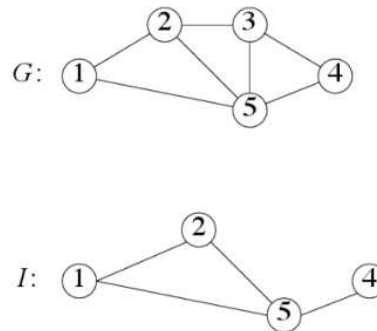
3.1 Gráfok

A gráfok az alapvető elemek ebben a munkában, mivel a helyek matematikai absztrakcióját reprezentálják, a köztük lévő élek pedig a helyek közötti útvonalakat. Számos probléma megoldására szolgálnak, amelyek valós vagy absztrakt objektumok formájában és valós vagy absztrakt kapcsolatok formájában jeleníthetők meg ezek között az objektumok között, például: forgalom, távközlési hálózat stb. A G_1 gráf két véges halmazból áll: a csomópontok összességéből (N) és a köztük lévő élek összességéből (C) úgy, hogy minden $g \in C$ kapcsolat két i és j csomópontot köt össze (1. ábra).



1. ábra: G_1 gráf, példa [7]

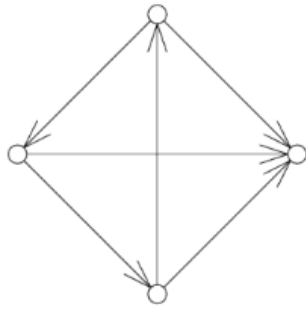
Egy egyszerű G gráf az, amelyben nincsenek párhuzamos élek (két azonos csomópontot összekötő kapcsolatok) és ciklikus élek (egy csomópontot önmagával összekötő kapcsolatok). Egy egyszerű gráfban az élek legnagyobb száma egyenlő az $NC(NC-1)/2$ -vel, ahol NC az N halmaz csomópontjainak száma. A teljes gráf egy olyan egyszerű gráf, amely az élek maximális számát tartalmazza. Ha két csomópont között összekötő él van, akkor azok szomszédos csomópontok. A G gráf részgráfja bizonyos csomópontok és a hozzájuk tartozó élek részhalmaza (2. ábra).



2. ábra: G gráf és I részgráf [7]

Ha a gráf csomópontjai konkrét objektumok, amelyek meghatározott pozícióval rendelkeznek a térben, akkor ezek euklideszi gráfok. Az útvonal az N -ből származó csomópontok sorozata, ahol minden két szomszédos csomópontot egy C -ből származó él köt össze. Ha az út ugyanabban a csomópontban kezdődik és végződik, akkor ez egy gyűrű (ciklus vagy ciklikus út). Az egyszerű útvonal olyan sorozat, amelyben a csomópontok nem ismétlődnek. Az Euler-útvonal (ciklus) olyan útvonal (ciklus), amely az összes élet tartalmazza, és amelyben minden él csak egyszer jelenik meg [8].

Egy gráf akkor kapcsolódik össze, ha van legalább egy él, amely bármely két csomópontot összeköti. Az irányított gráf olyan gráf, amelyen egy összefüggés rendezett pontpárként jelenik meg, és $(i, j) \neq (j, i)$. A digráfban különböző orientációjú párhuzamos kapcsolatok megengedettek (3. ábra).



3. ábra: Digráf, példa [7]

Gráfelméleti algoritmusok:

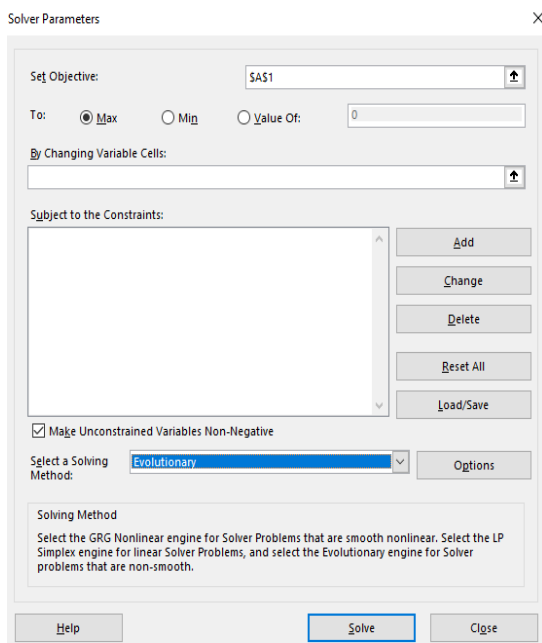
- Szélességi keresés (Breadth First Search – BFS)
- A legrövidebb út megtalálása – Bellman optimalitási elve
- Mélységi keresés (Depth First Search - DFS)

3.2 Útvonaltervezési módszerek

Ebben a részben a munkában használt módszerek, azaz az Excel Solver és a hangyakolónia algoritmus kerülnek bemutatásra.

3.2.1 Excel Solver

A Solver programok olyan matematikai optimalizáló alkalmazások, amelyek matematikai problémák megoldására használhatók. A cél egy olyan általános modell megalkotása, amelynek segítségével sok hasonló probléma megoldható. Az Excel Solver működési elve, hogy a korlátok figyelembevételével megvizsgálja a lehetséges megoldásokat, majd kiválasztja közülük a legjobbát. Az Excel Solver segítségével megkereshető egy képlet optimális értéke (minimális, maximális vagy specifikus célérték) egy úgynevezett céllében. Az Excel Solver ablaka a 4. ábrán látható.



4. ábra: Az Excel Solver ablakának kinézete

Az út létrehozása a (1) képlettel történik, ahol a valószínűsége, hogy a k . hangya az i . pontból a j . pontba megy az algoritmus t . iterációjában, az alábbi [10]:

$$p_{ij}^k = \frac{[\tau_{ij}(t)]^{\alpha} * [\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}(t)]^{\alpha} * [\eta_{il}]^{\beta}}, \text{ ha } j \in N_i^k \quad (1)$$

Ahol $\eta_{ij} = \frac{1}{d_{ij}}$, két csomópont közötti távolság reciprok értéke, $\tau_{ij}(t)$ pedig az (i, j) él feromontartalma.

A feromon frissítésére a (2) képlet kerül alkalmazásra:

$$\tau_{ij}(t + 1) = (1 - \rho) * \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k(t) \quad (2)$$

Ahol

$$\Delta \tau_{ij}^k(t) = \begin{cases} \frac{1}{L^k(t)}, & \text{ha az } (i, j) \text{ élen átment a } k. \text{ hangya} \\ 0, & \text{különben} \end{cases} \quad (3)$$

3.3 Útvonaltervezési kritériumok

Az útvonaltervezés kritériumai gyakran a távolságon vagy az utazási időn alapulnak [11]. A versenynaptár optimalizálásakor több szempontot is figyelembe kell venni, hiszen a versenyek sorrendjének meghatározása és a közöttük lévő útvonal tervezése összetett probléma. A versenynaptár útvonalát a következő kritériumok szerint lehet tervezni:

- Legrövidebb útvonal
- Leggyorsabb útvonal
- Időjárás és éghajlat
- Pénz
- Politikai és vallási aktualitások
- Egyéb hatások

3.3.1 Legrövidebb útvonal

Ebben a munkában az útvonal tervezése csak ez a kritérium alapján történt. Ha ez alapján tervezzük a versenynaptárt, az azt jelenti, hogy az autóknak, felszereléseknek, alkatrészeknek és a csapatoknak több ezer kilométerrel kevesebbet kell megtenniük, ezzel csökkenne az üzemanyag fogyasztás és a káros gázok kibocsátása is, illetve jelentős pénzügyi és időbeli megtakarítás is lenne.

3.3.2 Leggyorsabb útvonal

Gyakran nem a legrövidebb út a leggyorsabb is, ennek oka lehet a sebességkorlátozás, a terep jellege vagy a forgalmi torlódás.

3.3.3 Időjárás és éghajlat

Egy adott év Formula-1-es szezonja általában márciusban kezdődik. Mivel ilyenkor Európában általában még elég hűvös van, így az időjárás nem megfelelő a versenyzésre. Az elmúlt jó pár évben a szezonkezdő vagy valamelyik közel-keleti versenyhelyszínen volt, vagy Ausztráliában. Az első 2-3 megrendezett futam után, mikor Európában is emelkedni kezd a napi átlaghőmérséklet, akkor következnek az európai versenyek és kisebb-nagyobb kivételekkel a szárgüldő cirkusz őszi leginkább az öreg kontinensen is marad. Ősszel az időjárás körülmények miatt következnek a

távol-keleti helyszínek, majd Dél-Amerika és végül ismét a Közel-Kelet.

3.3.4 Pénz

A versenynaptár nagydíjainak sorrendjében az időjárás mellett a pénzé a legfontosabb szerep. A naptárban az első és az utolsó nagydíj lebonyolításáért plusz díjat kell fizetni, amit a legtöbb ország nem tud kifizetni, így hosszú évek óta az év utolsó nagydíja az Abu Dzabi Nagydíj. A pálya is a legrágább a Forma-1 eddigi történelmében, mivel a teljes komplexum megépítése több mint 1 milliárd dollárba került [12]. A nagydíjak szervezői átlagosan körülbelül 40 millió dollárt fizetnek a Liberty Media Corporationnek, a Formula-1 többségi tulajdonosának egy nagydíj megszervezésének lehetőségéért [13].

3.3.5 Politikai és vallási aktualitások

Az Orosz Nagydíjat 2022-ben a Kelet-Európában jelen levő bonyolult geopolitikai helyzet miatt azonnal törölték a versenynaptárból, így a naptár 22 nagydíjhelyszínt tartalmazott. A 2024-es versenynaptár szempontjából pedig fontos a ramadán kezdeti dátuma, ugyanis március elejére esik, abban az időszakban pedig akkor nem rendezhetnek nagydíjat a közel-keleti, főleg iszlám vallásúak lakta országokban, a ramadán végeztével, ami április elejére esik, pedig már túl magas a hőmérséklet egy nagydíjhétvége megszervezéséhez abban a térségben.

3.3.6 Egyéb hatások

A korábbi években különösen nehéz volt a Formula-1-es versenynaptár megszervezése. 2020 lelegején az Ausztrál Nagydíjat törölték az ausztrál erdőtüzek miatt. A következő futamnak és a szezon hivatalos rajtjának Bahreinben kellett volna lennie, ahová minden csapat meg is érkezett, azonban pénteken, egy órával az első szabadedzés kezdete előtt, a teljes versenyhétvégét törölték a koronavírus-járvány miatt. Közvetlenül ezután a többi futamot is törölték, így a 2020-as szezon hivatalos rajtja július elején volt Ausztriában. 2020-ban 14 helyszínen rendeztek, összesen 17 nagydíjat. 2021-ben a járvány miatt szintén történtek változások, de végül megtartották a nagydíjakat; 21 helyszínen és összesen 22 nagydíjat. A 2022-es évre visszatértek azok a nagydíjak, amelyeket a járvány miatt az előző két évben nem rendeztek meg, azaz az Ausztrál, a Kanadai, a Japán és Szingapúri Nagydíj. A Kínai Nagydíj a járvány miatt talán majd csak 2024-ben kerül vissza a versenynaptárba. A Katari Nagydíj 2021-ben debütált, de 2022-ben a labdarúgó-világbajnokság miatt nem került megrendezésre.

4 GYAKORLATI MEGVALÓSÍTÁS – ÚTVONALTERVEZÉS

Az útvonaltervezés többféle módon történt. Először Microsoft Excelben, az Excel Solver nevű bővítményével, majd a MATLAB-ban írt hangyakolónia algoritmust használó programban.

4.1 Excel Solver

A 2020-as versenynaptár végül 14 nagydíjhelyszínből és 17 megrendezett nagydíjhétvégéből állt. Három alkalommal dupla hétvége volt, ami azt jelenti, hogy két egymást követő hétvégén ugyanazon a helyszínen két, különböző nevű nagydíj került megrendezésre (1. táblázat).

1. táblázat: A versenypályák elnevezései és a földrajzi koordinátáik (2020)

	Versenypálya	Földrajzi szélesség	Földrajzi hosszúság
1	Red Bull Ring	47,219722	14,764722
2	Hungaroring	47,582222	19,251111
3	Silverstone Circuit	52,078611	-1,016944
4	Circuit de Barcelona-Catalunya	41,57	2,261111
5	Circuit de Spa-Francorchamps	50,437222	5,971389
6	Autodromo Nazionale Monza	45,620556	9,289444
7	Autodromo Internazionale del Mugello	43,9975	11,371944
8	Sochi Autodrom	43,410278	39,968271
9	Nürburgring	50,335556	6,9475
10	Autódromo Internacional do Algarve	37,232	-8,632
11	Autodromo Internazionale Enzo e Dino Ferrari	44,341111	11,713333
12	Intercity Istanbul Park	40,951667	29,405
13	Bahrain International Circuit	26,0325	50,510556
14	Yas Marina Circuit	24,467222	54,603056

A nagydíjhelyszínek közötti távolságot a (4) képlet alapján számoljuk, ami az euklideszi távolság kiszámításának képlete.

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2} \quad (4)$$

A 2. táblázatban a bal oldalon az optimalizálás előtti euklideszi módszerrel kiszámított távolságok, a sorrend és a távolságok összege látható, ez az a sorrend, ami a valóságban is volt, tehát a versenynaptár sorrendje. A jobb oldalon az optimalizálás utáni eredmények láthatók, az optimalizált út körülbelül 1/3-dal rövidült. Az eredeti összeg 252,741 (26 035 km), míg optimalizálás után 154,036 (15 867 km) lett az eredmény.

2. táblázat: Az útvonal hossza optimalizálás előtt és után (2020)

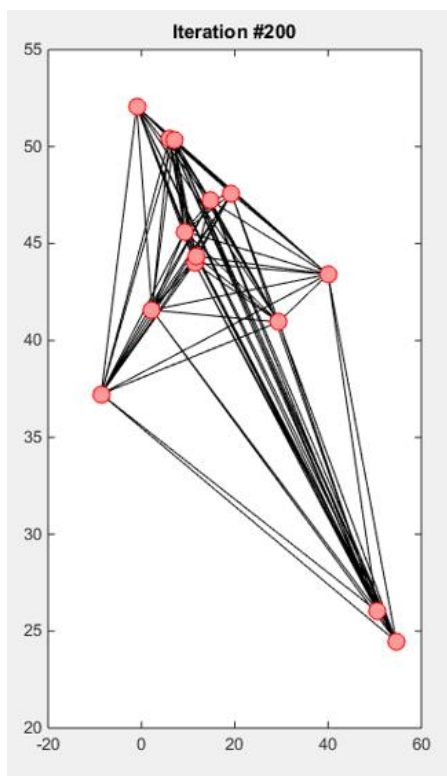
Sorszám	Távolság	Sorszám	Távolság
1	4,50101	1	8,41531
2	20,7608	9	0,98139
3	11,008	5	7,17851
4	9,61217	3	16,6857
5	5,84891	10	11,7251
6	2,64029	4	8,112
7	28,6024	6	2,64029
8	33,7392	7	0,48437
9	20,3574	11	18,0134
10	21,5516	12	25,8462
11	18,0134	13	4,38163
12	25,8462	14	23,9378
13	4,38163	8	21,1331
14	45,8778	2	4,50101
1		1	

Összesen	252,741	Összesen	154,036
----------	---------	----------	---------

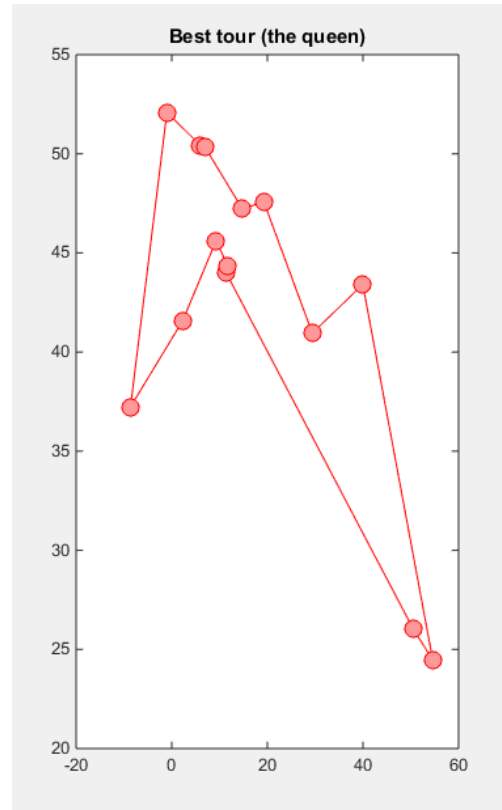
A 2020-as versenynaptár után ugyanez a módszer lett alkalmazva a 2021-es, 2022-es és 2023-as versenynaptárra, valamint az eddigi összes, azaz 77 nagydíjhelyszínre.

4.2 MATLAB – hangyakolónia algoritmus

A hangyakolónia algoritmust alkalmazó program tetszőleges számú hangyára és iterációra futtatható. A 2020-as versenynaptár 14 nagydíjhelyszíne esetén 10 hangyára és 200 iterációra lett lefuttatva a program. Az 5. ábrán látható a 200. iterációban a 10 hangya által bejárt útvonalak. A program lefutása után a legrövidebb útvonal hosszúsága 155,181 (15 985 km). A 6. ábrán koordináta-rendszerben ábrázolva látható a kapott legrövidebb útvonal. A 2020-as versenynaptár után ugyanez a módszer alkalmazva lett a 2021-es, 2022-es és 2023-as versenynaptárra, valamint az eddigi összes, azaz 77 nagydíjhelyszínre.



