Proceedings of the Jangjeon Mathematical Society
29 (2026), No. 1, pp. 37 - 50

www.jangjeon.or.kr
http://dx.doi.org/10.17777/pjms2026.29.1.004

A generalized net model of the intellectual game Lines

Evgeni Nonchev[1] **and Krassimir Atanassov**[2]

[1] "Prof. Asen Zlatarov" University
Burgas–8010, Bulgaria
and
"EXPOZY" LTD, Sofia 1303, Bulgaria
e-mail: *enonchev@gmail.com*

[2] Department of Bioinformatics and Mathematical Modelling
Institute of Biophysics and Biomedical Engineering
Bulgarian Academy of Sciences
Acad. G. Bonchev Str., Bl. 105, Sofia–1113, Bulgaria
e-mails: *k.t.atanassov@gmail.com*, *krat@bas.bg*

ABSTRACT: A Generalized Net (GN) model of the logic game Lines is described. This model gives a particular answer to the Open problem: can a GN represent an intellectual game? The model creates a basis for expanding the possibilities of the game Lines and provides an idea for GN-interpretation of other adaptive and intelligent games. The emphasis is placed on validating the player's moves and recognizing configurations in the game environment, as well as on the integration of classical algorithms such as A* within a GN-model.

KEYWORDS: Game Lines, Generalized net, Intellectual game

AMS Classification: $65Q85$

1 Introduction

Logical games are an important tool for developing cognitive and strategic skills, and Lines occupies a special place among them due to the symbiosis between simple rules and significant strategic depth. The formalization of game processes through Generalized Nets (GN, [1, 2, 5, 11, 12]) provides new opportunities for algorithmic research, automation, and adaptability of intellectual games. GNs extend the classical Petri nets by introducing tokens with initial and subsequent characteristics and Index Matrices (IM, [6]) with elements — predicates specifying the conditions of transitions, which allows modeling of complex processes, including game scenarios.

The article presents a detailed description of the logic game Lines [14] using GNs. Ideas for modifying and extending the original game Lines are discussed. This is the first example of an GN-model of an intellectual game, which provides a partial solution to the problem "can every intellectual game be described with a GN?" formulated in 1991 in [3, 4].

2 A short description of the game Lines

The digital logic game Lines is a modern computer adaptation of classic game models aimed at activating logical thinking, spatial orientation, and strategic planning. The game is accessible via the web platform [14, 15], without the need to install or create a user profile, which facilitates its use in a variety of educational and informal settings. The playing field is shaped like a square grid (usually 9×9 cells), with each cell being either empty or occupied by a colored ball. At the start of each new session, several balls of different colors appear in random positions. The player is free to move the balls diagonally and in rows, as long as there is a free path between the selected starting and ending positions. The main objective of the game is to arrange at least five balls of the same color in a straight line (horizontal, vertical, or diagonal). When a line is successfully formed, the balls are removed from the field and the player is awarded points.

A key element of the game mechanics is that after each move that does not result in the elimination of balls, new balls are added to random cells. This creates increasing difficulty as the playing space gradually fills up and the possibilities for strategic moves decrease. The game ends when there are no more free positions or valid moves.

From an interface point of view, Lines stands out with its clean and intuitive design. The color differentiation of the balls is clearly recognizable, and the visualization of the next three balls that will appear on the playing field allows the user to anticipate and plan future actions. This functional solution increases the strategic depth of the game by encouraging the formation of cognitive scenarios in which the player evaluates multiple possible move sequences before committing to an action. This anticipatory interface mechanism not only enhances player agency but also fosters metacognitive engagement—prompting users to simulate, compare, and optimize potential outcomes in working memory. By externalizing future game states through the preview of upcoming elements, Lines reduces cognitive load associated with uncertainty while simultaneously elevating decision-making from reactive pattern matching to deliberate tactical planning.

3 A generalized net model

The GN model (see Fig. 1) describes the dynamics of the game Lines — every action of the player, every change in the configuration of the game field, and every update of the game result. The GN model is of the minimal reduced GN type, since in its current form it is not necessary to take into account the time taken by the player or the computer to make their moves. This will be discussed in the Conclusion. Furthermore, it is not necessary to introduce priorities for transitions, places, and tokens in the model, and the capacity of each place is 1.

