FireGrid – Software for 2D Fire Spread Simulation Using the Game Method for Modelling

Deyan G. Mavrov*, Veselina Bureva

Prof. Dr. Asen Zlatarov University
1 Prof. Yakim Yakimov Blvd., Burgas 8010, Bulgaria
E-mails: dg@mavrov.eu, vbureva@btu.bg

*Corresponding author

Received: June 09, 2021 **Accepted: February 16, 2022**

Published: March 31, 2022

Abstract: *The paper presents FireGrid, an application software program for performing two-dimensional fire spread simulation using Atanassov's Game Method for Modelling (GMM). The software implements a model of fire spread with one or more starting points of ignition onto a planar grid of square cells that represent an idealized terrain of flammable areas of vegetation, and inflammable areas of rocks and water basins. The applications allows also locating a fire's starting point(s) by subtracting the initial configuration from the final one and decrementing all affected and adjacent cells by one. In addition to the preliminary defining the pattern of fire spread, manual control of the spread is allowed during simulation by selecting the cells that are to burn on the next iteration.*

Keywords: *Game Method for Modelling, GUI, Programming.*

General information

The purpose of the presented software is to implement the Game Method for Modelling (GMM) [1], specifically for the purpose of simulation and analysis of forest fires. GMM, which was inspired by the cellular automata-based Conway's Game of Life, has already been applied successfully for different problems as solving conflicts in social systems [6], oil transformation in marine environment [7], stochastic processes [2], etc.

Standardly, GMM models a space as an infinite two-dimensional orthogonal grid of square cells, which initially are assigned different values with respect to the modelled space characteristics, and a set of rules is predefined to represent the specific behaviour of the modelled process in terms of incrementing, decrementing or maintaining the values of the cells across the subsequent process iterations. The complete algorithm for applying the GMM is described in details in [1].

In the herewith proposed application of wildfire propagation modelling, terrain data is represented as a finite grid of cells, where each cell can be active or inactive. Each cell is assigned either a numerical value between '0' and '9' (where the value '0' represents an inactive cell, i.e., an area that has been completely devastated by the fire) or a letter representing inflammable areas (rocky areas, water basins). Cell values from '1' to '9' represent the flammable vegetation in various degree of density, and such cells are considered active.

To initiate a fire spread simulation, it is necessary to select a cell from the grid which indicates the location of the ignition point. Then, in a stepwise manner, by performing a predetermined, user defined, fire spread pattern, the fire propagates in successive iterations following the rules:

1. $R \to R$
2. $S \to S$
3. $n \to n - 1$, for $n \in [1, 9]$.

Defining the fire spread is enabled by mouse input from the user, by opening a grid of cells with the point of ignition placed in the center and selecting one or more neighbouring cells. In this way, the pattern of fire spread allows the user to interpret in GMM terms some specific parameters of a real-life wildfire like wind direction and intensity, or slopeness of the terrain.

Alternatively, if we already have a grid containing an intermediate/final state of a wildfire, it is possible to find a candidate starting point by cell-wise subtraction of the initial grid configuration from the final one, and then decrementing all cells by one until only cells with a value of 1 remain. This process has previously been described in [3, 4].

The software is written in the C# programming language using the Visual Studio 2019 IDE and the .NET Framework. The terrain matrix is represented through a graphical grid which is drawn on screen using the facilities of the `System.Drawing` namespace [5].

Control and monitoring panel

The program provides a control and monitoring panel (Fig. 1). During execution the control panel keeps track of the following parameters:

1. Number of active simultaneous fires.
2. Consecutive number of the fire iteration.
3. Count of all flammable cells (with values between '1' and '9').
4. Count of lit cells, e.g., cells that will be affected by 'fire' at the next iteration by having their value decremented according to the rule.
5. Count of scorched (affected) cells, e.g., cells where fire is already effectively burning at the present step, i.e., whose initial integer value has been decremented at least once since the beginning of the simulation.
6. Count of unaffected cells.
7. Count of burnt cells, e.g., cells whose value has been decremented down to 0 and can be neither lit nor scorched any longer.

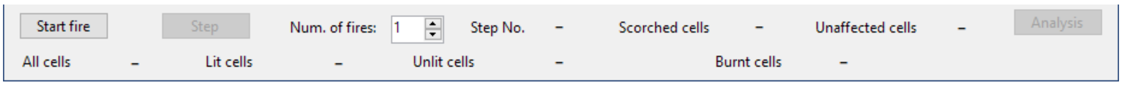
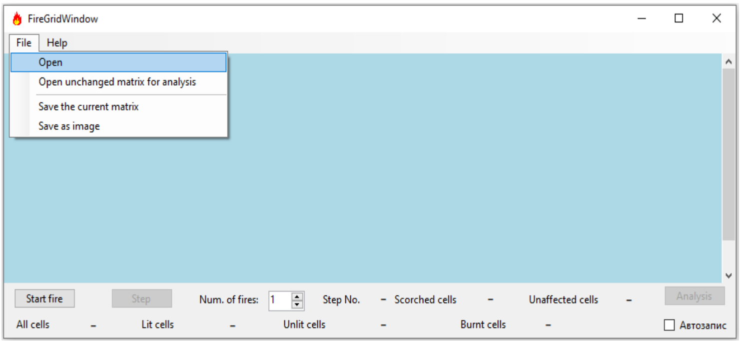


Fig. 1 Control and monitoring panel

GMM fire simulation

At launch the program's main window is empty. The user selects an input file with terrain data by selecting the option `Open` from the `File` menu (Fig. 2). The file must describe the cell grid as tabulations / newlines serving as cell / row delimiters. For example:

The program visualizes the input data as a grid of cells where inflammable areas like rivers and water basins are given in blue (R) and rocky areas are given in grey (S). The flammable cells with values from '9' to '1' are colored with gradually paler shades of green to represent the vegetation density, while cells valued '0' are colored white to signify the lack of flammable vegetation. If the terrain's shape is not rectangular, the remaining unused cells may be marked with the symbol “*”, which performs as the former two cell types but is visualized in black (Fig. 3).