5. ábra: A 200. iterációban a 10 hangya által bejárt útvonalak



6. ábra: A kapott legrövidebb útvonal koordináta-rendszerben ábrázolva

5 EREDMÉNYEK, ÖSSZEZÉS

A 3. táblázatban összegezve láthatóak a kapott legfontosabb eredmények. A táblázat az összes módszert (Solver és hangyakolónia algoritmus) tartalmazza, illetve az összes esetet amire alkalmazva lettek (2020-as, 2021-es, 2022-es, 2023-as versenynaptár és 77 nagydíjhelyszín). A 2020-as versenynaptár esetén az eredeti útvonal közel 40%-kal csökkent, a Solverrel és a hangyakolónia algoritmussal kapott eredmények között kicsit több mint 100 km a különbség. A 2021-es versenynaptár esetén szintén közel 40 %-os javulás látható, ebben az esetben közel 1000 km-re rövidebb útvonal lett meghatározva a hangyakolónia algoritmus segítségével, mint Excel Solverben. A 2022-es versenynaptár esetében az útvonal az eredetihez képest közel 55%-kal csökkent, közel 2000 km különbség van a Solver és a hangyakolónia algoritmus között. A 2023-as versenynaptár esetében az útvonal kevesebb mint felére csökkent az eredetihez képest, az Excel Solver jobban teljesített, mint a hangyakolónia algoritmus, ugyanis több mint 3000 km-rel rövidebb lett a kapott útvonal. Az összes eddigi, azaz 77 nagydíjhelyszín esetében lényegesen nagyobb a két módszer között különbség, mint az előző esetekben. A hangyakolónia algoritmus segítségével kapott útvonal közel 45 000 km-rel rövidebb, mint az Excel Solverben kapott. Nagyobb mennyiségű bemeneti adattal az Excel Solver nehezebben dolgozik, míg a kis mennyiségű adatokkal nincs problémája. A hangyakolónia algoritmus pedig kiválóan boldogul nagyobb mennyiségű bemeneti adattal is, csak a program futásának ideje lesz hosszabb.

3. táblázat: Összefoglaló táblázat a kapott eredményekről

	Eredeti	Excel Solver	Hangyakolónia algoritmus
2020	26 035 km	15 867 km	15 985 km
2021	64 947 km	40 823 km	39 684 km
2022	139 875 km	66 304 km	68 259 km
2023	149 087 km	69 484 km	72 746 km
77 helyszín	562 456 km	136 599 km	91 782 km

A jövőbeni tervek között szerepel még több metaheurisztikus módszer alkalmazása, illetve az útvonaltervezés végrehajtása a munkában említett többi kritérium szerint is.

IRODALOMJEGYZÉK

- [1] Stankovic, V. & Jovic, J (2006). *Ekonomika – Problem trgovačkog putnika*. vol. 52, br. 5-6, str. 198-202
- [2] Liu, X., Fotouhi, A. & Auger, D. (2022). Application of advanced tree search and proximal policy optimization on formula-E race strategy development. DOI: <https://doi.org/10.1016/j.eswa.2022.116718>
- [3] Laporte, G. (1992). The traveling salesman problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59(2), 231–247. doi:10.1016/0377-2217(92)90138-y
- [4] Jünger, M., Reinelt, G., & Rinaldi, G. (1995). Chapter 4 The traveling salesman problem. *Network Models*, 225–330. doi:10.1016/s0927-0507(05)80121-5
- [5] Kylie Bryant, Arthur Benjamin, Advisor, “Genetic Algorithms and the Traveling Salesman Problem”, Department of Mathematics, December 2000.
- [6] Dahiya C. & Sangwan S. (2018). Literature Review on Travelling Salesman Problem, *International Journal of Research - June 2018*
- [7] Gould, R. *Graph Theory*. Courier Corporation, 2012 ISBN: 9780486498065
- [8] Dasic T., & Stanic M. (2014). *Metode optimizacije – Grafovi algoritmi*
- [9] Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4), 28–39. doi:10.1109/mci.2006.329691
- [10] Agárdi, A. Klaszterezés hangya kolónia optimalizációval és tabu kereséssel. *Multidiszciplináris tudományok*, 11. kötet. (2021) 4 sz. pp. 120-126. <https://doi.org/10.35925/j.multi.2021.4.14>
- [11] Sampaio, C., Bandeira, J. M., Macedo, E., Vilaça, M., Guarnaccia, C., Friedrich, B., Coelho, M. C. (2019). A Dynamic Link-based Eco-indicator for supporting equitable traffic management strategies. *Transportation Research Procedia*, 37, 43–50. doi:10.1016/j.trpro.2018.12.164
- [12] <https://wtf1.com/post/how-much-does-it-cost-to-host-an-f1-race/>
- [13] <https://hu.motorsport.com/f1biz/news/mennyibe-kerulhet-egy-formula-1-es-auto/4392635/>

Mathematical models of tumor growth

Sara Medojević

University of Donja Gorica, Podgorica, Montenegro

sara.medojevic@udg.edu.me

Abstract —Cancer is one of the deadliest diseases, therefore it is necessary to develop tools that will be useful in the decision-making process. One of those are mathematical models that are used to predict the growth of tumors and the changes that occur as a result of radiation. The determination of the most suitable model is based on the type of tumor. The models presented in this paper are based on ordinary differential equations. Gompertz model is presented as a model that originated from the generalized two-parameter model. Although the Gompertz model is used for determining tumor volume, in this paper it is shown how it can be used to determine the maximum volume of a tumor. The second model presented in this paper is the kinetic model. Kinetic model is used for showing changes in cancer volume before, during and after the active radiation effect. In addition, it can be used for comparing the affect of different radiation doses. The Gompertz model is applied to a benign tumor (meningioma), while the kinetic model is applied to a malignant tumor (hepatocellular carcinoma). As observing the volume at a certain time span implies solving a system of ordinary differential equations, proposed models are written in the form of codes in MATLAB.

Keywords: mathematical models, Gompertz model, kinetic model, tumor growth, radiation

1 INTRODUCTION

Cancer is one of the most common causes of death in the world, which makes it one of the deadliest diseases. Statistics from the World Cancer Research Fund from 2020 show that 18.1 million people have cancer [9]. Which method of treatment is the most effective and how mathematical modeling can be helpful, it is necessary to understand what tumors are and how they develop.

A neoplasm is any abnormal growth of cells. The term tumor is usually used as a synonym for neoplasm. Tumors can be benign or malignant, which is one of the key factors that determine the way to fight the disease.

A stems tumor from a single cell that begins to abnormally expand (proliferation process). What characterizes a benign tumor is that it remains confined to its original location. It is usually enclosed in a fibrous capsule and does not spread to other tissues and organs in the body. As benign tumors remain localized at the place of origin, they can usually be removed by surgery and are not life-threatening. The fact that tumors originate from a single cell does not mean that they will remain benign. Normal cells mutate into cancerous cells through a series of changes that lead to abnormal cell growth, culminating

in malignancy. Only malignant tumors are considered cancers and their ability to invade normal tissues and spread throughout the body (metastasize) is what makes them so dangerous [4].

When making a decision on how to treat cancer, it is necessary to consider the type and stage of the cancer, location, volume and the rate of cell proliferation. While most benign tumors are usually removed by surgery, the removal of malignant tumors usually involves a combination of chemotherapy and radiotherapy. Radiation works by making small breaks in the DNA inside the cell. These interruptions prevent cancer cells from growing and dividing further [1]. Cancer cells then either lose the ability to proliferate or die. Although radiation is an effective method of cancer treatment, in addition to affecting cancer cells, it significantly endangers healthy cells and the immune system. Weakening of the immune system reduces the body's ability to defend itself. Also, the immune system has a role to identify and remove dead cells by breaking them down and eliminating them from the body. It is extremely important to see the impact of cancer on the surrounding tissues, the speed of growth as early as possible and accordingly make a decision on the method of treatment that kills the cancer the fastest and with the least severe consequences to the body. This problem can be solved by mathematical modeling.

Mathematical tumor modeling refers to the use of mathematical equations (most often differential and logarithmic), probability theory and computer simulations to understand tumor growth and spread. Due to its complexity and the individuality of each case, there are different models that monitor tumor growth. Thus, in the case of benign tumors, the rate of tumor growth and its dependence in relation to the developed new blood vessels (tumor vasculature) are most often monitored, while in the case of malignant tumors, the ratio of the volume of the tumor in relation to the amount of radiation, the number of proliferating, dead, removed cells, etc. is additionally monitored.

In the Section 2, Gompertz and kinetic models are presented. Kinetic model observes the tumor volume in relation to one dose of Gy (Gray) radiation.

In Section 3, the implementation of the proposed models in MATLAB will be presented, as well as their application to a benign tumor (meningioma) and a malignant tumor (hepatocellular carcinoma).

2 MATHEMATICAL MODELS

Mathematical models can be divided into two groups: two-parameter and multi-parameter. Two-parameter

models are usually applied on benign tumors, while multi-parameter models are usually applied on malignant tumors. In the following, the models derived from the generalized two-parameter model will be presented [6].

2.1 Gompertz model

One of the simplest model for modeling tumor growth is the logistic model. Tumor growth can be represented by a sigmoid curve, i.e. growing curve with one inflection point that converges asymptotically towards the maximum volume of the tumor K . What makes this model suitable for representing tumor growth is its characteristic that the growth coefficient (as the function approaches an asymptote) decreases, which is consistent with the general growth patterns of organs and organisms. Tumors whose volume is less than $1mm^3$ can grow without developing vascular structures. All tumors larger than $1mm^3$ develop new blood vessels so that nutrients can reach the tumor, allowing it to continue growing.

The tumor is usually not detected in the stages when its volume is less than $1mm^3$ (if we exclude controlled and experimental conditions), therefore $V_0 = 1$ can be taken for the volume at the initial moment $t = 0$.

$$\frac{dV}{dt} = aV \left(1 - \frac{V}{k}\right) \quad (1)$$

$$V(t = 0) = V_0$$

where a is the coefficient that depends on cell proliferation and V_0 is the initial volume. It is noticed that the expression $1 - \frac{V}{k}$ describes the probability that a cell will divide at a specific time (V depends on t).

Model (1) can be expanded to generalized logistic model where a is a coefficient that depends on cell proliferation.

$$\frac{dV}{dt} = aV \left(1 - \left(\frac{V}{k}\right)^v\right) \quad (2)$$

In order to solve problem (2), Cauchy's problem needs to be solved. Cauchy's problem has an explicit solution

$$V(t) = \frac{V_0 K}{(V_0^v + (K^v - V_0^v)e^{-avt})^{\frac{1}{v}}} \quad (3)$$

It can be noticed that the solution of equation (2) is gotten when $v = 1$ is inserted in equation (3).

By transforming the generalized logical model, the Gompertz model is obtained.

$$\frac{dV}{dt} = a e^{-\beta t} V \quad (4)$$

$$V(t = 0) = V_0$$

where a represents the initial cell proliferation and β is the decreased proliferation cell rate [7].

An essential characteristic of the Gompertz model is that it shows an exponential decline in the relative growth rate. The solution to the previous problem (4) is

$$V(t) = V_0 e^{\frac{a}{\beta}(1 - e^{-\beta t})}$$

indicating that maximum volume of the tumor is

$$K = V_0 e^{\frac{a}{\beta}}$$

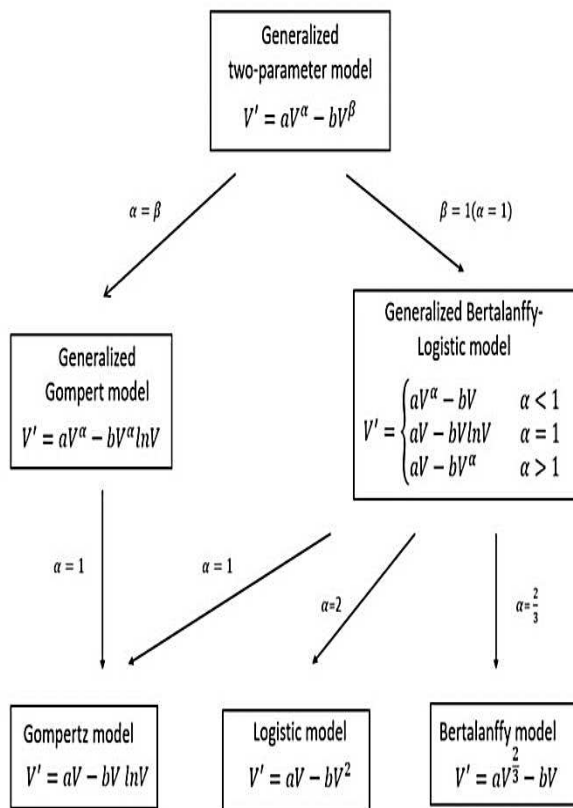


Figure 1: Mathematical models originated from the generalized two-parameter model

As stated in [3], the logistic model describes tumor volume as

2.2 Kinetic model

The kinetic model of tumor volume is applied to malignant tumors. This model shows what effect radiation has on malignant cells and how it affects the total volume of the cancer. Since a large dose of radiation cannot be given at once, it must be divided into lumps of smaller doses. This is why radiation treatments can last from a few days to 7 weeks, depending on the type and stage of the cancer. As stated in [8], this model observes the behavior of tumors after a single treatment with a lower dose of radiation. Cancer consists of dividing and non-dividing cells.

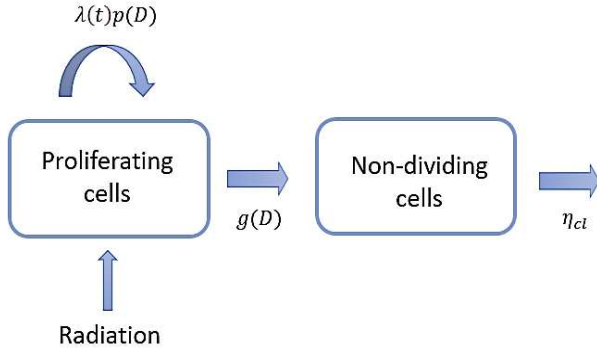


Figure 2: The effect of radiation on dividing cells

Figure 2 illustrates the effect of radiation on dividing cells. When dividing cells are exposed to radiation D , a certain number of cells continue to divide at a tumor growth rate $\lambda(t)$ per time t with a cell proliferation probability $p(D)$, while others stop dividing with probability $g(D)$.

Cells that do not divide are later removed from the organism at the cell removal rate η_{cl} . Cancer consists of dividing and non-dividing cells, therefore the total volume of cancer is obtained as the sum of the volume of the part of the dividing tumor cells $V_T(t)$ and the volume of the part of the non-dividing tumor cells $V_{ND}(t)$.

$$V = V_T(t) + V_{ND}(t)$$

This model belongs to the group of models based on ordinary differential equations (ODE). Then the volume of proliferating cells, as well as the tumor growth rate at time t can be obtained as

$$\frac{dV_T}{dt} = \lambda(t)V_T$$

$$\frac{d\lambda}{dt} = -\theta\lambda(0)\lambda$$

where θ is the retardation factor of the vascular structure, and $\lambda(0)$ is the initial tumor growth rate.

Tumor behavior can be observed during and after the active effect of radiation. Let t_R be the initial moment of radiation and t_{rad} be the time period of active radiation effect.

Let $t_R \leq t \leq t_R + t_{rad}$, then

$$\frac{dV_T}{dt} = \lambda(t)p(D)V_T - g(D)V_T$$

represents the volume of proliferating cells,

$$\frac{dV_{ND}}{dt} = g(D)V_T - \eta_{cl}V_{ND}$$

represents the volume of non-dividing cells,

$$\frac{d\lambda}{dt} = -\theta\lambda(0)\lambda$$

tumor growth rate.

Let $t > t_R + t_{rad}$, then

$$\frac{dV_T}{dt} = \lambda(t)V_T$$

represents the volume of proliferating cells,

$$\frac{dV_{ND}}{dt} = -\eta_{cl}V_{ND}$$

represents the volume of non-dividing cells,

$$\frac{d\lambda}{dt} = -\theta\lambda(0)\lambda$$

tumor growth rate.

There is constant $\chi(D)$ that depends on the radiation dose D with parameters α, β chosen so that the function best fits the data obtained

$$\chi(D) = \alpha D \left(1 + \frac{D}{\alpha/\beta}\right)$$

or

$$\chi(D) = \alpha D + \beta D^2$$

When radiation affects tumor cells, the cells either die or survive. The rate of surviving cells is defined as

$$S(D) = e^{-\chi(D)} = e^{-(\alpha D + \beta D^2)}$$

Then the tumor volume of dividing cells and non-dividing cells can be observed in the time before radiation t_- and in the time after radiation t_+

$$V_T(t_+) = S(D)V_T(t_-)$$

$$V_{ND}(t_+) = (1 - S(D))V_T(t_-)$$

Each individual cell will continue to divide with probability p or will not divide with probability $q = 1 - p$. Then the number of dividing cells N_D and the number of non-dividing cells N_{ND} at time T^* is obtained as

$$\frac{dN_D}{dt} = \frac{2p-q}{T^*} N_D \quad (5)$$

$$\frac{dN_{ND}}{dt} = \frac{q}{T^*} N_D \quad (6)$$

where initial condition values are $N_D(0)=N_0$ and $N_{ND}(0)=0$.