The GN contains 8 transitions, 21 places, and 3 tokens (see Fig. 1):

* the token π represents the player who initially determines the parameters of the game (field size, number of balls, minimum number of balls to be removed), and through its subsequent characteristics determines the state of the current

move;

* the token φ represents the playing field with a form of a grid with square cells, and the positions of the balls after each move of the player or the program;

* the token κ represents the counter of the points that the player receives.

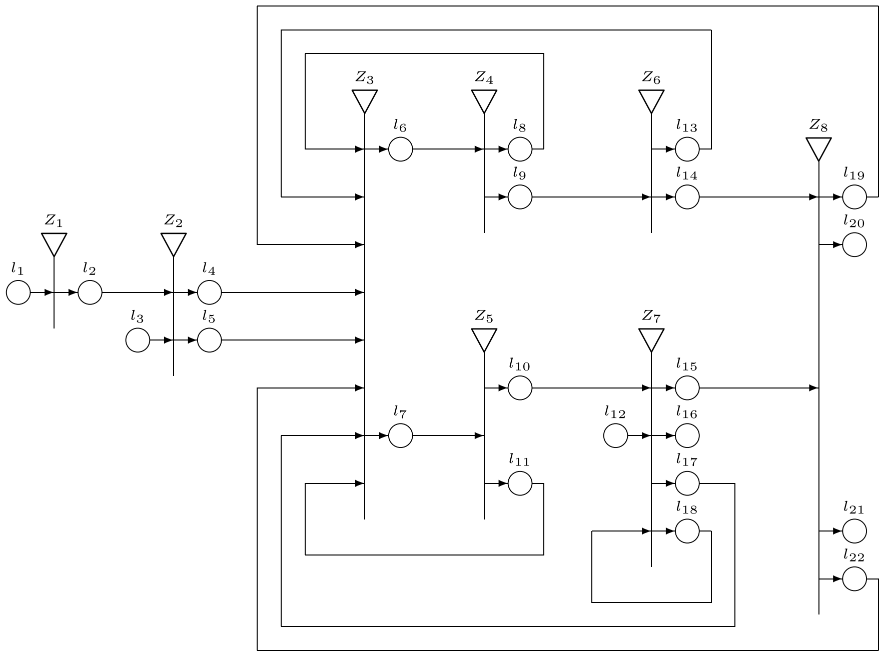


Fig. 1: The GN-model

In the first time-moment of the GN-functioning, the token π enters place l_1 with the initial characteristic

"*start of the game*".

The first GN-transition has the form:

$$Z_1 = \langle \{l_1\}, \{l_2\}, r_1 \rangle,$$

where

$$r_1 = \frac{\begin{array}{c|c} & l_2 \end{array}}{\begin{array}{c|c} l_2 & true \end{array}}.$$

The token π enters place l_2 with the characteristic

"*field size; initial number of balls; number of balls that must be generated on the next step of the game; minimum number of balls that can disappear in one step of the game* ".

In the second time-moment of the GN-functioning, the token φ enters place l_3 without an initial characteristic. The second GN-transition has the form:

$$Z_2 = \langle \{l_2, l_3\}, \{l_4, l_5\}, r_2 \rangle,$$

where

$$r_2 = \begin{array}{c|cc} & l_4 & l_5 \\ \hline l_2 & true & true \\ l_3 & false & true \end{array}.$$

The token π from place l_2 enters place l_4 without a new characteristic and the token φ enters place l_5 with the characteristic

"*a field with sizes shown in the last characteristic of the token π and balls (their number is determined in the second characteristic of the token π), randomly placed in the field cells*".

The next transitions have the following forms:

$$Z_3 = \langle \{l_4 l_5, l_8, l_{11}, l_{13}, l_{17}, l_{19}, l_{22}\}, \{l_6, l_7\}, r_3 \rangle,$$

where

$$r_3 = \begin{array}{c|cc} & l_6 & l_7 \\ \hline l_4 & true & false \\ l_5 & false & true \\ l_8 & true & false \\ l_{11} & false & true \\ l_{13} & true & false \\ l_{17} & false & true \\ l_{19} & true & false \\ l_{22} & false & true \end{array}.$$

The token π from each one of the places l_4, l_8, l_{13}, l_{19} enters place l_6 with the characteristic

"*a selected ball and the location where the player wants it to be moved*",

while token φ from each one of the places $l_5, l_{11}, l_{17}, l_{22}$ enters place l_7 without a new characteristic.

$$Z_4 = \langle \{l_6\}, \{l_8, l_9\}, r_4 \rangle,$$

where

$$r_4 = \begin{array}{c|cc} & l_8 & l_9 \\ \hline l_6 & W_{6,8} & W_{6,9} \end{array},$$

where
$W_{6,8} =$ "the location determined by the player is incorrect",
$W_{6,9} = \neg W_{6,8}$,
where $\neg P$ is the negation of predicate P.