9	5	7	5	3	5	4	6	8	2
8	6	7	6	5	4	5	7	7	2
7	5	7	5	3	S	S	8	7	2
6	4	6	5	4	S	S	7	8	2
4	5	8	7	5	4	5	6	6	2
5	3	4	3	2	7	9	8	7	2
2	4	5	4	3	9	R	R	R	R
5	3	4	5	6	9	R	R	R	4
8	9	7	6	8	R	R	7	7	4

Fig. 2 Initial state of the window

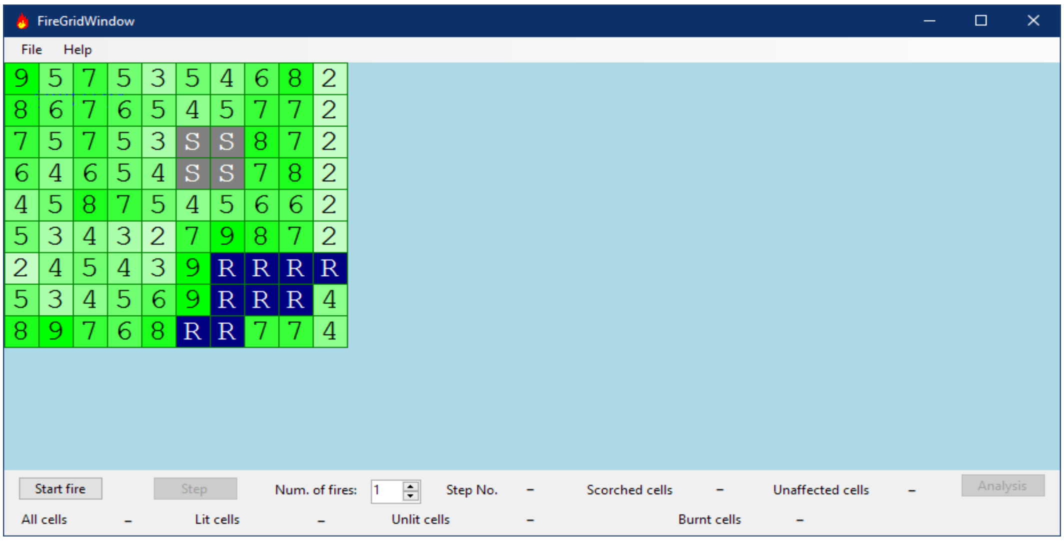


Fig. 3 Graphical visualization of the cell grid

The user initiates the fire spread simulation by pressing the Start fire button. The program requires obligatory input of the ignition point’s coordinates: to select these, a window is opened with Row and Column fields to be inserted (Fig. 4), where the top leftmost cell of the grid has coordinates ⟨0,0⟩.

In addition to the coordinates, the user must also define the fire spread pattern, i.e., determine by mouse’s input which of the neighbouring cells of a currently activated (lit) cell shall be further activated at the subsequent iteration. This spread pattern regulates the direction and speed with which the fire propagates, in a way to mimic real-life wind parameters, where the default pattern is as in the classical Conway’s Game of Life, i.e., only the four adjacent cells of a given cell are affected. If a cell contains vegetation, it gets lit and at the next step its value decreases by 1, otherwise it remains unchanged. If no cells are selected in the pattern grid, fire propagation would need to be manually controlled (see below).

After the fire is instantiated with a starting cell, the GMM procedure begins. The simulation steps are iterated by using the “Step” button (manually). The user may observe the changes in the grid, as shown in Figs. 5-8.

At first all numeric cells have their labels in black. Cells that have been lit, but have not yet had their value decremented, have their values colored in blue. At the next step, such cells get their values decremented by one. When the value of a cell gets decreased its numerical value gets colored in red. After a cell's value reaches '0' and may no longer get decremented the label with its value is set again to black colour.