The solutions of equations (5,6) is found as

$$N_D(t) = N_0 \exp\left(\frac{2p-q}{T^*} t\right)$$

$$N_{ND}(t) = \frac{q}{2p-q} N_0 \exp\left(\frac{2p-q}{T^*} t - 1\right)$$

Also, connection can be made between the number of dividing cells at time T_m before radiation and time T^* after radiation as

$$\exp\left(\frac{2p-q}{T^*} T_m\right) \exp\left(-\frac{2}{T^*} T_m\right) = S(D) = e^{-\chi(D)} \quad (7)$$

The solutions of equations (7) is found as

$$p(D) = 1 - \frac{T^*}{3T_m} \chi(D) \quad (8)$$

$$g(D) = \frac{\chi(D)}{3T_m} \quad (9)$$

As the period of cancer death cells lasts t_{rad} time, the number of proliferating cells that will die in t_{rad} time can be approximated to the survival fraction by $S(D)$

$$\int_{t_-}^{t_- + t_{rad}} g(D) N_D(t) dt \approx (1 - S(D)) N_D(t_-)$$

3 DESCRIPTION OF THE MODEL IN THE MATLAB ENVIRONMENT

The following mathematical models will be presented in MATLAB.

3.1 Gompertz model

The Gompertz model of tumor growth begins by defining the initial parameters: V_0 which represents the initial volume of the tumor in mm^3 , a which represents the initial cell proliferation rate in mm^3/day and b which represents the decrease proliferation cell rate in mm^3/day .

```
% Gompertz model
% Parameters are defined
V0 = 2162; % Initial tumor volume
a = 0.0005; % Initial cell proliferation rate
b = 0.001; % Decrease proliferation cell rate
% Time span is defined
tspan = [0,3650];
% ODE function is defined
dVdt = @(t,V) a * V * exp(-b * t);
% The system of ODEs is solved
[t,V] = ode45(dVdt, tspan, V0);
% Tumor volume is plotted
plot(t,V);
xlabel('Time (day)');
ylabel('Tumor volume (mm^3)');
title('Gompertz model of tumor growth')
```

Figure 3: Code for the Gompertz model in MATLAB

Then, the time period for which the growth of the tumor is monitored is defined, as well as the ODE (equation (4)). By solving equation (4), the volume of the tumor is determined. As the tumor volume is observed in a certain time interval, it is necessary to solve the system of ODE equations. It is best to use the *ode45* function, since it is a first-order ODE system. Finally, the obtained results are presented in a coordinate system in which the x-axis represents the time moments in which the tumor volume is observed and the y-axis represents the volume of the tumor.

3.2 Implementation on the benign tumor

The growth of a benign tumor (type of meningioma) located in the middle cranial fossa will be shown in the following section.

The MRI image shows that the initial volume is $V_0 = 2162$, $a = 0.0005$ is the initial rate of tumor proliferation and $b = 0.001$ is decrease proliferation cell rate¹.

In the Figure 5 tumor growth for $t = [0,3650]$ is shown. The model shows that the volume has logarithmic growth and that volume of the tumor after 10 years will be $V = 3518.51 mm^3$.

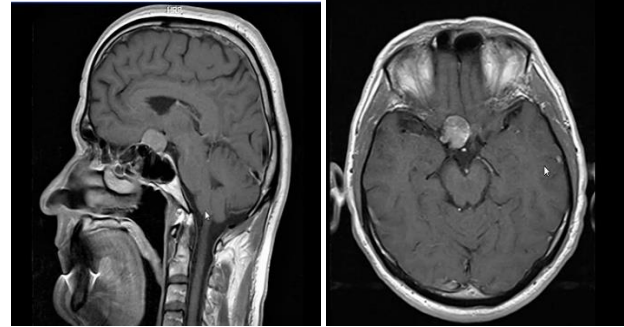


Figure 4: MRI image of a benign tumor

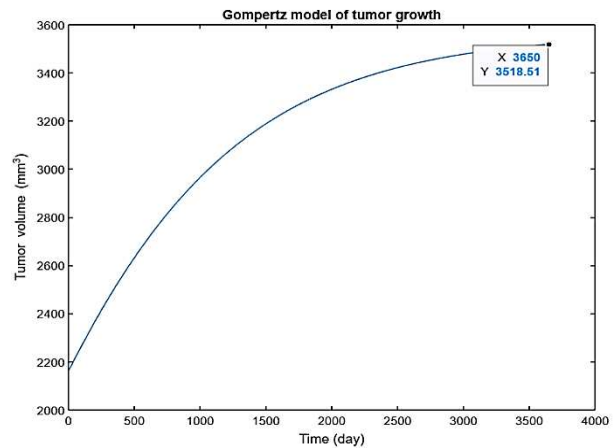
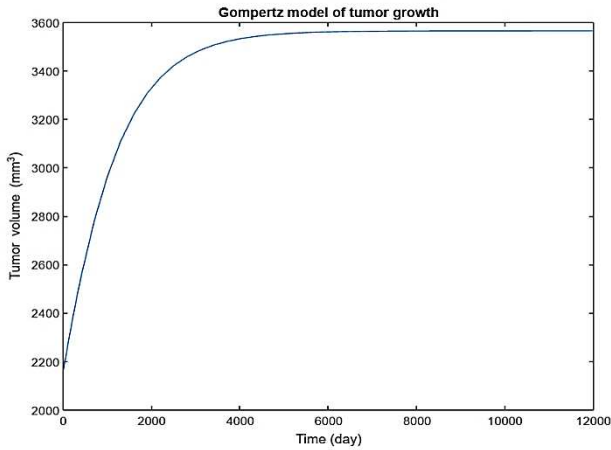


Figure 5: Graphic representation of tumor growth for $t = 3650$

¹ The parameters a and b were not obtained from the MRI scanning. Based on the history of the disease and research on meningioma type tumors, it is assumed that the parameters a and b have the given values.


 Figure 6: Graphic representation of tumor growth when $t \rightarrow \infty$

The tumor grows to its maximum size K which is determined by the growth rate, the location of the tumor and the tumor vasculature.

As the capacity K is on the asymptote of the function, from the graph in the Figure 6 it can be seen that $V \rightarrow 3564 \text{ mm}^3$, which if compared with the value obtained using the formula

$$K = V_0 e^{\frac{a}{\beta}} = 2162 * e^{\frac{0.0005}{0.001}} = 3564.54 \text{ mm}^3$$

is a good approximation of the proposed tumor growth over time.

In observing the growth of a tumor, we have to pay attention to the time it takes for the tumor to reach its double volume compared to the initial observation. This information can be important when making a decision about the method of treatment. The time it takes for a tumor to double its volume can indicate how fast the tumor is growing. As the observed tumor is located in the middle cranial fossa, due to its location and the insufficient amount of nutrients available to the tumor, regardless of the passage of time, the tumor will not reach its double volume, which can be concluded from the graph presented in the Figure 6.

3.3 Kinetic model

Kinetic model for monitoring tumor growth begins by setting the initial parameters: D which represents the radiation dose, θ which represents the vascular growth retardation factor, λ_0 which represents the cancer growth rate at the initial moment, TR which represents radiation start time, $trad$ representing the radiation time, $Tstar$ representing the time instant after radiation, Tm represents the time instant before radiation.

The proposed mathematical models can make predictions whether the predictions are made on a smaller or larger sample of data. A large sample would imply that the parameters at moments $t_{i_1}, t_{i_2}, t_{i_3}, \dots, t_{i_k}$ are known where $0 \leq i_1 < i_2 < \dots < i_k < \infty$. Due to the uniqueness of each individual sample, what can happen is that the predictions of the proposed model do not match the input data. In order to reduce the approximation error, the parameters α and β are introduced. The

parameters α and β represent the parameters taken to fit the model to the data obtained from the MRI scans.

```
% Parameters are defined
D = 6; % Radiation dose
theta = 0.2; % Vascular growth retardation factor
lambda_0 = 0.1; % Initial tumor growth rate
TR = 2; % Time of radiation
trad = 1; % Active radiation-effect time
Tstar = 5; % Time instant after trad
Tm = 1; % Time instant before TR
alpha = 0.01; % Parameter alpha
beta = 0.01; % Parameter beta

% Functions for p(D) and g(D) are defined
chi = alpha*D + beta*D^2;
p = @(D) 1 - (Tstar/(3*Tm))*chi;
g = @(D) chi/(3*Tm);

% ODEs are defined
ode_fun = @(t, y) tumor_volume_ode(t, y, D, ...
    lambda_0, theta, p, g, TR, trad);

% Initial conditions are set
VT0 = 67759.26; % Initial volume of proliferating
                % tumor
VND0 = 0; % Initial volume of non-dividing cells
lambda0 = lambda_0;

% ODEs are solved
tspan = [0 TR+trad+5]; % Time interval
y0 = [VT0; VND0; lambda0]; % Initial conditions
[t, y] = ode45(ode_fun, tspan, y0);

% The results are plotted
figure;
subplot(2,2,1);
plot(t, y(:,1));
xlabel('Time (days)');
ylabel('VT(t)');
title('Proliferating tumor volume');

subplot(2,2,2);
plot(t, y(:,2));
xlabel('Time (days)');
ylabel('VND(t)');
title('Non-dividing tumor volume');

subplot(2,2,3);
plot(t, y(:,3));
xlabel('Time (days)');
ylabel('lambda(t)');
title('Tumor growth rate');

subplot(2,2,4);
plot(t, p(D)*y(:,1));
hold on;
plot(t, g(D)*y(:,1));
xlabel('Time (days)');
ylabel('Cell transition rate');
legend('Dividing cells', 'Non-dividing cells');
```

Figure 7: The first part of the kinetic model code in MATLAB

```

function dydt = tumor_volume_ode(t, y, D, ...
    lambda0, theta, p, g, TR, trad)
    VT = y(1); % Proliferating tumor volume
    VND = y(2); % Non-dividing tumor volume
    lambda = y(3); % tumor growth rate

    if t >= TR && t < TR + trad
        dp = p(D);
        dg = g(D);
    else
        dp = 0;
        dg = 0;
    end

    dVTdt = lambda*p(D)*VT - g(D)*VT;
    dVNDdt = g(D)*VT - theta*VND;
    dlambda_dt = -theta*lambda0*lambda;

    dydt = [dVTdt; dVNDdt; dlambda_dt];
end
    
```

Figure 8: The second part of the kinetic model code in MATLAB

Then, the functions $p(D)$ and $g(D)$ (equations (8) and (9)) are defined and the initial conditions are set.

$VT0$ represents the total initial volume of dividing cells, and $VND0$ represents the total initial volume of non-dividing cells. After setting the time interval on which the cancer behavior is observed, the ODE system is solved using the *ode45* function.

As shown in Section 2.2, two cases are considered: when $t_R \leq t \leq t_R + t_{rad}$ and when $t > t_R + t_{rad}$. It can be noticed that when $t > t_R + t_{rad}$ probability is $p(D) = 0$ and $g(D) = 0$. Therefore, it is necessary to create a function that takes one value of the functions for $p(D)$ and $g(D)$ when $t > t_R + t_{rad}$ and the other in other cases.

Making a decision on what amount of radiation is the most suitable for a patient is extremely important. In order to make the best possible decision, it is necessary to create a model that compares the effects of different amounts of radiation on cancer. Having this information available allows doctors to make a decision at what time and for what amount to irradiate patient. These models are the most suitable when working with a large database, but they can also be used on a smaller database. In order for the model to make better predictions, what needs to be done is to monitor changes in tumor volume and enter new data. This would reduce the error made by the model.

The code presented in Figure 7 and 8 calculates the influence of one dose of irradiation on tumor volume, while the following code compares different doses of radiation on tumor volume.

As these codes are built on the same mathematical model, values of parameters $\theta, \lambda, TR, trad, Tstar, Tm, \alpha, \beta, VT0, VND0, \lambda_0$ as well as the definition of functions $p(D), g(D), VT, VND$ the only difference is that for the value of D is taken a list of values.

The code starts by setting the initial conditions. As the tumor volume must be calculated for each radiation dose D at each time instant t , a function must be created that will solve those systems of equations. A *for* loop is used so that the function can take each value from list D . Then the functions $p(D)$ and $g(D)$ are defined and solved so that the function takes one value when

$t_R \leq t \leq t_R + t_{rad}$ and another when $t > t_R + t_{rad}$. Then the obtained data are presented on the same graph. Time t is represented on the x-axis and tumor volume is represented on the y-axis.

```

% Parameters are defined
Tm = 1; % Time instant before TR
TR = 2; % Time of radiation
trad = 1; % Active radiation-effect time
Tstar = 5; % Time instant after trad
theta = 0.2; % Vascular growth retardation factor
alpha = 0.01; % Parameter alpha
beta = 0.01; % Parameter beta
eta_c1 = 0.1; % Cell removal rate
VT0 = 67759.26; % Initial volume of non-dividing cells
D_values = [5, 5.5, 6, 6.5]; % Radiation doses

% Time interval and initial conditions are set
tspan = [0, 20];
VT_init = VT0;
VND_init = 0;
lambda_init = 0.1;

% ODEs are solved for each radiation dose
for i = 1:length(D_values)
    D = D_values(i);

    % Functions for p(D) and g(D) are defined
    chi = @(D) alpha*D + beta*D^2;
    p = @(D) 1 - (Tstar/(3*Tm))*chi(D);
    g = @(D) chi(D)/(3*Tm);

    % ODEs for TR <= t < TR + trad are defined
    ode1 = @(t, y) [y(1)*lambda_init*p(D) -
        g(D)*y(1); g(D)*y(1) - eta_c1*y(2);
        -theta*lambda_init*y(3)];

    % ODEs for TR <= t < TR + trad are solved
    [t1, y1] = ode45(ode1, [TR, TR+trad], ...
        [VT_init, VND_init, lambda_init]);

    % ODEs for t > TR + trad are defined
    ode2 = @(t, y) [y(1)*y(3); -eta_c1*y(2);
        -theta*lambda_init*y(3)];

    % ODEs for t > TR + trad are solved
    [t2, y2] = ode45(ode2, [TR+trad, tspan(2)], ...
        [y1(end,1), y1(end,2), lambda_init]);

    % Solutions from both time intervals are
    % concatenated
    t = [t1; t2];
    y = [y1; y2];

    % The results are plotted
    plot(t, y(:,1));
    hold on;
end

legend('D = 5', 'D = 5.5', 'D = 6', 'D = 6.5');
xlabel('Time (days)');
ylabel('Tumor volume (mm^3)');
title(['Tumor volume in dependence of different ' ...
    'radiation doses']);
    
```

Figure 9: Kinetic model code for observing changes in cancer volume in relation to the amount of radiation in MATLAB

3.4 Implementation on the malignant tumor

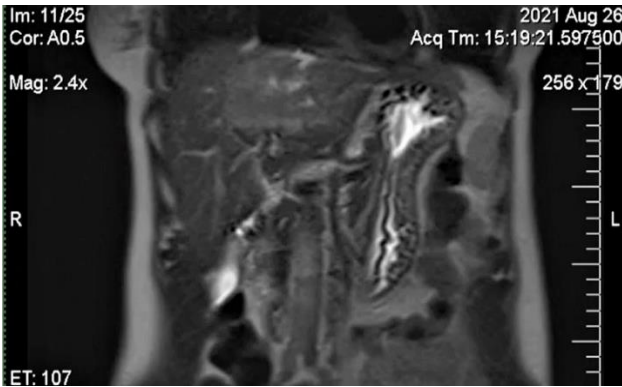


Figure 10: MRI image of a malignant tumor

The growth of a malignant tumor (type of hepatocellular carcinoma) located in the left lobe of the liver will be shown in the following section.

The MRI scan shows that the cancer volume is $V_T = 67759.26 \text{ mm}^3$, as this value is taken in initial moment $t = 0$, it is concluded that $V_{ND}=0$. The parameters are defined as : $D = 6\text{Gy}^2$ radiation dose, $\theta = 0.2$ the retardation factor of the vascular structure, $\lambda(0) = 0.01$ rate of cancer growth, $t_R = 2$ radiation start time, $t_{rad} = 1$ radiation time, $T^* = 5$ time instant after radiation, T_m time instant before radiation, $\alpha = 0.01$, $\beta = 0.01$, $\eta_{cl} = 0.1^3$.

At the moment $t = 0$ and $V_T = 67759.26 \text{ mm}^3$ as the cancer was not affected by radiation or chemotherapy. Due to the effect of radiation, cells that divide change into cells that do not divide, and then they die and get removed from the body. Therefore, from the Figure 11 it can be se-

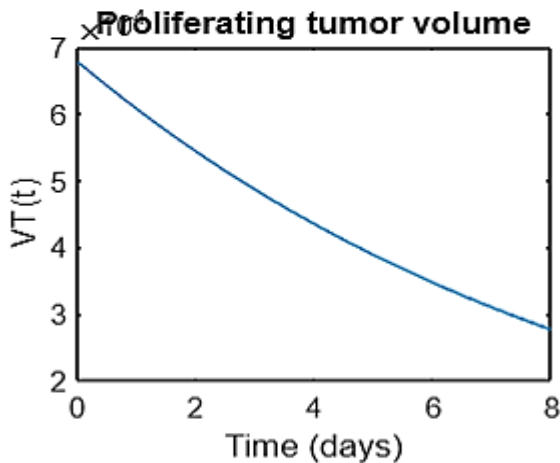


Figure 11: Graphic representation of the total volume of the proliferating cells

en how the total volume of proliferating cells decreases during the period of active radiation time effect, where as seen on the Figure 12 the total volume of non-dividing cells increases.