When the predicate $W_{6,8} = true$, the token π from place l_6 enters place l_8 with the characteristic

"*a new selected ball and the location where the player wants it to be moved*

or the player like to move the previously determined ball to another location

for this ball".

When the predicate $W_{6,9} = true$, the token π from place l_6 enters place l_9 without a new characteristic.

$$Z_5 = \langle \{l_7\}, \{l_{10} l_{11}\}, r_5 \rangle,$$

where

$$r_5 = \begin{array}{c|cc} & l_{10} & l_{11} \\ \hline l_7 & W_{7,10} & W_{7,10} \end{array},$$

where
$W_{7,10} =$ "the location determined by the player is correct",
$W_{7,11} = \neg W_{7,11}$.

When the predicate $W_{7,10} = true$, the token φ from place l_7 enters place l_{10} without a new characteristic.

When the predicate $W_{7,11} = true$, the token φ from place l_7 enters place l_{11} without a new characteristic (because the location where the player wants to move some ball is incorrect).

$$Z_6 = \langle \{l_9\}, \{l_{13} l_{14}\}, r_6 \rangle,$$

where

$$r_6 = \begin{array}{c|cc} & l_{13} & l_{14} \\ \hline l_9 & W_{9,13} & W_{9,14} \end{array},$$

where
$W_{9,13} =$ "there is a configuration on the field containing at least the minimum number of balls that must disappear",
$W_{9,14} = \neg W_{9,13}$.

When the predicate $W_{9,13} = true$, the token π from place l_9 enters place l_{13} with the characteristic

"*a new selected ball and the location where the player wants it to be moved*".

When the predicate $W_{9,14} = true$, the token π from places l_9 enters place l_{14} without a new characteristic.

In the second time-moment of the GN-functioning, the token κ enters place l_{12} with the initial characteristic

"0".

This token represents the initial, current and final counter state.

The last two GN-transitions have the forms:

$$Z_7 = \langle \{l_{10}, l_{12}, l_{18}\}, \{l_{15}, l_{16}, l_{17}, l_{18}\}, r_7 \rangle,$$

where

$$r_7 = \begin{array}{c|cccc} & l_{15} & l_{16} & l_{17} & l_{18} \\ \hline l_{10} & W_{10,15} & false & W_{10,17} & false \\ l_{12} & false & false & false & true \\ l_{18} & false & W_{18,16} & false & W_{18,18} \end{array},$$

where
$W_{10,15} =$ "there is not a configuration on the field containing at least the minimum number of balls that must disappear",
$W_{10,17} = \neg W_{10,15}$,
$W_{18,16} =$ "the game is over",
$W_{18,18} = \neg W_{18,16}$.

When the predicate $W_{10,15} = true$, the token φ from place l_{10} enters place l_{15} with the characteristic

"*new balls (their number is determined in the second characteristic of the token π), randomly placed in empty cells of the field*".

When the predicate $W_{10,17} = true$, the token φ from place l_{10} enters place l_{17} with the characteristic

"*the playing field without the balls to be removed, according to the second characteristic of the token π*".

When the predicate $W_{18,16} = true$, the token κ from place l_{18} enters place l_{16} with the characteristic

"*final number of points received by the player*".

When the predicate $W_{18,18} = true$, the token κ continues to stay in place l_{18} with the characteristic

"*the previous number of points plus the new points received by the player*".

$$Z_8 = \langle \{l_{14}, l_{15}\}, \{l_{19}, l_{20}, l_{21}\}, r_8 \rangle,$$

where

$$r_8 = \begin{array}{c|cccc} & l_{19} & l_{20} & l_{21} & l_{22} \\ \hline l_{14} & W_{14,19} & W_{14,20} & false & false \\ l_{15} & false & false & W_{15,21} & W_{15,22} \end{array},$$

where
$W_{14,19} =$ "the playing field is not filled and the player can continue the game",
$W_{14,20} = \neg W_{14,19}$,
$W_{15,21} =$ "the playing field is filled and the game ends",
$W_{15,22} = \neg W_{15,21}$.