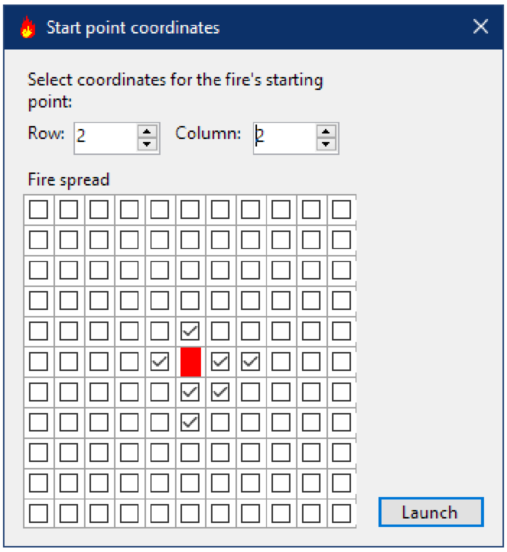


Fig. 4 Initial fire parameters

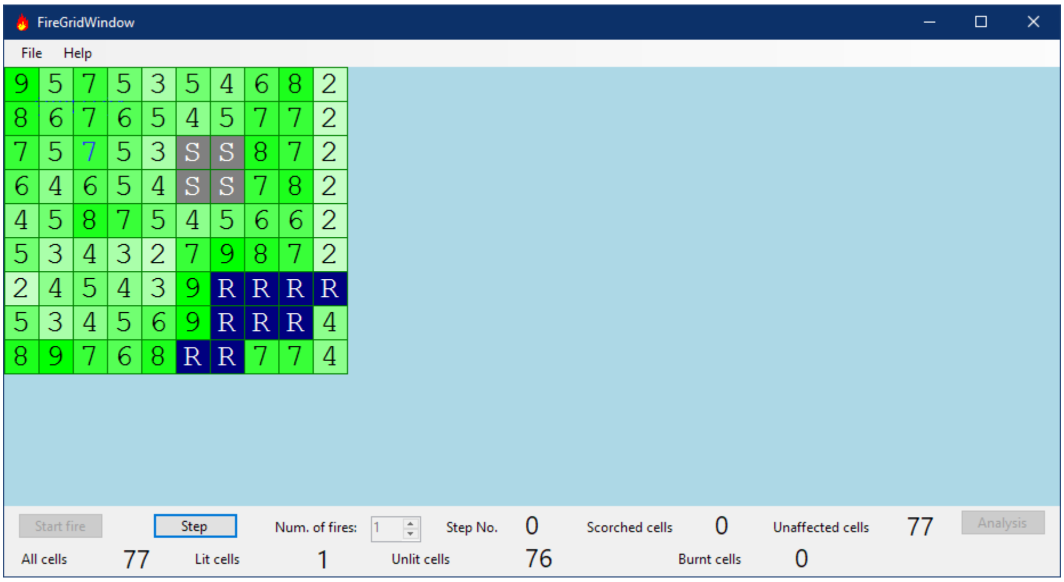


Fig. 5 Step 0 – immediately after starting the fire

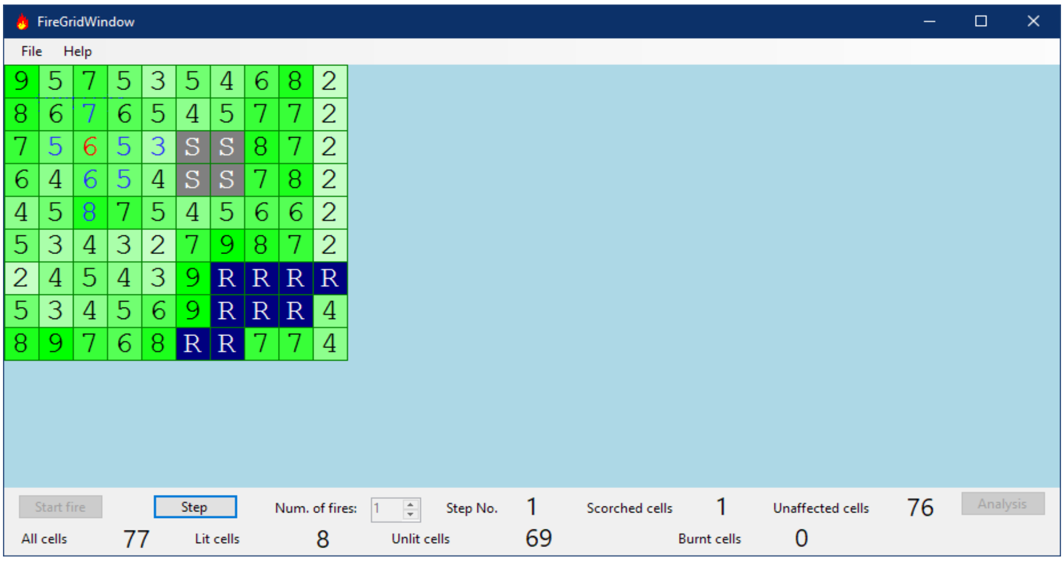


Fig. 6 Step 1 – the starting cell gets decremented by 1 and has its label coloured in red, and the cells that are to be subsequently affected are currently lit in blue

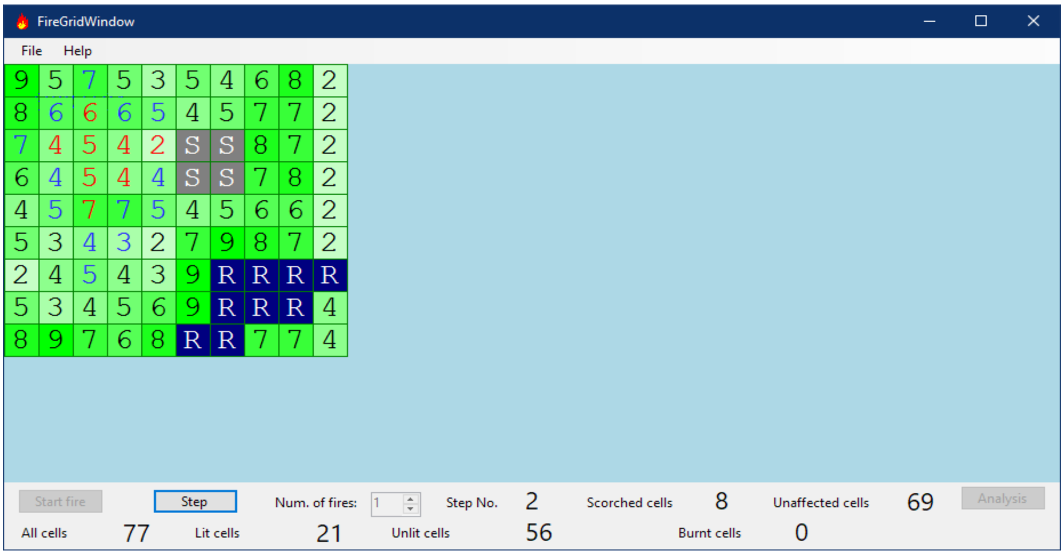


Fig. 7 Step 2 – after pressing the `Step` button twice, the fire starts propagating according to the selected pattern

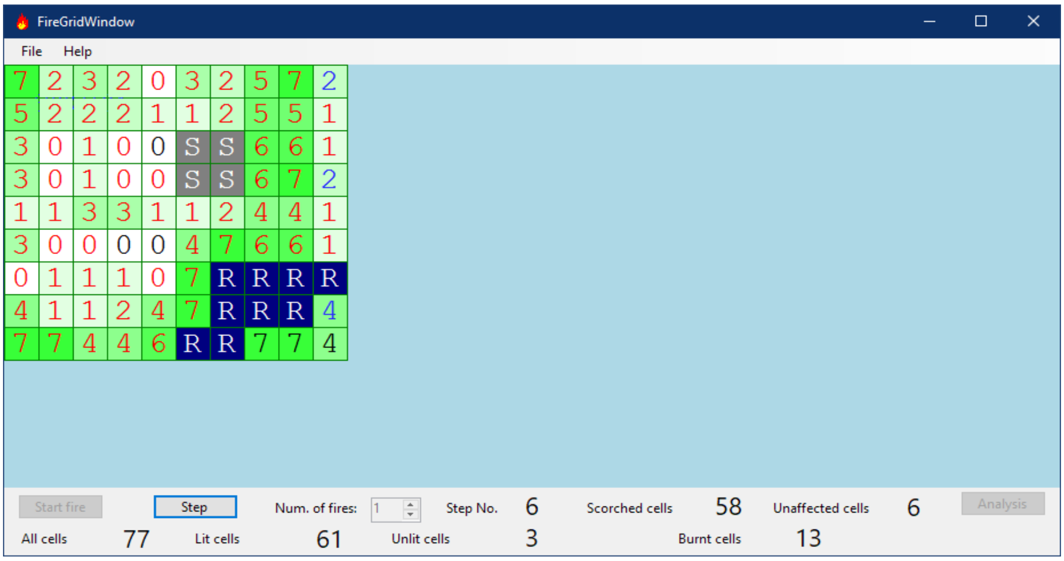


Fig. 8 Step 6 – the fire in the midway of burning. 13 cells have already burnt out.

Saving the grid

Over the course of the wildfire simulation, the current state of the grid can be saved in a text file delimited with tabulations and newlines. To save the grid in text format, one needs to open the File menu and select the option Save the current matrix. The resulting file is generated in the same file format as described above for input data (Fig. 9).

To save the grid as an image one needs to select the option Save as image, opening a window for the file name and destination directory; the grid is then rendered to a PNG image (Fig. 10).