As radiation affects the blood vessels that bring nutrients to the cancer, disrupting those connections slows tumor growth. As can be seen from the graph (Figure 13), the tumor growth rate is decreasing.

On the graph shown in Figure 14 it can be seen how the cell transition rate is obtained. The function that describes dividing cells as well as the function that describes non-dividing cells are convex.. The number of dividing cells is decreasing, because the radiation has affected the cells. Although the number of non-dividing cells should be increasing, because dividing cells are transitioning into non-dividing cells, the number of cells that died is greater than the number of non-dividing cells, therefore our function decreases. If it's kept up with the radiations and the graph is plotted as $t \rightarrow \infty$, the number of non-dividing cells and the number of dividing cells would approach the x-axis.

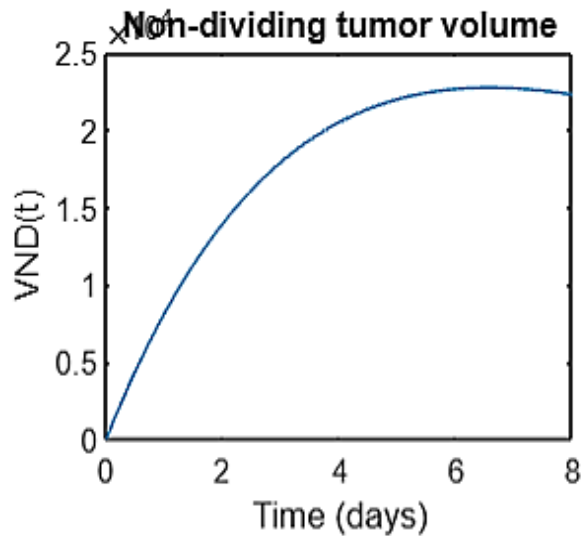


Figure 12: Graphic representation of the total volume of the non-dividing cells

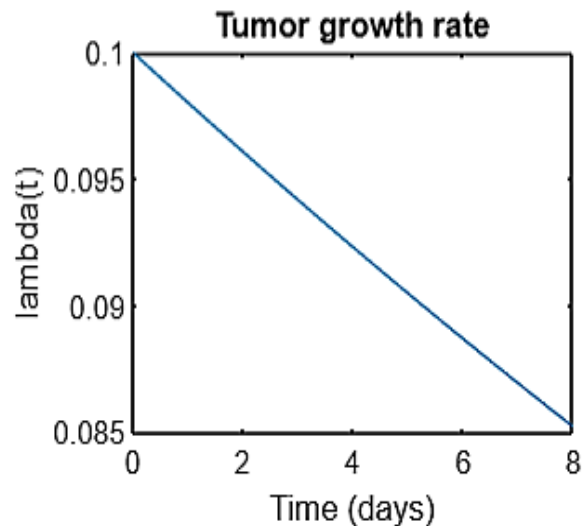


Figure 13: Graphic representation of the growth rate

² Gray (Gy) is the SI unit of the absorbed dose of ionizing radiation by the body (the amount of energy absorbed by the body under the influence of radiation) [5].

³ Note: The parameters are not obtained from the MRI scan. Based on the medical history and research on hepatocellular carcinoma, it is assumed that the parameters have given values [2].

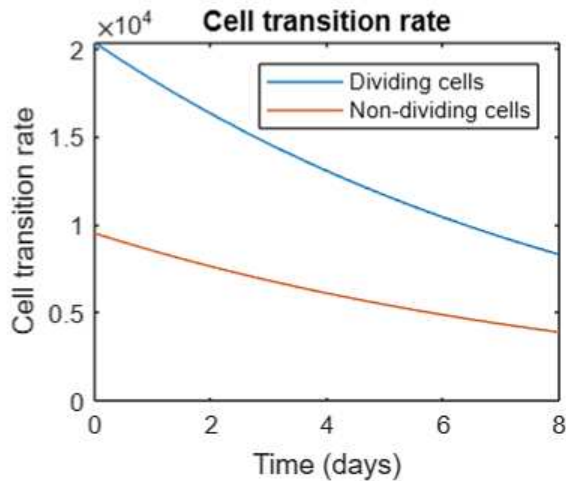


Figure 14: Graphical representation of cell transition rates from dividing to non-dividing cells

The dose of radiation to which a patient is exposed to depends on the type, stage and location of the cancer, the patient's age, previous medical history, etc.

There are two approaches: exposure to lower doses of radiation more times or exposure to higher doses of radiation fewer times.

On the graphic shown in Figure 15 it can be seen that the dose of 6.5 Gy has the greatest impact. For all 4 values, it can be seen that they have the greatest decrease during the active radiation effect, after which the volume is on the rise again. This happens because the cancer was treated with only 1 dose of radiation. For the graph to continue decreasing, multiple exposure to radiation would be necessary as well as chemotherapy.

Of course, higher dose of radiation does not mean a better outcome for the patient. Since radiation affects healthy cells in addition to malignant cells, it must be taken into account not to damage too many healthy structures. If the immune system is damaged, regardless of how much of the tumor is killed, body will not be able to fight the disease.

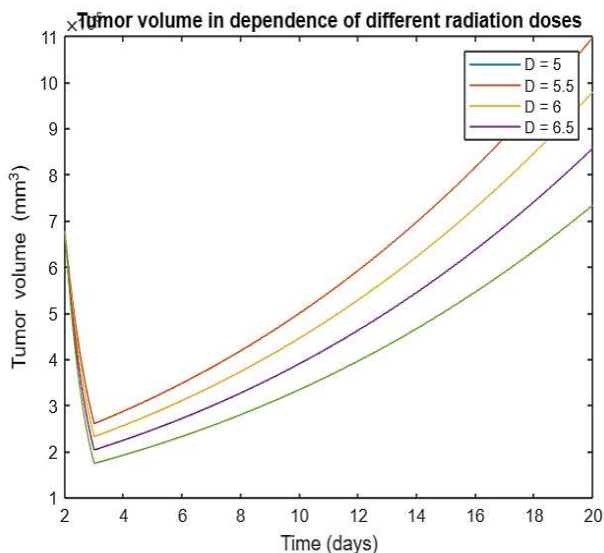


Figure 15: Graphical representation of the tumor volume in dependence of different radiation doses

4 CONCLUSION

This paper presents mathematical models that help to monitor the growth of benign and malignant tumors. It is shown that the proposed mathematical models can be used to monitor tumor growth, as well as how much influence the initial parameters have on the final outcome. The model can be adapted to follow avascular and vascular growth. Mathematical models of tumor growth are most often used during experimental studies of tumor growth (on animals) and the effect of radiation on them. A larger database improves the accuracy of the obtained data. This paper showed that the proposed models can be used to monitor tumor growth in humans as well. As the models were applied to a smaller sample, the reliability of the obtained results is debatable. In order to ensure the highest possible reliability of the obtained results, it is necessary to increase the sample size and observe tumor growth at shorter time intervals. Also, it would be desirable to introduce a confidence interval estimation that would describe the reliability of each obtained data.

5 SUMMARY

As cancer is one of the deadliest diseases, models have been developed that are used to predict tumor growth. Those are mathematical models which connect mathematical equations, probability and computer science. In this paper Gompertz and kinetic models are presented. Although these models are usually used in animal experiments and with larger database, in the paper it is shown that it can be used on humans and with smaller database. The Gompertz model was implemented on a benign tumor. It is shown how it will grow throughout time and its maximum volume. Two codes were written for a kinetic model that were implemented on a malignant tumor. It was shown what affect one dose of irradiation has on the tumor as well as how different doses of radiation affect the tumor. In order to improve the reliability of the data, it is necessary to work on a larger data base, shorter observation intervals and it would be desirable to introduce a confidence interval estimation that would describe the reliability of each obtained data.

ACKNOWLEDGEMENT

My most profound appreciation goes to Doc. dr Kruna Ratković, my mentor for her time, effort and support throughout my research. The part of this work was previously presented at the University of Donja Gorica as a part of the Research project subject. Also, I would like to thank my parents for helping me find data for my research as well as people who gave me access to their medical records and consent to the use of their personal data in this work. Finally, I want to thank my sisters and friends for their love, encouragement and support.

REFERENCES

- [1] American Cancer Society: [How Radiation Therapy Is Used to Treat Cancer | American Cancer Society](#)
- [2] Bang, A., Dawson, L.A (2019). Radiotherapy for HCC: Ready for prime time? *J Hep Reports* 1(2): [Radiotherapy for HCC: Ready for prime time? - ScienceDirect](#)
- [3] Benzekry, S., Lamont, C., Beheshtii, A., Tracz, A., Ebos, J.M., Hlatky, L., Hahnfeldt, P. (2014). Classical mathematical models

- for description and prediction of experimental tumor growth. *PLoS computational biology* 10(8): [Classical Mathematical Models for Description and Prediction of Experimental Tumor Growth | PLOS Computational Biology](#)
- [4] Cooper, G.M (1992).Elements of human cancer.Jones & Bartlett Learning: [Elements of Human Cancer - Geoffrey M. Cooper - Google Књиге](#)
- [5] Crnjanac, F. (2021).SLIKOVNE METODE U UROLOGIJI. Ph.D. thesis, University of Rijeka. Faculty of Medicine. Department of Urology :[.SLIKOVNE METODE U UROLOGIJI... - Google Академик](#)
- [6] Marušić, M. (1996).Mathematical models of tumor growth. *Mathematical Communications*1(2) :[\(PDF\) Mathematical models of tumor growth \(researchgate.net\)](#)
- [7] Stanojević, M. (2019).Matematički modeli rasta tumora :[masStanojevicMirjana.pdf \(bg.ac.rs\)](#)
- [8] Watanabe, Y., Dahlman, E.L., Leder, K.Z., Hui, S.K. (2016).A mathematical model of tumor growth and its response to single irradiation. *Theoretical Biology and Mathematical Modeling*13: [A mathematical model of tumor growth and its response to single irradiation - PMC \(nih.gov\)](#)
- [9] World Cancer Research Fund International (2020): [Worldwide cancer data | World Cancer Research Fund International \(wcrf.org\)](#)

Fuzzy Logic - Based control for autonomous vehicles in urban environments (application description)

Enisa Trubljanin

University of Donja Gorica (Faculty of Applied Sciences, Applied Mathematics)

Podgorica, Montenegro

enisa.trubljanin@udg.edu.me

<https://www.udg.edu.me/>

Abstract. Fuzzy logic is a mathematical approach used to model uncertainty in decision-making processes. Fuzzy logic has been widely applied in fields where uncertainties are encountered, such as automatic control, artificial intelligence, shape recognition system, robotics and other fields. Autonomous vehicles are vehicles that are able to move and function without the direct control of the driver. This paper shows how fuzzy logic makes decisions, what are the advantages and disadvantages of such a driving system in urban areas, the results of fuzzy logic "inference" and how fuzzy logic makes decisions instead of humans on the basis of forwarded information from the environment, and specifically the focus is on making decisions about speed and registering obstacles. These results are quite good, but it turns out that they still cannot completely replace humans, especially when it comes to some ethical issues.

Keywords: Autonomous Vehicle, Fuzzy Logic, Obstacle detection

1. INTRODUCTION

In recent decades, the development of autonomous vehicles has become one of the greatest achievements in the field of transportation. This innovative technology

opens up new perspectives for the future of transportation, especially in urban environments. Driving autonomous vehicles in urban environments is a complex challenge due to the presence of numerous variable factors such as traffic congestion, pedestrians, weather conditions, traffic signs and signaling. Fuzzy logic provides a suitable framework for driving autonomous vehicles in such complex environments. Basically, fuzzy logic relies on linguistic variables and inference rules to make decisions based on imprecise data [1]. This flexible and adaptive logic allows autonomous vehicles to behave according to changing conditions in urban environments. Using this logic, the aim is to make the autonomous vehicle as independent as possible from the driver and his need to react. The paper focuses on understanding the challenges of autonomous vehicles in urban environments and exploring ways in which fuzzy logic can solve these challenges. In the first part of the paper, the concept of fuzzy logic and its importance in the context of autonomous vehicles is explained. Also, the basics of fuzzy logic and how it can be applied in the management of autonomous vehicles are presented, with special reference to the urban environment. Safety and reliability of autonomous vehicles are also key aspects in this paper. The paper presents concrete examples of the application of fuzzy logic in driving autonomous vehicles in urban areas and how, with the help of this logic and other parts of the system, the vehicle makes a decision to change the speed or to notice the obstacles it encounters. The central place in the paper is occupied by fuzzy logic for the detection of obstacles and speed

control in the urban environment, the description and application of fuzzy logic in this process, examples and results, and examples of the implementation made in the Python programming language. Through all these analyzes and examples, the goal is to understand how fuzzy logic can be useful in driving autonomous vehicles in urban environments. The utilization of fuzzy logic technology holds the potential to enhance driving safety, efficiency, and comfort in urban areas by enabling intelligent decision-making based on imprecise data. This technology paves the way for intelligent, autonomous transportation that can adapt driving to different situations and contribute to the sustainability and efficiency of urban transportation systems. The International Federations of Robotics concluded that there have been 19% increase in the use of robots in the vehicle industry with a statistics of 86 200 to 106 300 units in just a space of three years 2004 – 2007 [2], and today these percentages have increased significantly.

2. FUZZY LOGIC

Fuzzy logic is based on clear and precisely determined rules, and rests on set theory. It allows modeling complex systems with uncertain or unclear inputs and making reasonable decisions based on those inputs [3]. An element may or may not belong to a set. Sets have clearly defined boundaries. In fuzzy logic, the belonging of one element to a certain set is not precisely defined, but the belonging is measured in, say, percentages [4]. These membership measures, scaled, can take values from 0 to 1. We use very imprecise sentences every day and they are not useful in fuzzy logic, for this reason we have to give up the binary decision in which something is true or false or black or white and turn it all into fog and "gray color". This logic is very close to the human perception of many things in life. It finds its application in many areas, such as autonomous vehicles systems. Many similar situations that are not clearly separated, which are a mixture of several things, are present around us every day, and fuzzy logic measures the truth of each statement in percentages. Fuzzy sets are the basic elements with which we describe imprecision. A discrete set contains elements with the same properties, while a fuzzy set contains elements with similar properties. The essence is to form a program code that implements the knowledge of an expert about some processes [5]. This system can be seen as a black box in which certain outputs are formed based on the inputs based on the dependence of the input units and the relationships between them.

3. HOW DOES AN AUTONOMOUS VEHICLE WORK?

Driverless vehicles are getting closer to reality. Every major vehicle manufacturer has its own project related to autonomous driving. Some components are already in use today, such as self-parking, automatic braking or cruise control. Autonomous driving technology is not simple. They rely on sensors that collect information from the environment. They use various technologies, such as sensors, cameras, radars, GPS and artificial intelligence, to make real-time driving and navigation decisions. That information is then sent to a very powerful computer unit that processes it [6]. Extremely accurate maps and an internal GPS system are inserted into the computer memory. The sensors use ultraviolet and infrared rays. In addition to sensors, cameras and lidar (Light Detecting and Ranging) are also used to collect information. Lidar is an optical measuring instrument that uses 32 or 64 lasers to measure the distance between objects. This information is needed to create a virtual 3D map [7]. Sensors, cameras and lidar can "see" stationary and moving objects, such as people, animals, obstacles on the road, other vehicles, read traffic signs. The processor processes the data generated in this way, creates a visual image and combines it with preloaded maps.

3.1 Challenges for autonomous vehicles

There are many challenges of operating autonomous vehicles in urban environments and obstacles that these vehicles need to overcome without consequence. Traffic congestion is one of them because urban environments are often characterized by heavy traffic and congestion that can affect the speed of vehicle movement. Autonomous vehicles must be able to adapt to these conditions, making decisions about changing speed, avoiding obstacles and optimizing the path [8]. Then caution towards pedestrians and their unpredictable movements. It is necessary to identify pedestrians, predict their movement and adjust driving in order to ensure the safety of all road users. In this context, fuzzy logic can be useful in analyzing sensor data to identify pedestrians. Just as cars that require a driver and must obey traffic signs, traffic signs and signaling represent a special type of challenge for autonomous vehicles. Correctly interpreting and reacting to traffic signs and signaling is key to safe driving in urban environments. Fuzzy logic can be applied to model and interpret various traffic sign and signaling situations, allowing

autonomous vehicles to recognize and properly respond to changes in those situations. Weather conditions, such as rain, snow or fog, can significantly affect driving conditions and require autonomous vehicle driving to be adjusted. Fuzzy logic can be applied to model the effect of weather conditions on vehicle performance and make decisions about optimal speed, braking or steering according to those conditions. Then the traffic flow in urban environments is often variable and unpredictable. Autonomous vehicles must be able to adapt to changes in traffic speed and density to maintain safety and efficiency. All these are challenges that an autonomous vehicle must successfully overcome.