When the predicate $W_{14,19} = true$, the token π from place l_{14} enters place l_{19} with the characteristic

"*a new selected ball and the location where the player wants it to be moved*".

When the predicate $W_{14,20} = true$, the token π from place l_{14} enters place l_{20} with the characteristic

"the game ends and the player has earned the points indicated as the last . characteristic of the token κ".

When the predicate $W_{15,21} = true$, the token φ from place l_{15} enters place l_{21} without a new characteristic.

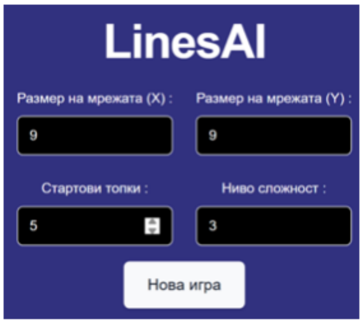
When the predicate $W_{15,22} = true$, the token φ from place l_{15} enters place l_{22} with the characteristic

"new balls (their number is determined in the second characteristic of the token π), randomly placed in empty cells of the field".

The GN-model described above provides an unambiguous interpretation of game actions, traceability of processes, and the possibility of integrating intelligent extensions.

4 Algorithmic implementation of the GN model for game Lines

The algorithmic approach not only automates the game process, but also allows for easy adaptation to extensions such as different difficulty levels, integration of an intelligent opponent, and automated training within the GN. In Fig. 2, three situational screens from the implementation of the Lines game using JavaScript are shown.



Размер на мрежата (X) : 9
Размер на мрежата (Y) : 9
Стартови топки : 5
Ниво сложност : 3

Figure 2. Setting the parameters of the game Lines

The figures show a screen view of the implementation of the game Lines. Fig. 2 shows the game configuration menu, where parameters such as grid size (X and Y), initial number of balls, and difficulty level can be set. Fig. 3 and Fig. 4 show the playing field, consisting of a square grid with colored balls on it, as well as a configuration of single color balls. Fig. 2 illustrates the initial configuration with scattered balls, and Fig. 4 shows a game situation in which lines of identical colour elements have been formed. This visualisation demonstrates the relationship between the configuration parameters and the dynamics of the game process.

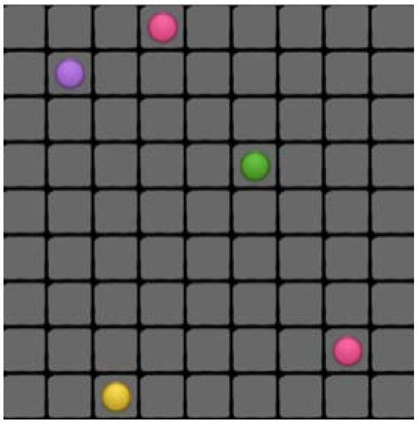


Figure 3. Initial screen of the game Lines playing field

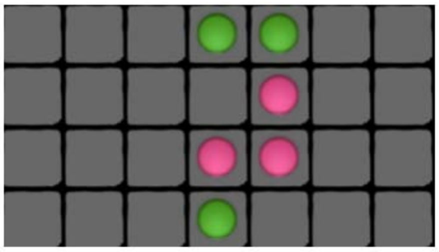


Figure 4. Configuration of the fields in the game Lines

4.1 Predicates and move validation

An important component of the GN-model are the predicates embedded in the IMs of the transition. They control the dynamics of the tokens and set the conditions under which it is possible to move from input to output places. Each predicate is a logical expression that combines a check of the game rules and the current state of the playing field.

Formally, in terms of Lines is architecturally realized through a lightweight, client-centric front-end technology stack grounded in HTML5, CSS3, and ECMAScript (JavaScript), facilitating zero-installation, registration-free execution within standards-compliant modern web browsers. This client-side paradigm mirrors the technical infrastructure prevalent in contemporary browser-based gaming ecosystems — including online casino platforms — where stringent requirements for runtime performance, adaptive responsiveness, and cross-platform interoperability are paramount.