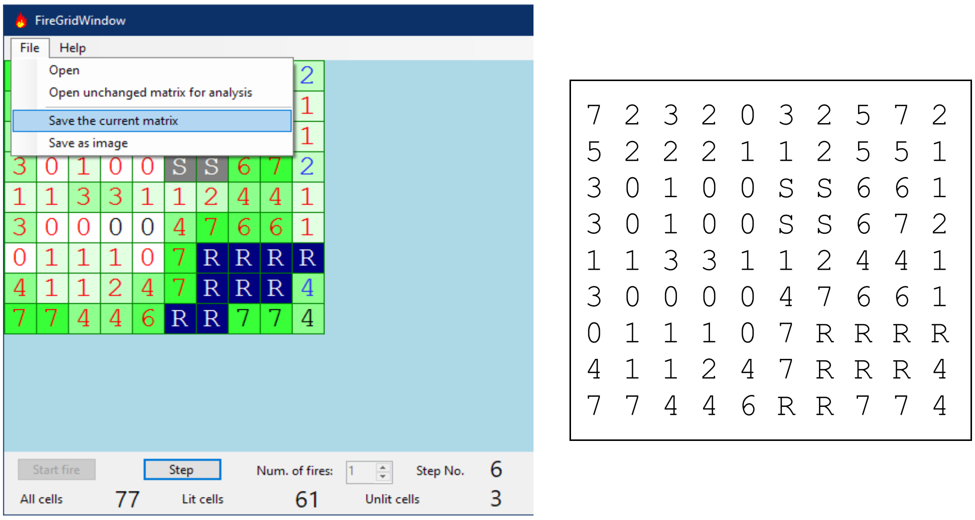


Fig. 9 Saving the grid to a text file

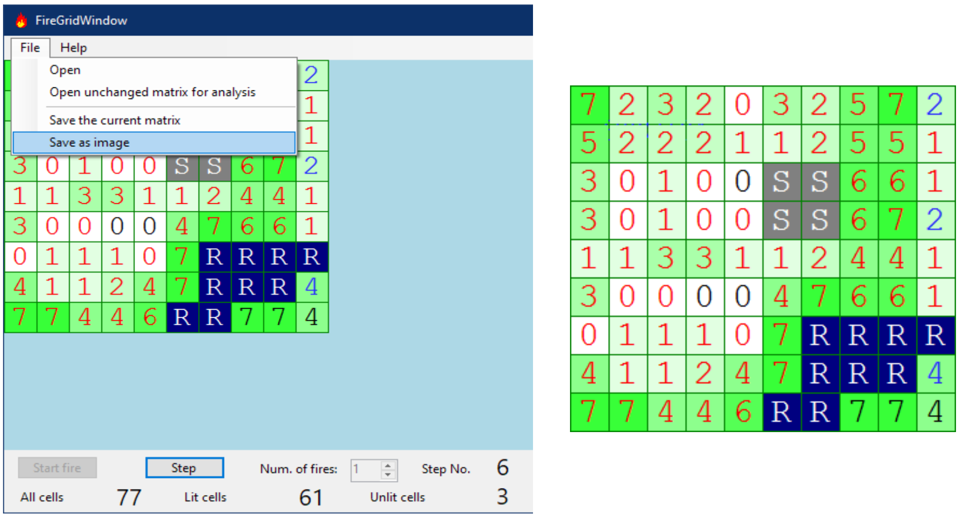


Fig. 10 Saving the grid to an image file

Initialization of multiple fires

The program allows the starting of two or more fires at the same time by selecting multiple start points. To do so, one is navigated by the system through the following steps (Fig. 11):

1. Enter the number of fires in the Num. of fires field.
2. For each fire, a separate window is opened prompting the user to input the coordinates of the ignition points and the fire spread patterns for each of the fires.
3. The Start fire button must be pressed.

After the fires have been activated they spread according to the GMM algorithm; whenever a cell is affected by more than one fire its rate of burning does not increase for this purpose, and its value is decremented by just 1 (Figs. 12-16).

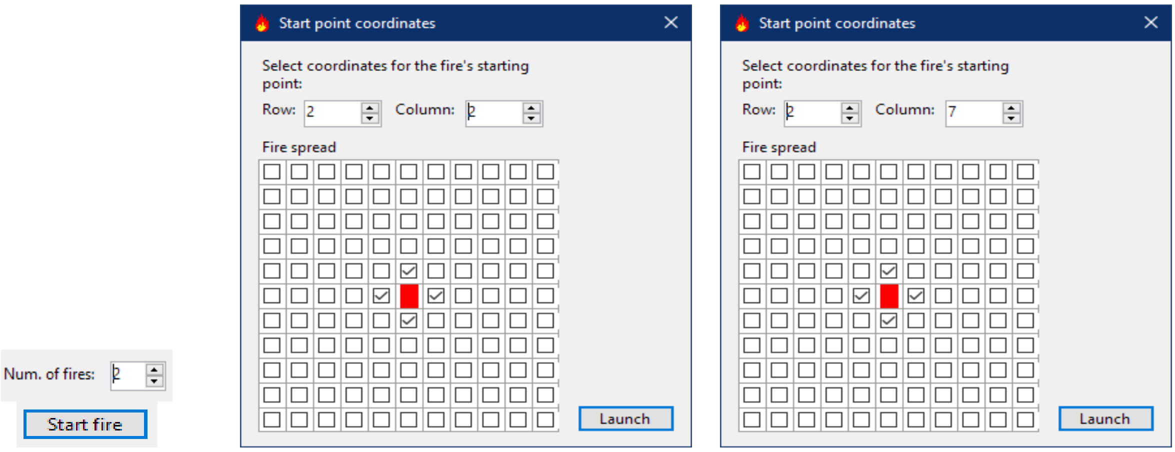


Fig. 11 Setting up two simultaneous fires
and configuring their spread patterns and starting points