3.2 Advantages and disadvantages of autonomous vehicles

It is almost difficult to talk about the balance between the advantages and disadvantages of such vehicles. One of the reasons is the raising of many ethical questions, so there are differences of opinion in the analysis of those questions. Perhaps the biggest advantage of autonomous vehicles is that you don't need any driving knowledge or a driver's license to own and operate one. Also, autonomous vehicles are programmed to follow traffic regulations and rules that will prevent human error. These vehicles are equipped with sensors that enable quick reactions in critical situations, which a person would not be able to do. Autonomous vehicles can use advanced algorithms for routing and path planning, optimizing traffic flow and reducing congestion. They can communicate with each other and synchronize their movements, thereby reducing road congestion and increasing the overall efficiency of the transportation system. On the other hand, sensor precision, software reliability and solving unpredictable traffic situations have not yet reached high accuracy. Vehicle automation may now be at a transitional level between partial and full automation, but in the coming years, cars will certainly be able to drive independently. Another disadvantage of these vehicles is setting the level of privacy, which is otherwise a huge problem nowadays. Autonomous vehicles generate large amounts of data about users' journeys, locations and habits. This raises issues of privacy and data security, as well as the need for adequate protection of user data. Autonomous vehicles require a high investment in the development, research and implementation of technologies. Currently, the costs of autonomous vehicles are high, making them inaccessible to a wide range of consumers. Also, maintenance and repairs of autonomous systems can also be expensive and demanding. As technology today leads to the loss of

many jobs, driving autonomous vehicles can have a negative impact on jobs that are related to driving, such as taxi, truck or bus drivers.

4. FUZZY LOGIC FOR OBSTACLE DETECTION AND SPEED CONTROL IN AN URBAN ENVIRONMENT

In the previous part of the paper, it was shown how fuzzy logic is present in almost the complete system that was created for the functioning of an autonomous vehicle. In the rest of the paper, the emphasis is on the example of the programmer's implementation of fuzzy logic for the management of autonomous vehicles in urban areas. It is described how to build a system for detecting obstacles and making decisions about avoiding collisions, as well as about vehicle speed control.

The vehicle is equipped with sensors that collect data about the environment, including the distance to other vehicles, pedestrians, obstacles and traffic signs (*Figure 1*). Fuzzy logic is used to process that data and make decisions based on a set of rules. For the implementation that was done, a work-generated dataset was used, with the parameters registered by the mentioned sensors. For this analysis, it is not important that the data are correct, but the emphasis is on fuzzy decision-making based on the transmitted information from sensors located on autonomous cars. Inputs are described using fuzzy sets. Fuzzy set A has a membership function (μ_A) where for each value x from the domain of the phase variable X, a membership to phase set A is assigned, which is a number in the range from 0 to 1 (*1*).

$$\mu_A(x) : X \rightarrow [0,1] \quad (1)$$

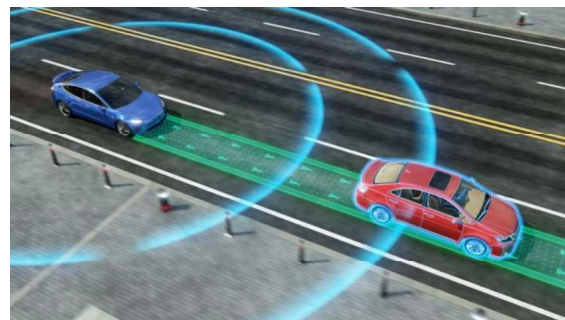


Figure 1: Display of data collection using sensors

[9]

The distance to the obstacle can have vague sets of "close", "moderately close" and "far". Then the rules are

defined that connect these inputs with the outputs of the system. If the distance to the obstacle is "close" and the vehicle speed is "high", then a decision can be made to brake the vehicle to avoid a collision. After defining inputs, outputs and rules, a fuzzy logic inference process is used to make decisions. This process uses rules and input data to make inferences about how the vehicle should be operated in a given situation. Decisions can be made based on a combination of multiple inputs and rules using fuzzy logic methods, such as maximum or minimum. The principle of maximum is used to combine inputs when the highest value of the input is to be emphasized, while for minimum it is the opposite, the minimum value of the input is emphasized. After the decisions have been made, the outputs are translated back into concrete actions taken by the vehicle. In case the decision is to brake the vehicle, appropriate action is taken to stop the vehicle or reduce speed. All the decisions made are further sent through the system, where the vehicle is enabled to react quickly by executing the order that is the result of fuzzy reasoning.

Below (Figure 2) is shown the output of the Python code that enables the input of the parameters registered by the sensors. In order to demonstrate how fuzzy logic makes conclusions in the given example, only two variables were observed, speed and distance, and of course in practice for making a decision with a vehicle it is necessary to observe more factors that influence the decision making, which were mentioned earlier in the paper. From the graph (Figure 2), it is possible to estimate how far the input value for distance belongs to the set "close" or "far" by looking at the values on the Y-axis at the point where the X-axis is on that value. Similarly, how far the output value belongs to the "decrease" or "obstacle" set by looking at the values on the Y-axis. The colored/shaded part on the graph represents the intensity area of belonging to a certain set or the resulting set. This is one of the ways of forming fuzzy sets, where in further actions the belonging to the set of all the variables is observed. This part of the graph indicates how high or low a given membership value is for a given set. The colored part is useful because it allows to visually identify the area in which there is a certain value or range of values for the intensity of affiliation. In this way, it can be estimated to which category a certain value belongs based on the intensity of belonging in that area. Drawing a conclusion from the results discussed in this way is precisely the task of fuzzy logic.

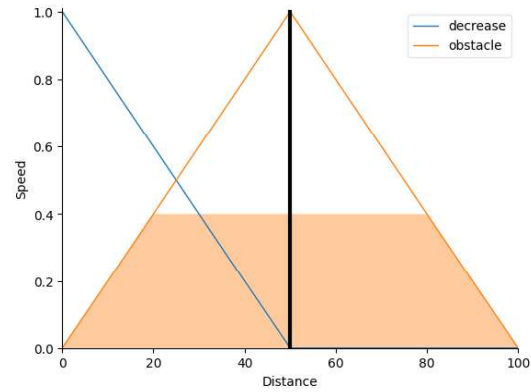


Figure 2: Vehicle speed control graph

The basic logic behind fuzzy-logic systems is the application of the operator's expert experience in designing a decision algorithm, whose input-output connections are represented by a collection of IF-THEN fuzzy rules that include fuzzy logic variables. In order to reach the final decision in the aforementioned analysis, this rule was used in the implementation as: IF the speed is low THEN set a high acceleration; which leads to the final decision on speed regulation. It is closely connected with fuzzy inference process which is formulated from fuzzy logic operators and fuzzy If-Then rules. To define consistency of given rules enables decision makers to select more important criteria. Researches on consistency for fuzzy control systems are still scarce [10]. By applying phase linguistic variables, the control rules phase and the approximate reasoning phase, it is possible to incorporate the operator's expert experience into the design of the control algorithm.

4.1 Connection between fuzzy logic and deep learning

Connecting fuzzy logic and deep learning in the context of autonomous vehicles can bring numerous advantages and open up new opportunities. Fuzzy logic is used for high-level decision making, while deep learning is used for low-level object recognition. Fuzzy logic can be used to make decisions about how an autonomous vehicle should react in a given situation based on a higher level of abstraction, while deep learning can recognize objects in the environment such as vehicles, pedestrians or obstacles (Figure 3). Deep learning enables the creation of different models that will provide information and results that fuzzy logic can use later to make its decisions. These results are then used as inputs to fuzzy logic for decision making.



Figure 3: Example of a learning model
[11]

By combining fuzzy logic and deep learning, one can harness the strengths and versatility of both methodologies. In this context, deep learning can be used to improve object detection, recognize traffic signs and signals, and predict the movement of vehicles and pedestrians (Figure 4). Fuzzy logic provides the ability to make decisions based on human knowledge and rules, while deep learning enables automatic learning of complex patterns and representations from large data sets. This combination can lead to improved performance and reliability of autonomous vehicles in urban environments. Also, this combination is excellent because it combines improvised human reactions and automatic reactions that deep learning models learn. The difference between static and dynamic environments in car object detection is another crucial aspect to consider. In car object detection, the term "static environment" refers to scenarios where the surrounding environment remains relatively unchanged over time. For instance, a parked car on a street or a car in a parking lot can be considered examples of objects in a static environment. In these cases, the primary challenge is accurately detecting and localizing the cars within the scene. On the other hand, a "dynamic environment" refers to scenarios where the surrounding environment is in constant flux.



Figure 4: Example of a learning model
[12]

This can include situations such as cars moving on a busy road, where the positions and orientations of cars

change rapidly. Detecting objects accurately in dynamic environments poses additional challenges due to motion blur, occlusions, and the need to track objects over time. In order to handle dynamic environments, advanced object detection algorithms incorporate motion modeling and tracking techniques. These techniques aim to predict the future location of objects based on their previous positions and motion patterns. By combining object detection with object tracking, autonomous navigation systems can handle dynamic environments more effectively. Furthermore, dynamic environments require real-time processing and decision-making capabilities to ensure the safe and efficient operation of autonomous vehicles. Processing speed becomes a critical factor in these scenarios, as the system needs to detect and respond to rapidly changing situations in a timely manner.

5. CONCLUSION

In this paper, the application of fuzzy logic in the context of the management of autonomous vehicles in urban environments was studied. The aim of the work is to show how fuzzy logic can be used in this area, then to point out the advantages and disadvantages of this type of transport. Fuzzy logic offers an effective approach for decision-making in the intricate situations that autonomous vehicles face in urban environments. Its connection with deep learning and artificial intelligence makes it a powerful tool for achieving safe and reliable control of autonomous vehicles. Deep learning can be combined with fuzzy logic to draw conclusions based on large amounts of data. This combination allows the autonomous vehicle to learn from experience and adapt to changing conditions in urban environments. It is important to note that the application of mathematics is key in fuzzy logic. Mathematical models and membership functions are used to describe and quantify uncertainty in data and decision making. Mathematical operations such as maximum and minimum are used to combine inputs and apply rules to produce output results. Fuzzy logic provides safety and reliability in the driving of autonomous vehicles, which is crucial for realizing more advanced and efficient transportation systems in urban areas. No one can yet predict when exactly automatic vehicles will take over. Software is powerless to answer driving dilemmas that are easy for humans and these are precisely the obstacles for which it is almost impossible to find a solution because this is the main difference between humans and robots. In the event that the vehicle cannot avoid a collision, is it better to hit a pedestrian or a tree on the side of the road? This significantly complicates the road to full automation of

vehicles. Despite the advancements in automation levels, reaching level three, which involves conditional driving automation, remains a partial achievement. The ongoing progress, driven by a combination of fuzzy logic and artificial intelligence, has significantly contributed to this advancement. Whether autonomous vehicles with all their advantages and disadvantages will become part of everyday life is still an open question.

6. REFERENCES

- [1] Bagloee, S.A., Tavana, M., Asadi, M. *Autonomous vehicles: challenges, opportunities, and future implications for transportation policies*, Springer, 2016
- [2] A. M. Adil and U. F. Aziz: *Sonar Based Obstacle Detection and Avoidance Algorithm*, in IEEE International Conference on Signal Acquisition and Processing, 2009
- [3] Fuzzy Sets. In: *First Course on Fuzzy Theory and Applications. Advances in Soft Computing*, Springer, 2005
- [4] Stanford Encyclopedia of Philosophy, *Fuzzy logic*, First published 2016
- [5] Enric Trillas, Luka Eciolazag , *Fuzzy logic*, Springer, First edition 2015
- [6] Ishaya Emmanuel, *Fuzzy Logic-Based Control for Autonomous Vehicle*, Published by MECS Publisher., 2017
- [7] Dejan S. Misović, Intelligent system for traffic management based on the application of fuzzy logic, doctoral dissertation, Belgrade, 2019
- [8] Fagnant DJ, Kockelman K., *Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations*. Part A, 2015
- [9] Robotics Business Review. "Infographic an animated look at how self driving cars work." roboticsbusinessreview.com, URL: <https://www.roboticsbusinessreview.com/unmanned/unmanned-ground/infographic-an-animated-look-at-how-self-driving-cars-work/>, Publication Date: 28 November 2019, Last Accessed Date: 27 May 2023.
- [10] Adilova, N.E. (2020). *Consistency of Fuzzy If-Then Rules for Control System*. In: Aliev, R., Kacprzyk, J., Pedrycz, W., Jamshidi, M., Babanli, M., Sadikoglu, F. (eds) 10th International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions, Springer, 2020
- [11] MCV UAB. "M5 visual recognition Image." mcv.uab.cat, URL: <https://mcv.uab.cat/m5-visual-recognition/>, Last Accessed Date: 27 May 2023.
- [12] Bundesministerium für Verkehr und digitale Infrastruktur (Federal Ministry of Transport and Digital Infrastructure). "New vehicle safety systems." bmdv.bund.de, URL: <https://bmdv.bund.de/SharedDocs/EN/Articles/StV/Roadtraffic/new-vehicle-safety-systems.html>, Last Accessed Date: 27 May 2023.

Usage of Fuzzy Logic and Machine Learning in Web Development

Zlatko Čović

Subotica Tech – College of Applied Sciences, Subotica, Serbia

chole@vts.su.ac.rs

Abstract — This paper presents the possibility of using fuzzy logic and machine learning in web development with PHP programming language. Fuzzy logic and machine learning found their applications in various fields of web development. The Python programming language is one of the most popular for developing applications that utilize fuzzy logic and machine learning. When it comes to integrated web systems using some form of AI, it's common to see the AI module implemented using the Python programming language, while the rest of the system is written using other programming languages and technologies. API endpoints are created for data exchange among different languages within such systems. PHP, as a server-side programming language, is used on over 75% of websites, according to the W3Techs report. The work briefly describes the most popular libraries in the PHP programming language that are used for applying fuzzy logic and machine learning. Implementation requirements and basic features are listed.

Keywords: fuzzy logic, machine learning, web development, php

1 INTRODUCTION

Fuzzy logic is a mathematical framework designed to address uncertainty and imprecision in decision-making and reasoning. It enables the representation of levels of truth, as opposed to rigid true or false values. Fuzzy logic employs fuzzy sets, fuzzy rules, and fuzzy inference to manage imprecise or ambiguous information and arrive at approximate decisions.

It is applied in diverse fields such as control systems, decision support systems, expert systems, natural language processing, and machine learning. Fuzzy logic empowers systems to process uncertain and subjective information, offering greater flexibility and nuanced reasoning abilities.

Fuzzy logic can be employed to enable adaptability in web interface design. In [1], authors presented development of web portal to support online shopping, with such characteristic. The web portal stored past shopping records and maintained the actual shopping list. Portal used collected data and a fuzzy approach to create adapted functionality in web interface. According to the results, it can be said that usage of fuzzy methodology for adaptation into Web interfaces can feasibly be implemented.

In various loyalty programs, employers create mobile applications and/or websites that utilize some form of machine learning and fuzzy logic. Based on information obtained from consumer habits (what they buy, when they buy, where they buy, how much they buy, etc.), algorithms are triggered to predict a personalized offer for the consumer. Based on reactions to the generated offer, the algorithm learns and continues to improve.

Accurate effort estimation is crucial in software development, as it directly influences project timelines and success. The study [2] introduces a model for estimating effort in web-based mobile app development using an object-oriented approach. The proposed model combines functional size measurement with object-oriented web app metrics and mobile characteristics. Estimations are performed using the Mamdani fuzzy logic method.

Artificial intelligence (AI) pertains to the development and deployment of intelligent systems capable of executing tasks that usually necessitate human intelligence. It encompasses the emulation of human-like intelligence in machines to equip them with the capacity to perceive, reason, learn, and make decisions.

Machine learning is a component of artificial intelligence that concentrates on creating algorithms and models enabling computers to learn from data and make predictions or decisions without explicit programming. This entails scrutinizing extensive datasets to discern patterns, extract meaningful insights, and construct predictive models.

The popularity of the Python programming language arises from its utilization in developing applications incorporating fuzzy logic and machine learning. This popularity is attributed to its rapid execution speed, relatively uncomplicated programming syntax, and a multitude of pre-existing libraries equipped to execute various tasks linked to artificial intelligence.

In scenarios that involve integrated web systems using some form of artificial intelligence, it is common to utilize an AI module that is created using the Python programming language. The rest of the system is typically implemented using other programming languages and technologies.

Within such systems, API endpoints are generated to facilitate data exchange across diverse languages. Data exchange is conducted in one of the accepted standard formats, but the JSON format is the most popular due to its processing speed and the fact that all programming languages have good functions and libraries for working with this data format.

In these systems, sometimes ML modules are created as standalone components with the option to import and export data into a text file without direct communication with the rest of the system.

As per the W3Techs report [3], PHP, functioning as a server-side programming language, is used on more than 75% of websites. The paper briefly outlines the most prominent libraries within the PHP programming language, utilized for the application of fuzzy logic and machine learning.

1.1 Application of Fuzzy logic in web development

In the realm of web development, various applications of fuzzy logic can be observed:

Enhanced Search Functionality: Fuzzy logic contributes to refining search capabilities on websites by yielding more precise and pertinent outcomes. This involves accommodating approximate matches and managing partial correspondences, empowering users to locate desired information even in the absence of exact search terms.

Efficient User Input Handling: Fuzzy logic proves valuable in adeptly managing user input within web forms. Its capacity to decipher and comprehend ambiguous or vaguely articulated inputs mitigates errors and elevates the overall user experience.

Facilitating Decision-making: Fuzzy logic plays a pivotal role in aiding decision-making processes within websites. Its ability to embody uncertainty and imprecision within decision models is particularly beneficial for recommendation systems, personalized content delivery, and adaptive user interfaces.