Leveraging HTML5's native multimedia APIs (e.g., Canvas, Web Audio, and Media Source Extensions), the application delivers rich interactive experiences without dependency on legacy plugins such as Adobe Flash. Concurrently, JavaScript orchestrates dynamic game-state management, event-driven user interactions, and real-time behavioral logic via asynchronous execution models and DOM manipulation, ensuring fluid UX even under constrained computational conditions.

The system's minimal server-side footprint—predominantly limited to static asset delivery and optional telemetry—coupled with an optimized, modular codebase, renders Lines exceptionally bandwidth-efficient. This design enables robust deployment across heterogeneous client environments, including low-resource devices and regions with intermittent or low-bandwidth connectivity. Compatibility spans desktop, tablet, and mobile form factors via responsive viewport adaptation and progressive enhancement strategies.

Consequently, Lines emerges as a pedagogically viable instrument for educational contexts characterized by infrastructural constraints, device diversity, or scalability demands — offering equitable, platform-agnostic access without compromising functional richness or pedagogical interactivity.

For example:

- $W_{6,8}$: checks whether the position selected by the player for moving the ball is invalid (e.g., outside the field, occupied cell, or no path). This predicate is described as follows:

  $$W_{6,8} = \neg IsValidMove(selected_ball, target\_cell, board)$$

  and it determines the possibility for the token π, associated with the ball selection, to move from the input to the output places;

  $$IsValidMove(selected\_ball, target\_cell, board)$$

  is a logical function that checks whether the selected ball move is valid according to the rules of the Lines game — for example, whether the target cell is free, located within the field, and whether there is a path from the selected ball to it. Therefore, if

  $$IsValidMove\ returns$$

  is true (valid move), then predicate $W_{6,8}$ is false, and vice versa — if the move is invalid, predicate $W_{6,8}$ becomes true.

- $W_{7,10}$: determines whether moving the ball is permissible given the current configuration and the rules of the game. The Lines game implementation uses a modification of the A* algorithm to search for the shortest free path between two cells. If such a route exists, the predicate is true:

  $$W(7, 10) = A * (start, target, board) \neq \emptyset.$$

  Therefore, the move is valid if the A* algorithm finds at least one free path between the selected ball (*start*) and the target cell (*target*) on the current configuration of the board (*board*). That is, the predicate is true when there is an accessible route; if there is no such route, the value is false.

- $W_{9,13}$: checks whether, after the move, a new line (horizontal, vertical, or diagonal) has been formed with at least the minimum required number of balls of the same color (for example, as in the original game this number is 5) that can be removed. Therefore,

  $$W_{9,13} = DetectLine(last\_move, board, min\_length)$$

  means that after the last move ($last\_move$), the model checks whether a line of balls of the same color with a length at least equal to the specified threshold (min_length) has appeared on the board ($board$). If such a line exists, the predicate is true and the removal of the corresponding balls and updating of the result is triggered; if there is not enough long sequence, the predicate is false and the game continues without removal of balls.

- $W_{10,15}$: determines the absence of a line to be removed, which leads to the addition of new balls to the board. Therefore,

  $$W_{10,15} = \neg W_{9,13}.$$

- $W_{18,16}$: determines the end of the game (for example, when all cells are filled or there are no possible moves). Therefore,

  $$W_{18,16} = (FreeCells(board) = 0) \vee (\neg HasValidMoves(board))$$

  formalizes the condition for the end of the game.

  $$FreeCells(board) = 0$$

  means that there are no free cells on the board; and

  $$\neg HasValidMoves(board)$$

  means that there are no valid moves available to the player. The predicate is true if at least one of the two conditions is met: the board is completely filled or there are no possible moves. In this case, the model stops the cycle and moves to the “End of game” state.

4.2 Pathfinding algorithm (A* algorithm)

Within the transitions in the GN, especially when validating the movements of the cores, the A* algorithm (see,, e.g., [13]) is integrated, which determines the existence of a free route between a selected ball and a target cell. In this way, each core transition is subject to the predicates in the indexed matrix and is objectively verifiable in real time.

