Author(s): Terziyan, Vagan; Terziian, Artur; Vitko, Oleksandra

Title: Creative and Adversarial Cellular Automata for Simulating Resilience in Industry 5.0

Year: 2024

Version: Published version

Please cite the original version:

Terziyan, V., Terziian, A., & Vitko, O. (2024). Creative and Adversarial Cellular Automata for Simulating Resilience in Industry 5.0. Procedia Computer Science, 232, 1491-1502. https://doi.org/10.1016/j.procs.2024.01.147

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 232 (2024) 1491–1502

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on Industry 4.0 and Smart Manufacturing

Creative and Adversarial Cellular Automata for Simulating Resilience in Industry 5.0

Vagan Terziyan [a,*], Artur Terziian [b], Oleksandra Vitko [c]

[a] *Faculty of Information Technology, University of Jyväskylä, 40014, Jyväskylä, Finland*
[b] *Private Lyceum "Professional", 61024, Kharkiv, Ukraine*
[c] *Department of Artificial Intelligence, Kharkiv National University of Radio Electronics, 61166, Kharkiv, Ukraine*

Abstract

Emerging Industry 5.0 pushes advanced automation towards resilient solutions with enhanced human role. Resilience as an ability to sustain processes in the face of disruptions and adversarial attacks requires careful modelling and simulation. Cellular automata are efficient mathematical models used to simulate the behavior of complex systems, which change their state based on a set of predefined rules. In this paper, we suggest several updates to cellular automata (particularly Conway's "Game of Life") to address resilience. These include "Life and Creation", "War and Peace", and their hybrid "War and Creation" capable of addressing the important components of resilience, such as controllable creativity and adversarial interactions. Inherited in these updates and known advantages of cellular automata, such as simplicity, emergent behavior, parallelism, and adaptability, makes it a powerful simulation tool for a wide range of Industry 5.0 systems that involve humans, smart infrastructure, their complex and adversarial interactions, safety, and resilience.

Keywords: cellular automata; simulation; Industry 5.0; resilience; creative models; adversarial models

1. Introduction

Returning humans back-into-the-loop drives the "Industry 4.0 to Industry 5.0 transformation". Getting value from human-centric nature [1] and, at the same time, addressing emergent resilience concerns [2] are the main dimensions of Industry 5.0 research and development. An extremely complex, highly dynamic and possibly adversarial human-

* Corresponding author. *E-mail address:* vagan.terziyan@jyu.fi

10.1016/j.procs.2024.01.147

machine-automation infrastructure requires appropriate modelling and simulation approaches to foresee and address the resilience concern. In this paper, we are going to study and modify the popular cellular automata (CA) modelling and simulation formalism [3] making it more suitable (i.e., being "creative" and "adversarial" as additional features) to address the issues of resilience in future Industry 5.0 applications.

The rest of the paper is organized as follows: Section 2 describes the connection between CA paradigm and smart manufacturing applications; Section 3 suggests a few modifications to traditional CA to address the major resilience components; and we conclude in Section 4. Related work will be cited throughout the paper.

2. Why Cellular Automata are Useful for Industry 4.0 and Smart Manufacturing?

Cellular automata (CA) [3] are mathematical models that describe the behavior of a system made up of many interacting components, where each component is updated based on a set of rules that depend on its neighboring components. CA models have been used in various applications [4], including Industry 4.0 and smart manufacturing. Cellular automata, depending on their type, can provide valuable insights into various complex systems, helping manufacturers and other organizations make better decisions and optimize their operations for maximum efficiency and profitability.

The specific properties of CA make it a powerful tool for modeling and simulating a wide range of systems, from simple physical systems to complex social and economic ones [5]. By simulating these systems using CA, engineers and designers can optimize their designs, improve their performance, and reduce risks and costs associated with implementing these systems in the real world. These specific properties, which make CA suitable for many applications, including Industry 4.0, smart manufacturing, and smart cities, are as follows: *local interactions* [6] (CA models are based on simple local interactions between individual cells, which can be easily programmed and simulated. These interactions can be described using simple rules and can be easily modified to model complex behaviors); *emergent behavior* [7] (the collective behavior of a CA system emerges from the local interactions of individual cells. This emergent behavior can be complex and difficult to predict, making CA useful for modeling and simulating systems that exhibit complex behavior); *scalability* [8] (CA models can be scaled up or down depending on the complexity of the system being modeled. This makes them useful for modeling systems that range in scale from small, simple systems to large, complex systems); *parallel processing* [9] (CA models can be efficiently simulated using parallel processing techniques. This makes them suitable for simulating large systems with many cells or agents); *modularity* [10] (CA models can be modularized, allowing different components of the system to be modeled separately and then integrated into a larger model. This makes it easier to design and test different components of the system); and *flexibility* [11] (CA models can be easily modified and adapted to model different systems and scenarios. This makes them useful for simulating and testing different designs and scenarios before implementing them in the real world).