1.2 Application of machine learning in web development

Personalization and Recommendation Systems: Machine learning analyzes user data for personalized website recommendations, often seen in e-commerce, streaming, and social media platforms. Decision Trees and Random Forests algorithms are versatile and can be used for both classification and regression tasks.

Natural Language Processing (NLP): ML, especially NLP, creates chatbots, virtual assistants, and voice interfaces for websites, enhancing user interaction.

Fraud Detection and Security: ML detects website fraud patterns (e.g., payment fraud, fake accounts) to enhance web app security.

Predictive Analytics: ML algorithms analyze website data for predictions and insights, aiding in user behavior forecasts, campaign optimization, and data-driven decisions. Linear Regression is an algorithm that can be suitable for these types of tasks.

Text classification tasks: Naive Bayes algorithm could be used for tasks like sentiment analysis in social media or customer support interactions.

Image classification (recognizing objects in images): Support Vector Machines (SVMs) excel in classification tasks, particularly in situations where there exists a distinct and well-defined separation between classes.

1.3 Application of AI in web development

Artificial intelligence (AI) is seamlessly integrated into the web development industry, handling increasingly intricate tasks. Most popular 7 ways AI is used in a web development are:

- Coding Assistants
- Artificial Design Intelligence (ADI)
- Smart Chatbots for a Personalized End-User Experience
- Voice-Based Interactions for a Convenient Search Experience

- Search Engine Optimization (SEO) Made Easier
- Dynamic Content Personalization
- Website and Vulnerability Testing [4]

1.3.1 Coding Assistants

Tools such as Codex and Github's Copilot aid programmers by proposing the subsequent lines of code they're actively engaged with, or even compiling small code segments for integration. Nevertheless, these tools are limited to managing concise and uncomplicated programs, and their precision rates are modest [4].

1.3.2 Artificial Design Intelligence (ADI)

Artificial Design Intelligence (ADI) can produce complete e-commerce websites tailored to users, covering a range of elements including visual design, layout, and content. Through a series of straightforward inquiries and drawing upon its extensive training data, ADI can create websites that match any objective and accommodate a variety of tastes. One of the examples is Wix ADI, technology that has the capability to create individual websites [4].

1.3.3 Smart Chatbots for a Personalized End-User Experience

An AI chatbot enables users to quickly access information through natural language queries, leveraging website content and customizable responses. Intercom offers AI-driven chatbots that seamlessly integrate into websites, apps, and other software solutions for companies [4].

1.3.4 Voice-Based Interactions for a Convenient Search Experience

Voice search offers a more efficient way to access information compared to manually browsing through written content. Users can verbally pose their question, and an AI system optimized for voice recognition retrieves and delivers the relevant information audibly. Apple's virtual assistant, Siri, can retrieve information and perform various tasks through voice commands [4].

1.3.5 Search Engine Optimization (SEO) Made Easier

An AI-driven SEO tool can manage these tasks on your behalf. It can generate effective headlines and topic suggestions for your writing, along with supplying the essential keywords to enhance your content's ranking.

SEO.ai is software for swift and effective search engine optimization content creation [4].

1.3.6 Dynamic Content Personalization

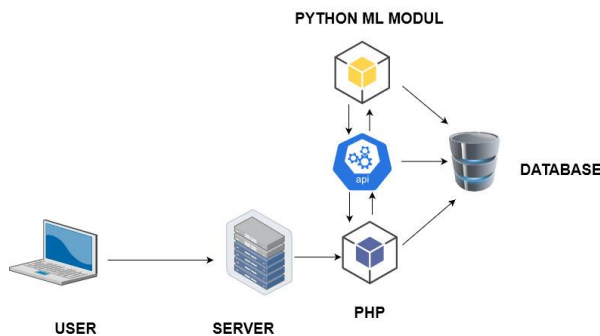
AI-powered dynamic content personalization enables custom experiences for large customer bases, leveraging diverse data sources to understand behaviors and craft tailored marketing strategies. Braze is a customer engagement platform that helps brands maintain audience engagement through customer-centric interactions [4].

1.3.7 Website and Vulnerability Testing

Conventional website code testing requires hands-on effort, with programmers closely overseeing outcomes, refining tests, and monitoring changes. AI can automate simpler tests, allowing programmers to focus on complex issues, streamlining testing and improving efficiency. Diffblue Cover uses AI to automate Java unit test generation, aiding programmers in Java application creation, updating, and maintenance [4].

2 WEB SYSTEM WITH ML MODULE

Integrated web systems often incorporate a Python-based ML/AI module, while other languages and technologies handle the rest of the system. API endpoints facilitate data exchange across languages, commonly using JSON due to its speed and universal support. In some cases, ML modules function independently, allowing data import and export via text files without direct system communication. The figure (1. figure) shows the most common architecture of integrated web system with ML module. In many systems, the AI/ML module is physically located on a separate server.



1. figure: Example of architecture of integrated web system with ML module

PHP is the most popular server-side programming language, and it is used on over 75% of websites. For using PHP instead of Python in ML modules, it is necessary to be familiar with existing classes and libraries.

3 PHP CLASSES AND LIBRARIES

PHP has several classes and libraries that are used for performing fuzzy logic and machine learning operations. One of the more popular built-in PHP functions is *levenshein()*, which calculates Levenshtein distance between two strings.

3.1 Levenshtein

The *Levenshtein* Distance (LD) is a fuzzy matching technique used to measure the difference between two strings. The resulting number indicates the degree of dissimilarity between the two strings. A higher *Levenshtein* edit distance signifies a greater disparity between the two terms.

```
levenshtein(
    string $string1,
    string $string2,
    int $insertion_cost = 1,
```

```
    int $replacement_cost = 1,
    int $deletion_cost = 1
): int
```

The Levenshtein distance is defined as the minimal number of characters you must replace, insert or delete to transform string1 into string2 [8].

If insertion_cost, replacement_cost and/or deletion_cost are unequal to 1, the algorithm adapts to choose the cheapest transforms. E.g. if \$insertion_cost + \$deletion_cost < \$replacement_cost, no replacements will be done, but rather inserts and deletions instead [8].

As an illustration, when measuring the distance between "Cristian" and "Christian," the distance is 1, as they differ by only one letter, "h." This term is commonly used interchangeably with "edit distance".

3.2 FuzzyDM library

Fuzzy logic decision making library is one of the PHP's libraries created for the Fuzzy logic. Library can be installed with composer via *Packagist* web site (<https://packagist.org/packages/ketili/fuzzydm>). It requires 5.3.0 or above version of PHP.

FuzzyDM is a small PHP library to make multi criteria decisions. It relies on fuzzy-logic fundamentals and employs simple membership and aggregate functions. With this library, decisions can be made based on some predefined numeric features.

3.3 PHP AI libraries

Integrating AI into PHP programming is achievable through various libraries and frameworks. Notable ones within the PHP ecosystem, like PHP-ML and Rubix ML, enable this incorporation. They provide a wide range of AI algorithms, including neural networks, decision trees, and clustering, allowing for the creation of intelligent applications.

Additionally, PHP frameworks like Laravel and Symfony come with specialized tools and extensions that facilitate the use of AI libraries and seamless integration of AI capabilities into web applications.

3.4 PHP-ML

PHP-ML is a PHP library dedicated to machine learning. It offers a diverse array of algorithms and tools for tasks like classification, regression, clustering, dimensionality reduction, and beyond. The design prioritizes user-friendliness and seamless integration with existing PHP applications. PHP-ML is released under the MIT Licence. It requires PHP version 7.1 or above. Complete documentation of usage of this library and sample examples for every feature can be found at <https://php-ml.readthedocs.io/en/latest/>.

Here are some advantages of PHP-ML [5]:

- Seamless Integration: PHP-ML effortlessly integrates with various PHP frameworks and libraries.
- User-Friendly Interface: The library boasts a user-friendly interface, facilitating an easy entry point for developers venturing into machine learning.

- **Thriving Community:** PHP-ML benefits from a robust developer community, actively contributing to its advancement and offering support.
- **Diverse Algorithmic Support:** PHP-ML accommodates an extensive array of machine learning algorithms, encompassing classification, regression, clustering, and more.
- **Swift Execution:** PHP-ML is engineered for efficiency, ensuring rapid execution of machine learning algorithms.
- **Open-Source:** PHP-ML is open source, affording developers the freedom to utilize and customize it according to their specific requirements.

PHP-ML offers a lot of features. Here are some of them [5]:

- **Modular Approach:** PHP-ML presents a collection of distinct components that stand alone effectively or meld seamlessly to construct a comprehensive machine-learning system.
- **Easy Usage:** PHP-ML boasts a user-friendly interface, ensuring accessibility for both newcomers and seasoned machine learning experts.
- **Extensive Algorithmic Diversity:** PHP-ML encompasses a vast spectrum of machine learning algorithms, including classification, regression, dimensionality reduction and clustering.
- **Open-Source Advantage:** PHP-ML operates as an open-source solution, granting the freedom to utilize and adapt as per individual requirements.
- **Streamlined Data Preprocessing:** PHP-ML streamlines data preprocessing capabilities, encompassing functions like normalization, standardization, and feature scaling.
- **Robust Cross-Validation:** PHP-ML facilitates robust cross-validation capabilities, a pivotal tool for appraising the efficacy of machine learning models [5].
- **Integrated Visualization:** PHP-ML seamlessly integrates built-in visualization features catering to data and machine learning models.
- **Versatility Across Formats:** PHP-ML offers support for a multitude of data formats, encompassing CSV, TXT, and JSON, simplifying compatibility with diverse data sources.

Several general procedures need to be involved in utilizing PHP-ML. First it is necessary to install the library using Composer, which will add the library to your project dependencies.

As it works with data, it needs to be loaded into an array or a dataset object. The data can be read from a database, a file, or any other source.

After loading, preprocess data as needed for machine learning, including normalization, outlier removal, and converting categorical variables to numerical. Data division is necessary in training and testing sets. The training set instructs the model, and the testing set assesses the effectiveness of the model.

The next step is to choose a machine learning algorithm that is proper for a given problem. Model can be trained by

providing the training data and the selected algorithm to the relevant class in PHP-ML.

The testing data can be passed to the trained model for the assessment of its performance. Once your model has been trained and tested, it can be employed for predictions on new data. The trained model can be saved to a file for future reuse. Deployment to a web application or any other software system where it can be utilized for predictions is also possible. Below a simple example of classification is shown [5].

```
require_once __DIR__ . '/vendor/autoload.php';

use Phpml\Classification\KNearestNeighbors;

$samples = [[1, 3], [1, 4], [2, 4], [3, 1],
[4, 1], [4, 2]];
$labels = ['a', 'a', 'a', 'b', 'b', 'b'];

$classifier = new KNearestNeighbors();
$classifier->train($samples, $labels);

$classifier->predict([3, 2]);
// return 'b'
```

Example scripts are available in a separate GitHub repository <https://github.com/php-ai/php-ml-examples>. One of the scripts is *Code Review Estimator*, which is a simple showcase of machine learning for code review cost estimation.

Many of the examples have been used in articles written by the author of this library, for example: Prediction of air pollution with k-Nearest Neighbors. The k-Nearest Neighbors algorithm is a straightforward method for classification (supervised learning). Its operational principle involves identifying the k closest neighbors to the target value, usually determined using a distance metric [5].

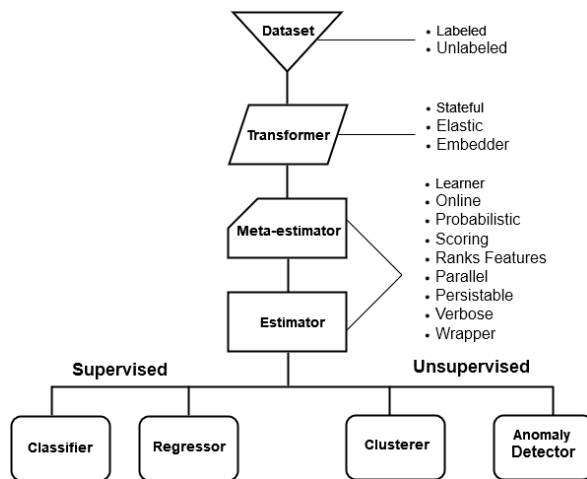
3.5 Rubix ML

Rubix ML is a free open-source machine learning library that allows you to build programs that learn from your data using the PHP language. It is the most popular ML library for PHP. It provides tools for the entire machine learning life cycle from ETL to training, cross-validation, and production with over 40 supervised and unsupervised learning algorithms. Rubix ML strives to enhance the accessibility of machine learning for PHP developers, offering a user-friendly platform to construct, train, and assess machine learning models [6].

It can be installed using *Composer*. Library requires PHP version 7.4 or above. *Tensor extension* for fast Matrix/Vector computing is recommended. It is a library and extension that provides objects for scientific computing in PHP. All Rubix ML projects are constructed to run from the PHP command line interface (CLI). The reason almost always boils down to performance and memory consumption [6].

Optional extensions for working with the Rubix ML library can also be installed: GD extension for image support, Mbstring extension for fast multibyte string manipulation, SVM extension for Support Vector Machine engine (libsvm), PDO extension for relational database support and GraphViz for graph visualization [6].

A flexible modular architecture characterized by several fundamental abstractions and their corresponding types and interfaces is employed by Rubix ML [6]. The figure (2. figure) shows the Rubix ML system architecture.



2. figure: Rubix ML system architecture

Some benefits of using Rubix ML include [5]:

- **User-Friendly Design:** Rubix ML is crafted for ease of use, ensuring a straightforward learning curve. Its intuitive and uniform interface empowers developers to effortlessly build machine-learning models.
- **Versatility:** Rubix ML accommodates an extensive array of machine learning algorithms, granting developers the freedom to select the one that aligns most closely with their requirements. Additionally, it permits users to tailor their models and algorithms to suit specific preferences.
- **Efficiency:** Rubix ML is engineered for high performance and scalability, enabling it to process sizable datasets and intricate models with optimal effectiveness.
- **Data and Model Visualization:** Rubix ML offers resources for visually representing data and models. This aids developers in gaining a deeper comprehension of their data and enhancing their models.
- **Open source:** Rubix ML is freely available for use, and developers can actively contribute to its ongoing development.

In addition to the mentioned advantages, Rubix ML has features that make it a good choice for use in various projects. Some of them are:

- Accommodates a diverse array of machine learning algorithms.
- Furnishes a user-friendly and uniform interface.
- Enables training on various devices (e.g., CPU, GPU).
- Includes tools for feature selection and data preprocessing.
- Facilitates model persistence and serialization.

- Allows for hyperparameter tuning and model selection.
- Provides support for ensemble learning and pipelines.
- Comes with extensive documentation and a substantial collection of examples.
- Permits the development of custom models.
- Applicable in both research and production settings [5][6].

Example projects can be found in Tutorial page [7]. Many of them come with usage instructions and a pre-cleaned dataset. The most numerous examples are related to prediction, recognition, and analysis.

For example projects to be more easily understood, it is recommended to master the basic steps in creating project structure and applying this library. Additionally, you should have a fundamental understanding of various types of machine learning such as classification and regression [7].

Below, the steps in the basic example will be presented.

3.5.1 Obtaining data

Projects of machine learning usually begins with a question. Define the questions and collect the answers for them. For examples, data should be structured and represented in a 2-d array (or matrix), and if labels are used, they should be presented as a 1-d array.

```

$samples = [
    [3, 4, 40.2],
    [2, 2, 17.5],
    [2, 3, 14.7],
    [3, 4, 52.0],
];

$labels = ['graduated', 'ungraduated',
'graduated', 'ungraduated'];
    
```

3.5.2 Construction of Labeled or Unlabeled dataset

In Rubix ML, datasets are conveyed using specialized containers known as Dataset objects. Dataset objects manage the selection, subsampling, splitting, randomization, and sorting of both samples and labels they encompass. There are two categories of datasets: Labeled and Unlabeled.

```

use Rubix\ML\Datasets\Labeled;
$dataset = new Labeled($samples, $labels);
    
```

3.5.3 Estimator selection

The core of the Rubix ML library is composed of estimators. They furnish the *predict()* API and are tasked with generating predictions for unfamiliar samples. Estimators that can be trained with data are referred to as Learners and necessitate training prior to making predictions. Example shows the usage of K Nearest Neighbors classifier.

```

use Rubix\ML\Classifiers\KNearestNeighbors;
$estimator = new KNearestNeighbors(3);
    
```

3.5.4 Training

The process of feeding the learning algorithm data so that it can build an internal representation (or model) of the task it is trying to learn is known as training. This representation is comprised of all the parameters (except hyper-parameters) that are needed to make a prediction.

To initiate training, provide the training dataset as an argument to the *train()* method on the learner instance.

```
$estimator->train($dataset);
```

3.5.5 Making Predictions

The *predict()* method can be used with unlabeled dataset on newly trained estimator. The process of generating predictions is referred to as inference, as the estimator utilizes the model created during training to deduce the label of unknown samples.

```
use Rubix\ML\Datasets\Unlabeled;
```

```
$samples = [
    [3, 4, 40.2],
    [2, 2, 17.5],
    [2, 3, 14.7],
    [3, 4, 52.0],
];
```

```
$dataset = new Unlabeled($samples);
$predictions = $estimator->predict($dataset);
```

The estimator produces the predicted class labels for the unknown samples.