- Input: coordinates of the selected token (associated with a ball), coordinates of the target cell, and the current configuration of the playing field;

- Process: the A* evaluation function is used, which combines heuristics (Manhattan or Euclidean distance) and accumulated costs, with the result interpreted in the predicates $W_{7,10}$;

- Output: a list of cells representing a valid path, if one exists; an empty list if no accessible trajectory exists.

In this context, the A* algorithm finds the shortest path on the board by calculating a value for each cell n:

$$f(n) = g(n) + h(n),$$

where $g(n)$ is the actual cost to the cell and $h(n)$ is a heuristic estimate of the remaining distance (in Lines, the Manhattan metric or Euclidean distance is used).

The process starts from the selected ball (start) and searches for the target cell (target), maintaining two sets: $OPEN$ (cells to be considered) and $CLOSED$ (already processed). At each step, the cell with the lowest value of the function f is selected. If it is the target, the path is found; if there are no more cells to consider, no path exists. In the GN-model, the result is interpreted in the formal predicate $W_{7,10}$:

- if
  $$A * (start, target, board) \neq \emptyset,$$
  there is a valid path and the move is permissible;
- if the result is $\emptyset$ — there is no path and the move is blocked.

4.3 Detection of lines to be removed

After each successful move, the GN-model activates a corresponding predicate from the indexed matrix, which analyzes the state of the playing field for the presence of a line with the minimum required length of balls of the same color. For each direction (horizontal, vertical, diagonal), the following check is performed:

$$DetectLine(x, y, color, board, min_length) = \exists direction[consecutive(x, y,$$

$$color, direction) \geq min\_length.$$

If such a configuration is found, the line is removed and the result is updated through the κ-token. Otherwise, the process moves on to the next move.

4.4 Modelling the game cycle

The GN-model provides a complete game cycle in which the sequence of transitions describes the dynamics of the tokens in the model:

1 initialization (Z_1, Z_2) — setting parameters and forming the initial configuration of the field through the input characteristics of the tokens;

2 move selection and validation (Z_3, Z_4, Z_5) — moving the tokens through selection positions, determining the validity of the move through predicates in the IMs;

3 line check (Z_6) — analyzing the configuration using predicates for the presence of sequences of balls in a row, column, or diagonal with a minimum length;

4 updating the result (Z_7) by adding the new token κ, which takes into account the change in points achieved by the player;

5 check for end of game (Z_8) — evaluation using predicates for completeness of the playing field or lack of possible moves, leading to the end of the model.

5 Prospects for expansion and applications of the GN-model in game Lines

5.1 Opportunities for automation and intelligent agents

The formalization of game Lines through GN provides a basis for automating the game process. The GN-model not only ensures transparency of the logic of each action, but also facilitates the implementation of elements from the artificial intelligence toolkit as an autonomous player or strategy analyzer. By expanding the characteristics of the token π, different behavior profiles of game Lines can be implemented, for example:

- a stochastic opponent, where decisions on moves are made based on probability rules;
- heuristic or strategic player, which uses algorithms to maximize the result based on predicting the next game states;
- Learning agents, integrated through reinforcement learning or neural networks, optimize strategy based on accumulated experience. The adaptive nature of the agents implements a process of game strategy optimization based on accumulated experience and feedback from previous moves. This allows agents to improve their behavior over time and increase the effectiveness of their decisions in different game situations. In the GN-model, these agents will be represented by new ones, for example, α-tokens, which will collect information about the environment and/or player behavior through their characteristics. Another token (β-token), representing a second type of agents, will process the information collected in the characteristics of the α-tokens for the purpose of documenting the process and/or forecasting. Currently, there are only three GN-models of multiagent systems [8, 9, 10].

5.2 Integration into educational and research environments

The GN-model of the game Lines can be easily used to develop interactive training systems in which the process of problem solving and strategy implementation can be analyzed and visualized in real time.

In educational practice, such a model serves not only as a gaming environment, but also as a demonstration platform for analyzing algorithms, cognitive processes, and decision-making dynamics.

In addition, the GN-approach allows for simulations at different levels of difficulty, adaptation to the age and competence of the participants, as well as the conduct of controlled cognitive experiments – for example, studying decision-making time, modeling errors, or analyzing strategic choices with limited resources.