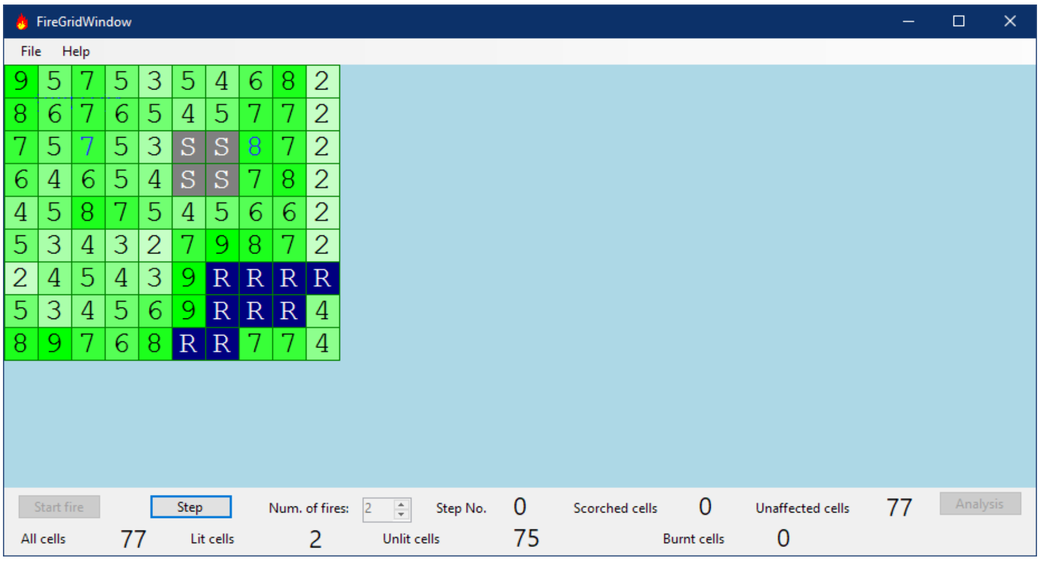


Fig. 12 Initial state after activating two fires

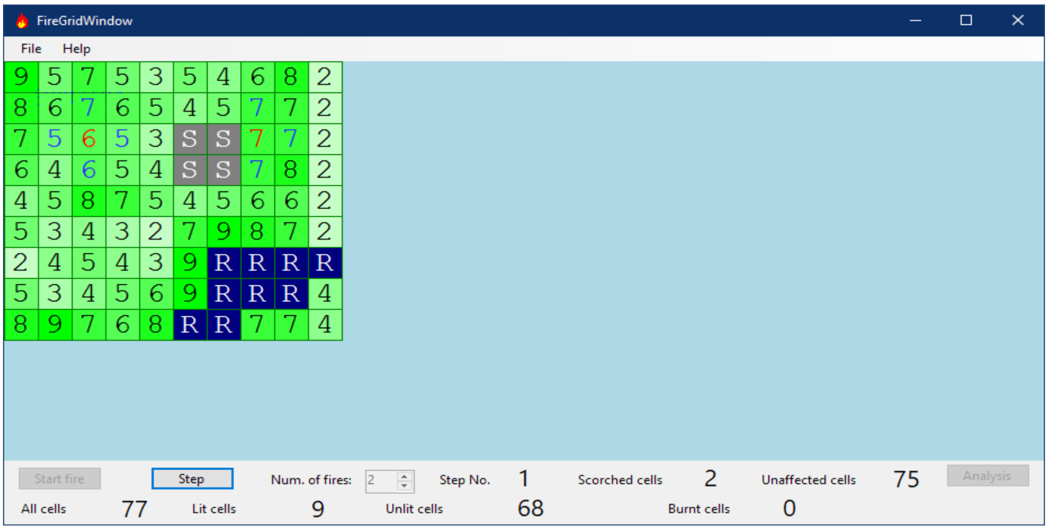


Fig. 13 Step 1

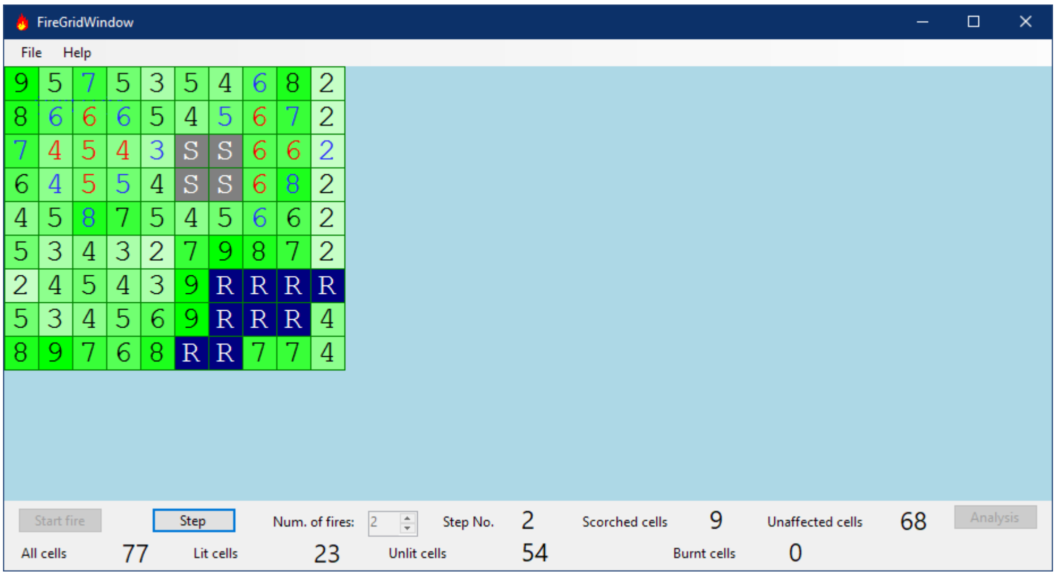


Fig. 14 Step 2

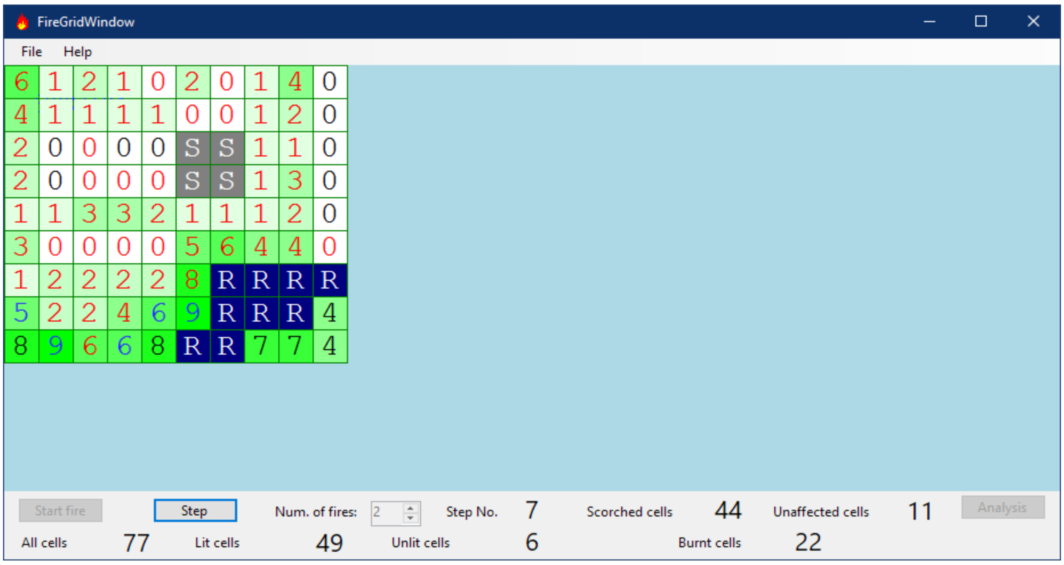


Fig. 15 Step 7

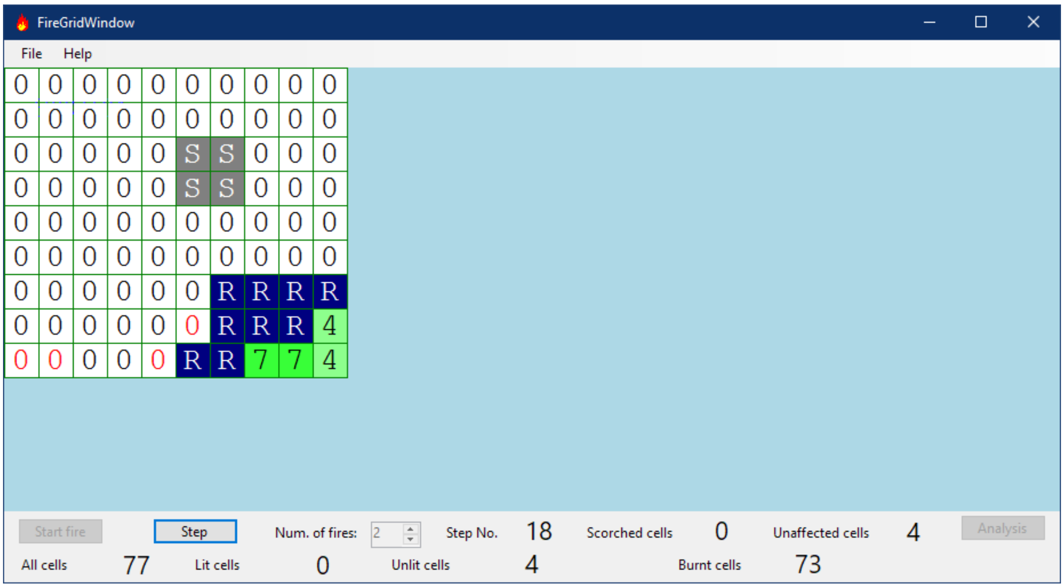


Fig. 16 End of fires – the chosen spread patterns do not allow reaching the last four cells with flammable vegetation

Manual fire control

When desired the user may lit grid cells manually by clicking on them with the left mouse button. They will start getting