3.5.6 Model Evaluation

By setting some data aside, the model can be tested on data it has never encountered before. This technique, known as cross validation, aims to evaluate an estimator's capacity to generalize from its training.

```
use Rubix\ML\CrossValidation\HoldOut;
$validator = new HoldOut(0.2);
```

The *HoldOut* validator allocates a portion of the dataset for testing and reserves the remainder for training. The user needs to specify the ratio of testing to training samples as a constructor parameter for the *Hold Out* validator. For example, factor of 0.2 (20%) of the dataset for testing is chosen.

```
use Rubix\ML\CrossValidation\HoldOut;
$validator = new HoldOut(0.2);
```

For the *test()* method on the validator, it is necessary to select a compatible validation Metric to serve as the scoring function. In this example, Accuracy metric is used.

```
use Rubix\ML\Datasets\Labeled;
use Rubix\ML\CrossValidation\Metrics\Accuracy;
```

```
$dataset = new Labeled($samples, $labels);
```

```
$score = $validator->test($estimator, $dataset,
    new Accuracy());
```

4 CONCLUSIONS

Fuzzy logic and machine learning are potent technologies applicable across diverse fields, including web development.

The web development industry is already fully integrated with artificial intelligence, which is now handling increasingly complex tasks.

Python is highly favored for developing applications with fuzzy logic and machine learning. In integrated web systems employing AI, the AI module is typically written in Python, while the rest of the system uses different languages and technologies. API endpoints facilitate data exchange between languages in these systems. In the web environment, PHP can be a valuable alternative to the Python programming language, boasting strong support for AI through popular libraries.

The paper presented PHP classes and libraries for executing fuzzy logic and machine learning operations. Description of features and short examples of usage are explained.

REFERENCES

- [1] H. O. Nyongesa, Tian Shicheng, S. Maleki-Dizaji, S. . -T. Huang and J. Siddiqi, "Adaptive Web interface design using fuzzy logic," Proceedings IEEE/WIC International Conference on Web Intelligence (WI 2003), Halifax, NS, Canada, 2003, pp. 671-674, doi: 10.1109/WI.2003.1241293.
- [2] Agusta, S., Suharjo, Girsang, A.S. (2018). Effort Estimation Development Model for Web-based Mobile Application Using Fuzzy Logic, *Telkonnika*, vol. 16, no.5, pp.2082-2090, doi: 10.12928/TELKOMNIKA.v16i5.6561.
- [3] Web Technology Surveys, <https://w3techs.com>
- [4] Springboard, Monica J. W. (2023). 7 ways AI Is Being Used in Web Development, <https://www.springboard.com/blog/data-science/ai-web-development/>
- [5] Haider, H. (2023). Artificial Intelligence in PHP programming, <https://www.linkedin.com/pulse/artificial-intelligence-php-programming-haris-haider>
- [6] Rubix official page, <https://rubixml.com/>
- [7] Rubix ML example projects, <https://docs.rubixml.com/2.0/index.html#tutorials-example-projects>
- [8] PHP Levenshtein function, <https://www.php.net/manual/en/function levenshtein.php>

Soft computing methods in audit

Lágy számítás alkalmazása ellenőrzés területén

Péczka Polett

Óbudai Egyetem

Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar

Budapest, Hungary

peczka.polli@gmail.com

Abstract — The use of soft computing methods is very versatile. It plays a very important role in artificial intelligence, but it can also be used in much simpler ways. My aim was to show the so-called everyday side of this knowledge in engineering, so I will demonstrate the application of neural networks through a verification example.

Keywords: soft computing, neural network, audit

Összefoglalás — Lágy számítási módszerek felhasználása nagyon sokoldalú. A mesterséges intelligenciában nagyon jelentős szerepet tölt be, de ennél sokkal egyszerűbben is fel lehet használni. Céлом volt, hogy a műszaki életben úgymond mindennapi oldalát is bemutassam ennek az ismeretnek, így egy ellenőrzési példán keresztül mutatom be az alkalmazását a neurális hálózatoknak.

Kulcsszavak: Lágy számítás, neurális hálózat, ellenőrzés

1 BEVEZETÉS

A műszaki élet kistestvére szerintem az ellenőrzés. Mint oly sok területen, a műszaki életben is nagyon fontos a minőség megőrzése. A minőség szabvány szerinti definíciója: „egy tevékenység, folyamat, termék, rendszer vagy személy, vagy ezek bármilyen kombinációja, azon jellemzőinek összessége, amely befolyásolja a képességét, hogy meghatározott és elvárt igényeket teljesítsen.” [1] Ennél szerintem sokkal kézzelfoghatóbb ez a kifejezés. A minőség az előre meghatározott követelményekkel szembeni megfelelés, a felhasználói igény kielégítésével. Azonban nem szabad elfelejteni azt sem, hogy a minőség nagyon szubjektív kifejezés, nem lehet konkrétan meghatározni, hogy egy termék mikor „minőségi”, hiszen függ a felhasználók igényeitől, amely mind más és más. [2] A nagy üzemek kevesebb időt és költséget szeretnek erre fordítani, azonban a vásárlók megtartása érdekében elengedhetetlen egyfajta kompromisszumot kötni. Mindenki hallhatott már az ár-érték arányról, azonban a minőségnek semmi köze nincsen az árhoz, mégis én azt gondolom, hogy egy ilyen vizsgálat során nem szabad megfeledkezni arról sem. Ezért érdemes bevezetni a megfelelés fogalmát, hiszen ez objektíven mérhető. Egy termék vagy megfelelő, vagy nem, nincsen köztes részek. A legkisebbtől kezdve a legnagyobbig sorolhatnám a vállalkozásokat, de egy biztos, mind a profit miatt üzemel. Ennek az eljárásnak, amelyet a következőkben felvázolok,

egy finomított, továbbdolgozott formája alkalmas lenne megkeresni azokat a kritikus pontokat az üzemekben, gyárakban, vállalkozásokban, ahol a minőség romlik. A példamunkámban egy 4 állomásos üzemben, 210 napot, körülbelül fél évet fogok vizsgálni. A feladat végén ki lehet értékelni, hogy melyik állomáson mekkora a hatékonyság, milyen gyakran követnek el hibát. Céloom, hogy egy életszerű, de nem túl bonyolult feladaton keresztül bemutassam a neurális hálózat működését. Sajnos ez a rendszer még nem teljesen automatizált, hiszen maguk az ellenőrzési feladatok még humán erőforrást igényelnek, azonban távolabbi céljaim között tartom számon, hogy magát az ellenőrzési feladatokat is automatizálni lehessen.

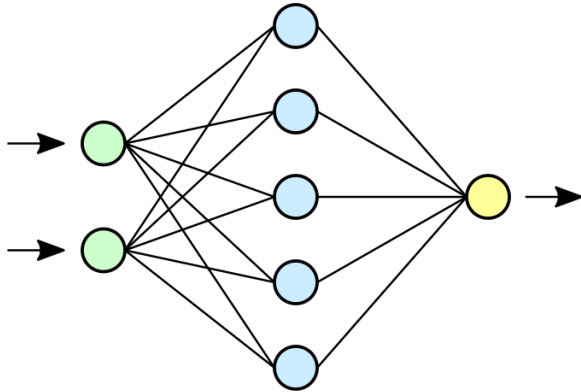
2 LÁGY SZÁMÍTÁSI MÓDSZEREK

A mesterséges intelligencia kutatásában jelentős terület a lágy számítási modell. Alapgondolata a biológiai ismeretek felhasználásán alapuló adatfeldolgozás. A hagyományos számítástechnikával ellentétben közelítő modellekkel dolgozik, tolerálja a pontatlanságot, bizonytalanságot és a részleges igazságot. Elmélete az 1980-as évekre tehető, de mára már számos helyen sikeresen alkalmazzák. [3][4] A számítástechnika fejlődése is segíti a gyors előmenetelét, hiszen az egyre nagyobb teljesítményű processzorok kedvező áron megszerezhetőek, nagyon apró mikrochipek is egyre nagyobb számban megtalálhatók a piacon. Az egyre összetettebbé váló rendszereket nagyon nehéz hagyományos vezérlőrendszerekkel irányítani. A lágy számítási módszer kombinálja a Fuzzy logikát, neurális hálózatot és a genetikus algoritmust, így ezek nem versenyeznek, hanem kiegészítik egymást. Kijelenthető, hogy a komplex problémák megoldására célszerű alkalmazni ezen megoldási lehetőséget. [5]

2.1 Neurális hálózat

A fuzzy logikát már több évtizede alkalmazzák komplex szabályozási feladatokra. Ehhez képest a neurális hálók még gyerekcipőben járnak. Ezt a biológiai idegrendszer, az agy inspirálta. A hálózat elemei a neuronok, amelyek össze vannak kötve egymással. Célja, hogy a gépi „gondolkodást” közelítse az emberihez. A neurális hálózatot úgy kell betanítani, hogy a bemenetek ismert halmaza a kívánt kimenetet eredményezze. A rendszernek előre meghatározott tanulási szabályt kell „megtanítani”, ami alapján el tudja végezni a súlyozást is. Előnye, hogy képes kezelni hiányos adathalmazokat is, visszacsatolásoknak köszönhetően minimálisra

csökkenthető a kimeneti jel hibája. Az eredmény attól függ, mennyire pontosak a rendelkezésre álló adatok. Hátránya, hogy esetenként túlzott mértékű képzésre lehet szükség. A neurális hálózatok több rétegből épülnek fel: bemeneti, rejtett és kimeneti. [5]



1. ábra: Neurális hálózatok [6]

2.2 Rekurrens neurális hálózatok

A rekurrens neurális hálózatok memóriája emlékszik, az előző lépésekben kapott eredményekre. Az össze bemenet ugyanazt a feladatot hajtja végre. A független rétegeket azonban átalakítja függő rétegekké. Ekkor az összes rétegre ugyanazt a súlyt adjuk meg, ezzel csökkentve a visszatérő paraméterek és rétegek számát. Ezáltal a neurális hálózat könnyebben memorizálja, tanulja meg az előző kimenetet, hogy a kimenet a következő bemenet lesz. Az összes rejtett réteg összekapcsolható egyetlen ismétlődő rétegbe, hogy a súlyok az összes rétegnél azonosak legyenek. [7]

A rekurrens neurális hálózatoknál fontos a sorrend, annak megfelelően következnek a új bemenetek. Miután lezajlott a belső folyamat a kimeneti jel lesz az új bemeneti jel. A belső folyamatokban játszódhatnak le az előre betáplált szabályok, így vagy a soron következő lesz az új bemenet, vagy ugyanaz, ami volt.

3 FELADAT RÉSZLETEZÉSE

A gépészet területén nagyon gyakran szembe találok magam azzal, hogy nagyon sok plusz költség keletkezik egy-egy tervezési hibából, a rosszul megírt ajánlatból, a gyártáskor, szereléskor pontatlanul, hanyagul elvégzett feladatokból. Sokszor az előttünk rosszul elvégzett munkából adódóan hibázunk, azonban mégis a mi munkánk lesz rosszul elvégezve, ezzel plusz munkát kényszerülünk megcsinálni. Lavinaként működve, egymás hibáiból következnek be a hibák. Egy hibás felmérés magával vonzza a hibás tervezést, ami miatt a gyártásban is bekövetkezhetnek hibák, amelyek majd csak a szereléskor fognak szembe tűnni, például amikor a híddaru beleüt az acélszerkezetbe. A precíz munka nagyon fontos a műszaki területen is, ezért sokszor érdemes lenne több ellenőrzést végrehajtani a munkaállomások között. Egy jó visszaigazolás lehetne arra is, hogy milyen arányokban osszák szét a bónuszt, ami akár motiváló erővel hathatna, hogy jobban működjön a vállalkozás.

Az általam vett példában 4 munkaállomást fogok megvizsgálni, minden nap más állomást. A folyamatot az ajánlatkészítés, felmérés fázissal kezdem el, majd a soron következő a tervezés lesz, utána a gyártás és végezetül a szerelést fogom górcső alá venni. Ha megfelelő a minőség,

a jól működik az a sor, akkor azt egy mátrixban eggyessel jelölöm, azonban, ha hibát találok és nem megfelelőnek minősítem, akkor nulla értéket kap. Ezek segítségével létrehozok egy 210 elemből álló mátrixot, amely egyesekből és nullákból áll. Ez bárhogy alakítható, azonban ez fog segíteni abban, hogy el tudjuk dönteni, hogy melyik lépés lesz a következő. Mivel a feladatot rekurrens neurális háló segítségével fogom megoldani, ezért fontos, hogy legyen egy sorrendi mátrix is, amely segítségével fogjuk tudni, hogy melyik munkaállomás a soros. Itt fontos, hogy a második elemmel kell kezdeni a sort, akkor fogjuk kapni az első lépésben a helyes eredményt, tehát ha megfelelő volt, akkor a tervezést, ha nem volt megfelelő, akkor újra az ajánlat, felmérés állomást.

Természetesen ezt manuálisan, kézzel is meg lehet oldani, azonban a lágy számítási módszerek legfőbb területe a mesterséges intelligencia fejlesztése, így én csak az első pár lépést készítettem el kézi megoldással, hogy szemléletesen be tudjam mutatni, hogy milyen elemekből milyen algoritmus alapján számol a program. A kézi számolás mellett nagyon időigényes, sok hibázási lehetőséget rejt, hiszen egy monoton feladatról beszélhetünk. A MATLAB programot választottam ki, hogy megoldjam a feladatot, hiszen ez a platform remek lehetőségeket nyújt a mátrixműveletek megoldására.

3.1 Mátrixok meghatározása

Mint már fentebb említettem a különböző elemeket mátrixokba rendeztem. Eddig a megfelelőség meghatározására szolgáló naptárról és a sorrendiséget szemléltető mátrixról volt szó. A naptár maga a vizsgálati időszak. A sorrendiségért felelő mátrix (1) első oszlopa a tervezés, második a szerelés, harmadik a gyártás, a negyedik pedig az ajánlat, felmérés.

$$L = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (1)$$

Ezek mellett szerepel a Kezdeti mátrix, amiben a különböző munkaállomásoknak a „kódját” tanítottam meg a programnak. Mivel 4 különböző állomást kell felprogramozni, így ezek a mátrixok 1 soros és 4 oszloposak. Itt is csak egyeseket és nullákat használtam, az egyesek mindig egy oszloppal el vannak csúsztatva. (2, 3, 4, 5)

$$A = [1 \ 0 \ 0 \ 0] \quad (2)$$

$$T = [0 \ 1 \ 0 \ 0] \quad (3)$$

$$G = [0 \ 0 \ 1 \ 0] \quad (4)$$

$$S = [0 \ 0 \ 0 \ 1] \quad (5)$$

Fő mátrixok soraiba tartoznak a hibamátrixok, amik felelősek azért, hogy a kimeneten melyik értéknek kell majd szerepelnie. Ezeknek már csak két elemük van, ami azért következik be, mert itt két dolgot szeretnénk megfigyelni, megfelelő: nincs hiba (7) vagy nem megfelelő: hibás (6) az adott állomás munkája.

$$H = [1 \ 0] \quad (6)$$

$$N = [0 \ 1] \quad (7)$$

Ezekon kívül, még úgynevezett segédmátrixok fogják előre juttatni a program működését. Az egyik egy egyszerű egység mátrix (8), valamint általam elnevezett igaz (9) és hamis (10) mátrix.