5.3 Scalability and modularity

A significant advantage of the GNs is its open and modular architecture, which allows for easy expansion of the model without compromising its internal consistency. This is achieved through hierarchical operators defined over GNs in [2, 5]. This allows for the introduction of new functional mechanisms, such as:

- adding additional rules by integrating new types of cores (e.g., special balls, bonus elements, or obstacles), as well as defining alternative conditions for ending the game,
- implementing multiplayer modes by including more π-tokens, which allows simulations of competitive or cooperative scenarios.

As can be seen from the description of the game Lines in its extended form, we propose the generation of dynamic or adaptive game boards by changing sizes, starting configurations, and difficulty parameters, which can be controlled directly through the GN toolkit. It provides flexibility to the model and allows the creation of different interaction scenarios. Such an approach expands the applicability of the model by creating conditions for simulating diverse game environments and adapting to the competence level of the participants.

This modularity illustrates the robustness and flexibility of the GN-formalism and makes it applicable not only to static but also to evolving game environments.

6 Conclusion

The formalization and algorithmic modeling of the logic game Lines using a GN shows that the GN is not only a theoretical apparatus but also a practical tool for implementing complex game scenarios, i.e., the GNs can be used as a means of describing intellectual games — a problem that has remained unsolved since 1991 (see [3, 4, 7]). This opens up a new area of application for the GNs. Until now, they have been applied to model various components of artificial intelligence tools (expert systems, neural networks, etc., see [7]), in biology and medicine, in the chemical industry and economics, in transport and telecommunications, and in many other areas (see, e.g., [1, 11, 12]).

In conclusion, we must mention that the developed approach can be adapted to a wide class of logical and strategic games with similar dynamics, such as XO Game, Tetris, Minesweeper and others.

References

[1] Alexieva, J., E. Choy, E. Koycheva. Review and bibloigraphy on generalized nets theory and applications. In:- A Survey of Generalized Nets (E. Choy, M. Krawczak, A. Shannon and E. Szmidt, Eds.), Raffles KvB Monograph, No. 10, 2007, 207–301.

[2] Atanassov, K. Generalized Nets, World Scientific, Singapore, 1991.

[3] Atanassov K. Open problems in the theory of generalized nets. Preprint IM-MFAIS-1-91, Sofia, 1991.

[4] Atanassov, K. Introduction in Generalized Nets Theory. Burgas, Pontica Print, 1992 (in Bulgarian).

[5] Atanassov, K. On Generalized Nets Theory, Prof. M. Drinov Academic Publ. House, Sofia, 2007.

[6] Atanassov, K. Index Matrices: Towards an Augmented Matrix Calculus, Springer, Cham, 2014.

[7] Atanassov. K. Generalized Nets and Intuitionistic Fuzziness in Data Mining. "Prof. Marin Drinov" Academic Publishing House, Sofia, 2020.

[8] Trifonov, T., K. Atanassov. On some generalized net models of multiagent systems. Ministry of Education and Research, Vol. VII, 2002. No. 37, 49-52.

[9] Trifonov T., K. Atanassov. On some generalized net models of multiagent systems - Part II. Proceedings of the Third Int. Workshop on GNs, Sofia, 1 Oct. 2002, 49-58.

[10] Trifonov, T., K. Atanassov. On some generalized net models of multiagent systems (Part 3). Issues in Intuitionistic Fuzzy Sets and Generalized Nets, Vol. 2 (K. Atanassov, J. Kacprzyk and M. Krawczak, Eds.), Wydawnictwo WSISiZ, Warsaw, 2004, 59-66.

[11] Zoteva, D. and M. Krawczak. Generalized Nets as a Tool for the Modelling of Data Mining Processes. A Survey. Issues in Intuitionistic Fuzzy Sets and Generalized Nets, Vol. 13, 2017, 1-60.

[12] Zoteva, D., N. Angelova. Generalized Nets. An Overview of the Main Results and Applications. In:- Research in Computer Science in Bulgarian Academy of Sciences (K. Atanassov, Ed.), Springer, Cham, 2020.

[13] Introduction to the A* Algorithm, https://www.redblobgames.com/pathfinding/a-star/introduction.html, 7 Sept. 2025

[14] https://lines.myexpozy.com, 7 Sept. 2025

[15] Lines.NET Game. https://www.codeproject.com/Articles/7031/ Lines-NET-game, 7 Sept. 2025