decremented at the next execution step. If an empty spread pattern is entered when the fire is initialized (Fig. 17), only the starting cell is automatically lit and the only other cells that will burn will be those that have been manually lit (Figs. 18-20).

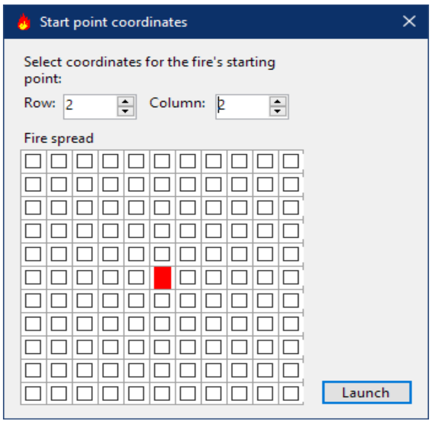


Fig. 17 Entering an empty spread pattern

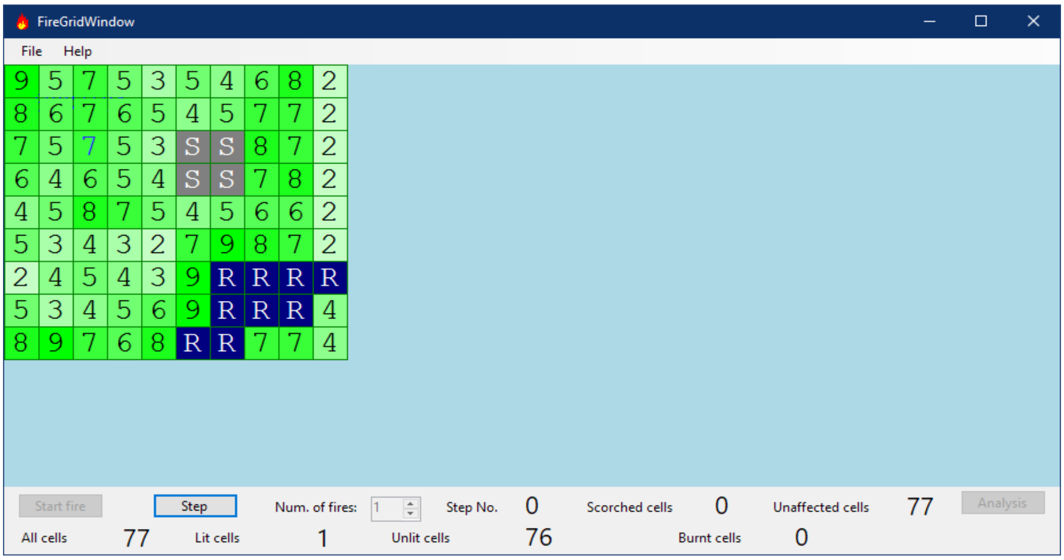


Fig. 18 Step 0 – only the start cell is lit

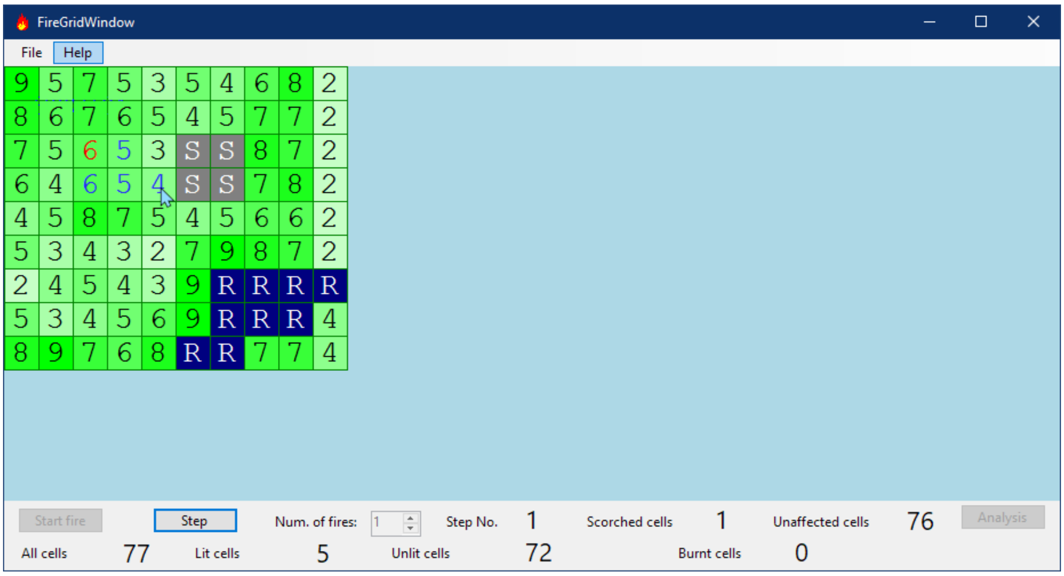


Fig. 19 Step 1 – the starting cell burns by 1 and we can light new cells with the mouse cursor