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

$$I = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix} \quad (9)$$

$$F = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \quad (10)$$

A mátrixok méretéből is lehet következtetni, hogy mely mátrix melyik másikkal fog az első körben „együttműködni”, hiszen vannak 4 oszlopú és 2 oszlopú mátrixaink. A kiinduláskor a bemenő mátrixok a kezdeti állapot, ebben az esetben az ajánlat készítés valamint a hibamátrix annak megfelelően, hogy volt hiba, vagy sem. A belső rész két részre osztható Először is a bemeneti állapot, egység mátrix és a sorrendi mátrixból keletkezik egy új mátrix, aminek az eredménye egy 1 oszlopú, 8 sorú mátrix, ahol az első 4 elem az aktuális állapot, a következő négy pedig a következő állapot. (11)

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad (11)$$

Másik része pedig a minőséget, hibát fogja feldolgozni. Mivel a későbbiekben a belső rész összegezve lesz, ezáltal annak is hasonló mátrixnak kell lennie. Itt a nem megfelelő állapothoz tartozó hiba mátrixot és az igaz/hamis mátrixot fogjuk felhasználni. (12)

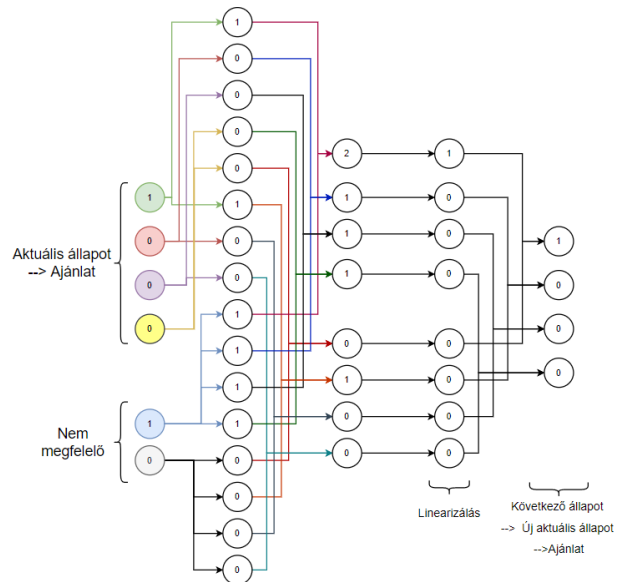
$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ \underline{1} & \underline{0} & \underline{1} \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (12)$$

Ezután összegezve a belső rész elemeit, megkapjuk a megoldást. Itt még el kell végezni egy egyszerű linearizálás, hiszen egy helyen az egytől és nullától különböző értéket kapunk. Tehát a maximum érték helyére egyet kell írni, minden más érték benne nulla lesz. Végezetül az egyszerűség jegyében az első négy értéket össze kell adni a mátrix második négy értékével. Itt már összetéveszthetetlenül megkapjuk, hogy mely állomás következnek. Az érthetőség jegyében a következő két ábrán (13):

$$\begin{bmatrix} 1 & 1 & 2 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 3 & 1 \\ 0 & 1 & 2 & 0 \\ 0 & 1 & 2 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (13)$$

3.2 Manuálisan elvégzett RNN

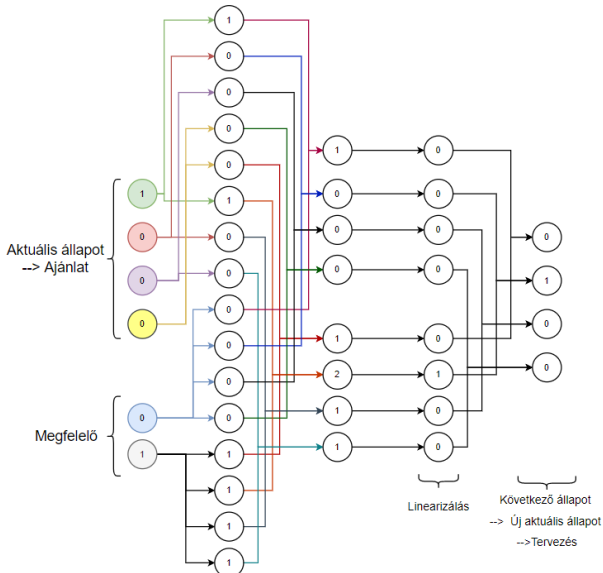
Manuálisan elvégezve is ezt a gondolatmenetet követtem. Ezáltal az első két lépést elvégeztem a MATLAB-ban elkészített program nélkül.



2. ábra: A második állomás megállapítása neurális hálózattal

Már a 2. ábrán is jól megfigyelhető a neurális hálózatoknak a felépítése. Látszik, hogy a kimenet lesz az új bemenet, azzal kell elvégezni a következő napi rutin ellenőrzést. Mivel a második bemenet az mondja, hogy nem megfelelő, ezért a következő bemenetet meg kell

ismételni. Ebből következik, hogy szintén az ajánlatadás ellenőrzése kerül sorra.



3. ábra: A harmadik állomás megállapítása neurális hálózattal

A 2. ábrától eltérően a 3. ábrán a bemenő jelek az ajánlat, hiszen ezt a lépést meg kellett ismételni, és a megfelelő hibamátrix. Az előzőekhez hasonlóan megismételtem a lépéseket. Mivel a hiba jelet megváltoztattam, hogy megfelelt a munkaállomás működése, ezért a kimenet „ugrott”, a tervezés lett, így a harmadik lépésben már a bemeneti jel a tervezés. A MATLAB programom 210 napra kiszámította, hogy mely napokon milyen ellenőrzések voltak. A programozási felületen szintén ezeket a műveleteket végeztem el. Egy for ciklusba számolva másodpercek alatt elvégzi a műveleteket. Ehhez csak a kezdeti értékeket, mátrixokat kell betáplálni, valamint azt, hogy az aktuális napon megfelelő vagy nem megfelelő a működés. A megírt programál már csak a naptár értékeket kell változtatni, és már működik, a többi adat állandó.

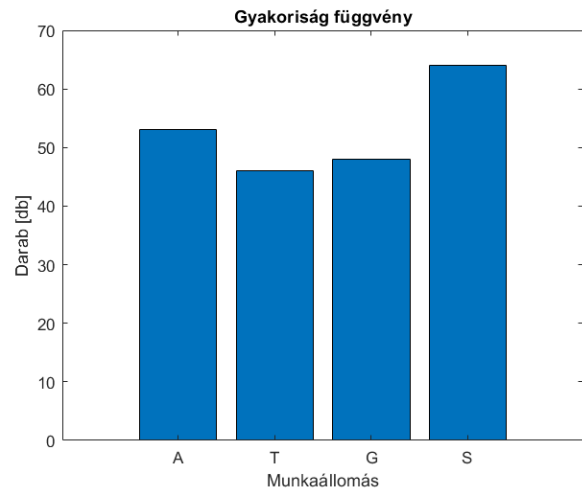
3.3 MATLAB program használata

A feladathoz különböző kiegészítő funkciókat nem használtam, csak a mátrix műveleteket végeztem el, amelyre tökéletesen alkalmas. Mindenképpen a program használatának előnye, hogy hatalmas adathalmaz esetén is nagyon gyors választ kapunk, kevesebb a hibázási lehetőség, egyszer jól megírt program már magától végzi a feladatot, hiszen a szükséges „tananyagot” megtanítottuk neki. Nem mellesleg a mestersége intelligencia ugródeszkája, ezért is fontos, hogy bár nagyon szép ábrák készülhetnek kézi elkészítési módszerrel, szükséges a gépi megoldás. Egyéb előnyei közé sorolható, hogy nem csak magát a hálózatot lehet leprogramozni, de ezeket az adatokat fel lehet könnyedén használni, például diagramok elkészítésére, gyakoriságot számítani, vagy összegezni, hogy mennyi olyan alkalom volt, amikor ténylegesen történt hiba. Az adatokat ki lehet exportálni excel formátumba is, így szélesebb körben fel lehet.

A program első felét lefuttatva csupán megkapjuk azt az adatsort, amely megmutatja, hogy mikor melyik munkaállomás volt ellenőrizve. Tehát a neurális hálózatot lehet lefuttatni. Ezzel az adatsorral tovább lehet dolgozni, egyéb fontos adatokat is ki lehet szűrni belőle.

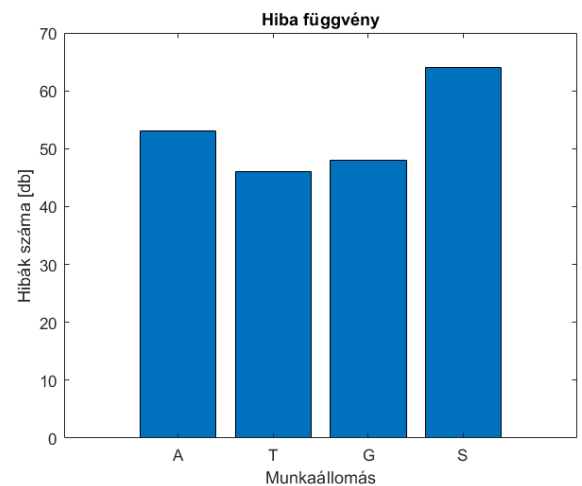
4 KIÉRTÉKELÉS

Az lehet tudni, hogyha 210 napon keresztül zajlanak az ellenőrzések, akkor ideális esetben ajánlat, felmérés és tervezés állomásokra 53 alkalommal esik az ellenőrzés, míg a gyártásra és szerelésre csupán 52 alkalommal. Az ellenőrzéseket egymás után, 3 alkalommal végeztem el, hogy azt is ki lehessen értékelni, hogy mekkora hatással voltak az ellenőrzések a munkaállomások működésére.



4. ábra: Gyakoriság függvény

A 4. ábrán a gyakoriság függvény látható. Ebből látható, hogy a bizonyos állomásokon mennyiszor zajlottak ellenőrzések. Már itt is megfigyelhető, hogy biztosan voltak hibák. Az ajánlat állomáson 45, tervezéskor 48, gyártásnál 48, míg a szerelésnél 64 alkalommal fordultak meg az ellenőrző szerv tagjai. Azonban sajnos ez még nem elégséges információ, hiszen alacsony számok alapja lehet az is, hogy hogy valamelyik állomáson kimagaslóan sok hiba volt, ritkán feleltek meg az elvárásoknak. Természetesen szűrhetünk le belőle adatot, de az csak sejtésként kezelhetjük. Ami ténylegesen megadja az információt, az az, hogy a neurális hálózatban mennyi olyan alkalom volt, amikor a bemenet megegyezett a kimenettel. Ezt le kellett szűkíteni arra, hogy ez külön kiértékelhető legyen az állomásokra is, ezzel megfigyelve, mennyire működnek hatásosan. Ezt hibafüggvénynek neveztem el és az 5. ábrán szemléltetem a kapott eredményeket.

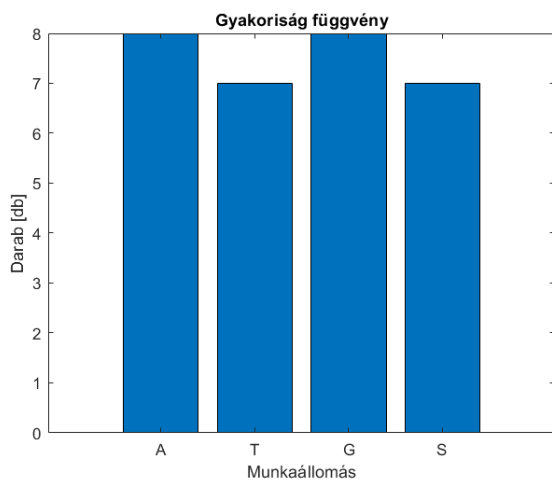


5. ábra: Hiba függvény

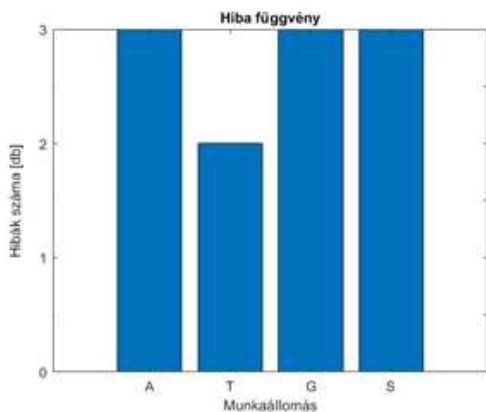
A sejtés bebizonyosodni látszik, hiszen a legtöbb hiba, 38, a szerelési soron történt. Második helyen a nem megfeleléségi listán az ajánlatkészítés áll, 26-szor kellett megismételni az ellenőrzést, a tervezést egy hajszállal előzte meg a gyártás, a kettő között összesen 3 ellenőrzés volt. Gyártás 22, a tervezés 19 újra ellenőrzéssel végzett ebben a vizsgálati időszakban.

Levonható ebből az a következtetés, hogy összesen 105-ször kellett megismételni valamelyik munkaterepen a folyamatot, tehát 105 alkalommal egyezett meg a bemeneti a kimeneti jellel. A vállalkozás ez által 50%-os megelégedettséggel üzemel. Ha szeretnénk növelni a megelégedettséget, akkor legfőképpen az ajánlat készítést, felmérést és a szerelést kell javítani. A hibáknak a 36%-a szerelésből, közel 25%-a pedig az ajánlatokból származik.

Szeretném megvizsgálni, hogy a következő időszakban, harminc napban sikerült-e fejlődést elérni és javítani a megelégedettséget.



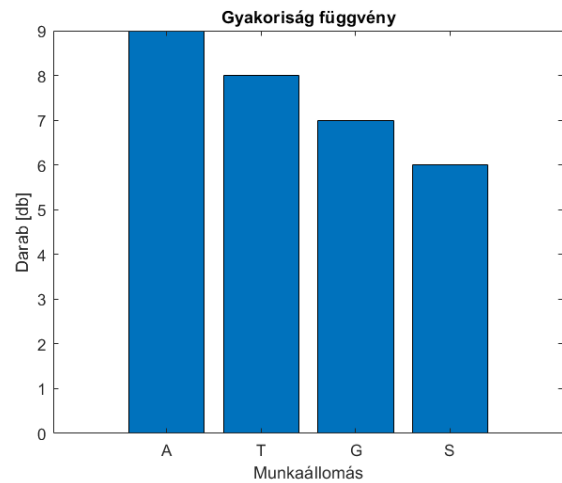
6. ábra: Gyakoriság függvény



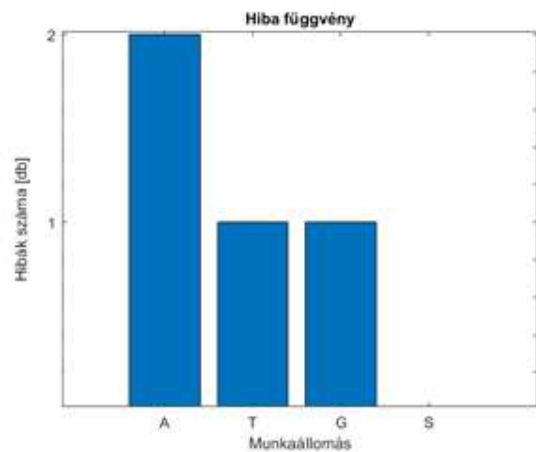
7. ábra: Hiba függvény

Az idő hiányában a különbség az előző ellenőrzéshez képest, hogy csak 30 napot vizsgáltam, valamint a mérésnél is ugyan az volt a vizsgálati tartomány. A 6. ábrán megfigyelhető, hogy voltak hibák, hiszen az első két munkaállomáson 8, míg a maradék állomásokon 7 ellenőrzésnek kellett volna hibátlan működés esetén végbe mennie. A 7. ábra többet mond, hiszen megfigyelhető, hogy a három olyan állomás van, ahol háromszor zajlott le ismételt ellenőrzés, tervezési fázisban pedig csak kétszer. Ez azt jelenti, hogy összesen 11 „hibás nap” volt, így

kijelenthető, hogy ezekben a napokban javult az üzem megfelelése, 36%-os a hibarátá.



8. ábra: Gyakoriság függvény



9. ábra: Hiba függvény

E vizsgálati időpontban is sokat javult a vállalkozás, egyre nagyobb a megfelelés. Itt már körülbelül 15%-os a hibarátá, ami már kifejezetten jónak mondható.

Ez a csökkenés betudható a jutalmak növekedésének, de akár a rövidebb ellenőrzési időszaknak is betudható. Sajnos az idő szűke miatt csökkentenem kellett a vizsgálati időt, azonban szerintem így is látványos, a téma szempontjából fontos, neurális hálózatok gyakorlati alkalmazása az ellenőrzés területén.

5 ÖSSZEZÉS

Célom az volt, hogy az üzemhez tartozó minőségbeli gondok javuljanak, ezzel növelve a megelégedettséget is. A minőség romlása vezethet a profit csökkenéséhez, így az üzemekben nagyon fontos a minőséget ellenőrizni. Különböző eljárásokkal meghatározhatók a gyenge pontok, amiket célszerű vizsgálni. A program fejleszthető lenne még a feladat különböző súlyzó értékekkel, például mennyi plusz költség van egy-egy rossz ajánlat elkészítésekor, vagy mekkora plusz költséggel jár egy nem megfelelő, sok selejtet termelő gyártás. Ezzel a fejlesztett eljárással pontosabb értéket kaphatunk, ezzel párhuzamban növelve a profitot is. Nagy előny lenne, ha az ellenőrzési feladatokban is nagyobb szerepet kapjon az automatizálás. Az automatizálható részeit elkészítettem, azonban szeretnék

a továbbiakban még foglalkozni ezzel a témával, hiszen nagyon sok rejtett potenciál van még benne. Véleményem szerint a neurális hálózatok alkalmazásának területét tovább lehet bővíteni az ellenőrzési feladatok vizsgálatával.

Végezetül szeretném megjegyezni, hogy a neurális hálózatoknak a felhasználási módja nagyon változatos. Nem véletlen ez a mesterséges intelligencia alapja, hiszen az emberi gondolkodáshoz olya hasonlóan működik. Napról napra több lehetőség van, amelyre felhasználható, akár olyan hétköznapi dolgokra, amelyet ebben a cikkben levezettem.

6 KÖSZÖNETNYILVÁNÍTÁS

Ez a cikk az Óbudai Egyetem Bánki Donát Gépész és Biztonságtechnikai Mérnöki Kar és a Magyar Fuzzy Társaság támogatásával készült. Szeretném megköszönni Dr. Lukács Judit Tanárnő segítségét és a téma vezetését.

7 HIVATKOZÁSOK

- [1] MSZ EN ISO 8402: Minőségmenedzsment és minőségbiztosítás
- [2] Pietila, G., Lim, T. C. (2012). Intelligent systems approaches to product sound quality evaluations–A review. *Applied Acoustics*, 73(10), 987-1002. DOI: 10.1016/j.apacoust.2012.04.012
- [3] Kóczy T. László, Tikk Domonkos, Botzheim János: *Intelligens rendszerek*, Széchenyi István Egyetem, Győr, 2017
- [4] Fazekas István: *Neurális hálózatok*, Debreceni Egyetem, Debrecen, 2013
- [5] Ibrahim, D. (2016). An overview of soft computing. *Procedia Computer Science*, 102, 34-38.
- [6] https://upload.wikimedia.org/wikipedia/commons/3/3d/Neural_net_work.svg
- [7] <https://hu.photo-555.com/4082704-recurrent-neural-networks-rnn>