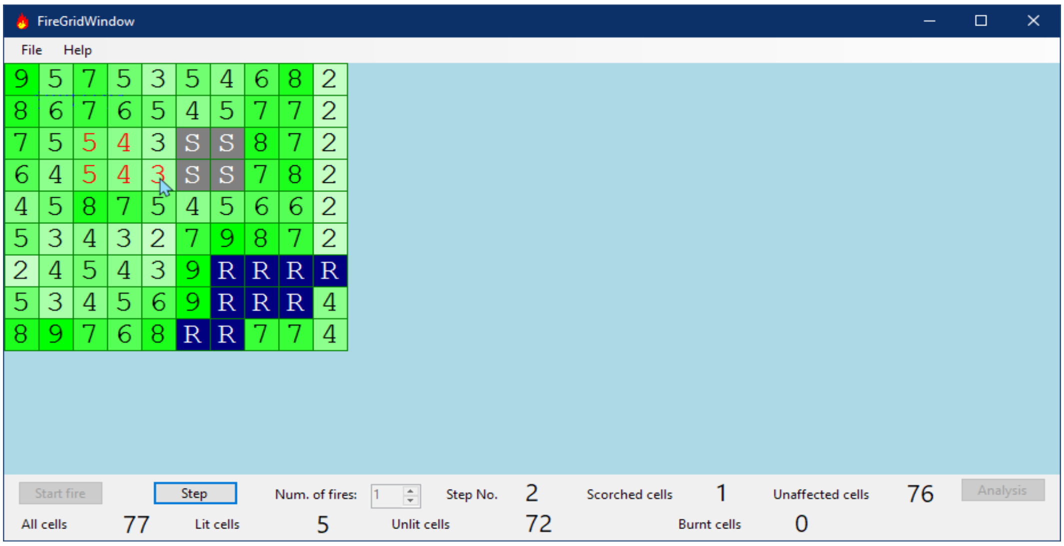


Fig. 20 Step 2 – all lit cells burn by 1

Locating a fire's ignition point

To make a prediction regarding the location of the cell containing the fire's ignition point, we will first need a grid which represents a fire at the midway of burning or one that has been extinguished. For that grid to be simulated in this direction, it is not allowed all of the cells to have been burnt to '0'. In the example below, we will use the grid we saved at Fig. 9. First, we need to open the text file in the usual manner. Then we must open the File menu and select the option Open unchanged matrix for analysis (Fig. 21). We need to select a matrix representing the terrain's state before the fire (Fig. 3).

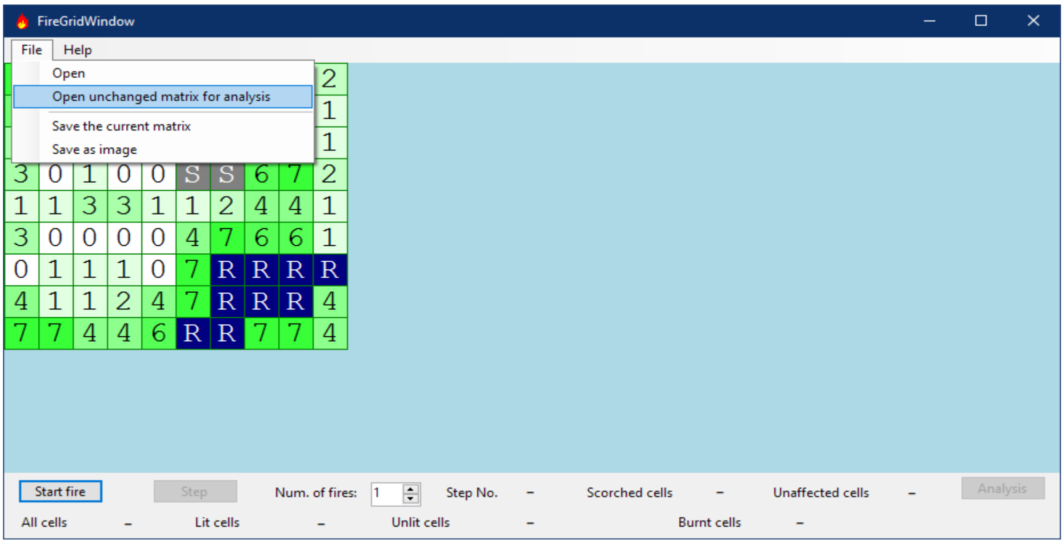


Fig. 21 Selecting an initial state matrix

Both grids need to have the same size. The program will then calculate the difference between the two grids (Fig. 22).

Finally, the user must press the Analysis button to begin the calculation process, where all numeric values get decremented by one at each step. The program will stop this process when the grid is left with cells containing only '0' and '1' (Fig. 23).

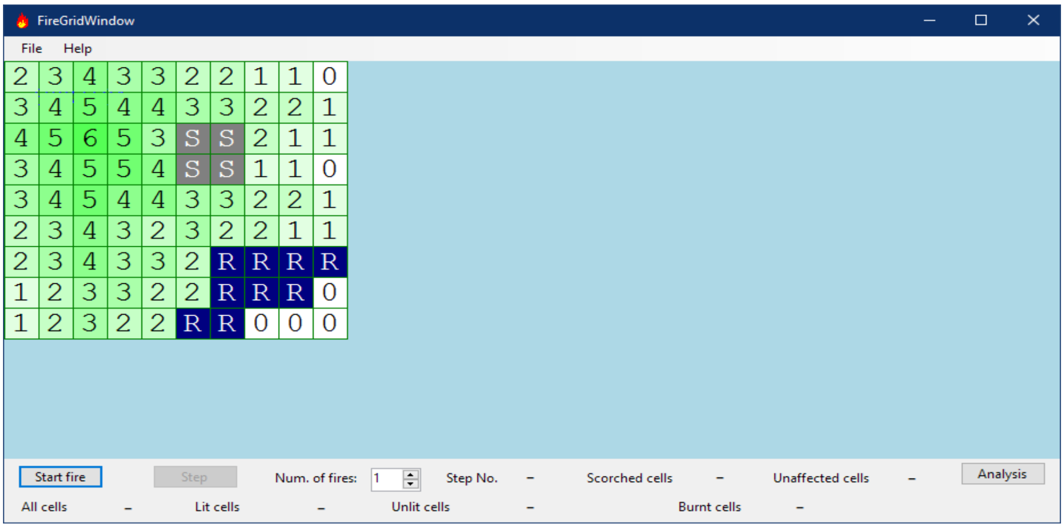


Fig. 22 Difference between the final and initial grid

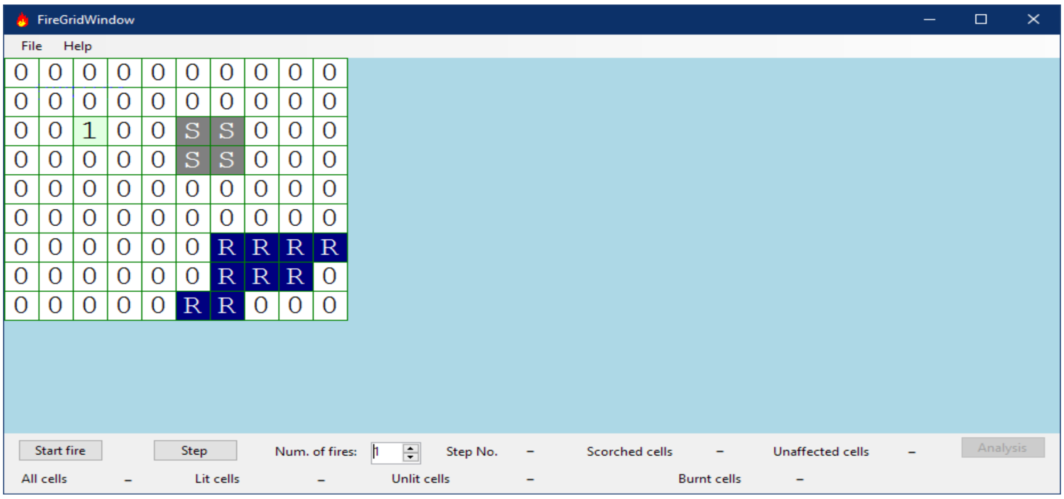


Fig. 23 The grid after the analysis. Here the '1'-s point to the plausible locations where the fire might have started.

Conclusion

The GMM forest fire spread simulation software *FireGrid* has been developed to allow automatic testing of a wildfire spread taking into consideration parameters like wind direction and intensity. This allows reducing the time needed to investigate fires using the Game Method for Modelling. While the software currently presents a rather idealized model that offers room for further improvement and elaboration, it already gives the user a number of features like simulation of complex fires with several ignition points, and prediction of the location of the potential ignition points on the basis of a current state of its propagation.

Wildfire propagation models and simulations are necessary for the analysis of real-life wildfire scenarios and for generating information about the potential fire behaviours. The software can be utilized by the firefighting departments as a training kit, for locating the appropriate places for establishing artificial firebreaks, or planning the human and technical resources necessary for managing actual wildfires.

Acknowledgement

The research is funded by the Bulgarian National Science Fund under Grant Ref. No. DN-16-6/2017 "Integrated Approach for Modelling of the Forest Fires Spread".

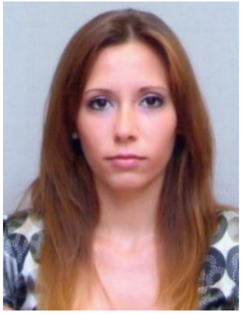
References

1. Atanassov K. (2011). Game Method for Modelling, Prof. M. Drinov Academic Publishing House, Sofia.
2. Atanassov K., E. Sotirova (2013). On Some Applications of Game Method for Modeling, Part 4: Interpretations of Some Stochastic Processes, Proc Jangjeon Math Soc, 16(3), 301-310.
3. Bureva V., E. Sotirova, P. Vassilev, V. Atanassova, O. Roeva, K. Atanassov, H. Tsakov (2022). Application of Game Method for Modelling to Locate a Forest Fire Ignition Point in the Presence of Wind, Uncertainty and Imprecision in Decision Making and Decision Support: New Advances, Challenges, and Perspectives, Lecture Notes in Networks and Systems, 338, 280-293, https://doi.org/10.1007/978-3-030-95929-6_22.
4. Bureva V., L. Atanassova, K. Atanassov, A. Delkov (2019). Application of the Game Method for Modelling for Locating the Wildfire Ignition Point, Proceedings of the 4th International Conference on Numerical and Symbolic Computation Developments and Applications, 11-12 April 2019, ISEP – Instituto Superior de Engenharia do Porto, Porto, Portugal, 397-414.
5. Microsfot Documentation: System.Drawing Namespace. Available online at: https://docs.microsoft.com/en-us/dotnet/api/system.drawing.
6. Ismaili S., S. Fidanova (2019). Application of Game Theory and Evolutionary Algorithms in Solving Conflicts in Social Systems, Int J Bioautomation, 23(3), 293-302, doi: 10.7546/ijba.2019.23.3.000545.
7. Sotirova E., S. Sotirov, A. Dimitrov, K. Atanassov (2013). On Some Applications of the Game Method for Modelling: Part 3: Simulation of Oil Transformation in Marine Environment, Proc Jangjeon Math Soc, 16(2), 293-300.

Senior Assist. Prof. Deyan G. Mavrov, Ph.D.
E-mail: dg@mavrov.eu

Deyan Mavrov studied at the Burgas Free University from 2006 to 2012, where he received a B.Sc. degree in Computer Science and a M.Sc. degree in Business Information Tenchnology. Since 2012, he has been working at the Prof. Dr. Asen Zlatarov University of Burgas, where he also defended his Ph.D. Thesis in 2016. His research interests include finding new applications of Index matrices and intuitionistic fuzzy sets and developing software implementations.

Assoc. Prof. Veselina Bureva, Ph.D.
E-mail: vbureva@btu.bg



Veselina Bureva graduated in Communication and Information Systems at "Konstantin Preslavsky" University of Shumen. In 2014 she completed Ph.D. degree in Computer Systems and Technologies Department at Prof. Dr. Assen Zlatarov University, Burgas. Since 2019, she has been an Associate Professor in the Department of Computer Systems and Technologies at the Faculty of Technical Sciences at the Prof. Dr. Assen Zlatarov University. Her current research interests are in the field of big data and data science, index matrices, generalized nets modelling, etc.

© 2022 by the authors. Licensee Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).