Here are some possible applications of CA in Industry 4.0, which include similar steps in the simulation process:

Quality control: CA can be used to simulate the behavior of materials and products [12] under different conditions by modeling the physical and chemical processes that occur within the materials and aiming to identify potential defects and improve the quality of the final product [13]. Simulation process includes: identifying the properties (e.g., density, thermal conductivity, viscosity, etc.) of the material that are relevant to the simulation; identifying the state variables (temperature, pressure, chemical composition, etc.) that describe the behavior of the material; defining the rules that govern how the state variables change over time based on the inputs and feedback in the system; defining the neighborhood, i.e., adjacent molecules, atoms, or particles; simulating the behavior of the material over time using CA simulation software and taking into account the initial conditions, inputs, and feedback loops; analyzing the results and identifying changes in the material properties, chemical reactions, or other phenomena; and, finally, optimizing the material properties for specific applications, such as strength, durability, or conductivity among others.

Supply chain optimization: By modeling the behavior of the supply chain using CA [14], it is possible to optimize the flow of materials and products between suppliers, manufacturers, and customers, helping to reduce costs, increase efficiency and improve customer satisfaction. CA-driven simulation of the supply chain includes: identifying the different stages of the supply chain, including suppliers, manufacturers, distributors, and customers; identifying the state variables (e.g., inventory levels, order quantities, lead times, etc.) that describe the behavior of the supply chain; defining the rules that govern how the state variables change over timed; identifying the neighboring components (e.g., adjacent suppliers, manufacturers, distributors, and customers) that affect the behavior of each component;

simulate the behavior of the supply chain over time using CA simulation software; analyzing the results and identifying bottlenecks, inefficiencies, or other issues in the supply chain; optimizing the supply chain (e.g., adjusting inventory levels, order quantities, lead times, etc.) to minimize costs and improve customer satisfaction.

Process optimization: CA can be used to simulate the behavior of different manufacturing processes [15] (e.g., assembly lines), helping to identify bottlenecks and inefficiencies and optimize the process for maximum production efficiency and product quality. CA-driven process simulation includes: identifying the different stages of the manufacturing process and the inputs and outputs at each stage; identifying the state variables that describe the behavior of the manufacturing process; defining the rules; identifying the neighboring components that affect the behavior of each component (e.g., adjacent machines, products, or raw materials); simulating the process behavior with some CA simulation software; analyzing the results including identifying bottlenecks, inefficiencies, or other issues in the process; optimizing the manufacturing process for maximum efficiency, quality, and safety.

Autonomous systems: CA can be used to model and optimize autonomous systems' (e.g., robots or drones) behavior, aiming to reduce the risk of accidents, improve performance, and increase safety [16]. CA-driven simulation process for autonomous systems includes: identifying the environment in which the autonomous system will operate, including the terrain, obstacles, and other agents; identifying the state variables that describe the behavior of the autonomous system (e.g., position, orientation, speed, acceleration, etc.); defining the rules; identifying the neighboring components that affect the behavior of each component, e.g., nearby obstacles, other agents, or waypoints; simulating autonomous system using CA software; analyzing the results and identifying areas where the system may encounter obstacles or issues with navigation; optimizing the autonomous system behavior for maximum efficiency, safety, and effectiveness.

Smart cities: CA simulations can provide valuable insights into the behavior of smart city systems [17], allowing city planners and engineers to optimize their operations for specific applications and environments, i.e., improve the performance of the city, reduce resource consumption, enhance the quality of life for citizens, among other potential benefits. CA-driven smart city simulation process includes: identifying the smart city system to be modeled (e.g., traffic, energy grids, waste management, etc.); identifying the state variables that describe the behavior of the system (e.g., the flow of traffic, the amount of energy consumed, the amount of waste produced, etc.); defining the rules; identifying the neighboring components (e.g., adjacent buildings, vehicles, infrastructure, etc.); simulating the behavior of the smart city system over time using CA software; analyzing the results and identifying bottlenecks, inefficiencies, or other issues in the system; optimizing the smart city system for maximum efficiency, sustainability, and livability.

The type of CA used in different applications above depends on the specific requirements and characteristics of the system being modeled. There are several types of CA, each with its own unique characteristics and applications. Here are some examples of how different types of CA can be used in industry:

- *Elementary CA* [18] is the simplest type of CA, with only two possible states for each cell (0 or 1). It is often used in industrial applications such as image processing, data compression, and error correction.
- *Totalistic CA* [19] is a type of CA in which the state of a cell depends only on the sum of the states of its neighboring cells. It is often used in industrial applications such as modeling fluid dynamics, crystal growth, and chemical reactions.
- *Stochastic CA* [20] is a type of CA in which the state of a cell is probabilistic rather than deterministic. It is often used in industrial applications such as modeling random processes, financial systems, and stock markets.
- *Continuous CA* [21] is a type of CA in which the states of the cells are continuous rather than discrete. It is often used in industrial applications such as modeling the behavior of materials, heat transfer, and fluid flow.

The specific possibilities of each type of CA depend on its unique characteristics and applications. For example, elementary CA is useful for industrial applications that require simple, binary processing, while totalistic CA is useful for modeling complex physical and chemical processes. Stochastic CA is useful for modeling systems with random behavior, while continuous CA is useful for modeling systems with continuous states.

CA-based game-like models, which we are going to represent in this study, are expected to facilitate the simulation and optimization of human-smart infrastructure interactions, adversarial scenarios, safety measures, and overall system resilience. We envision their adoption as a valuable toolset that empowers industries to enhance proactively their adaptive capacity, risk management strategies, and long-term sustainability. The fusion of advanced automation with human ingenuity, as we are going to demonstrate in our models, positions industries to navigate complex disruptions and adversarial dynamics in the evolving Industry 5.0 paradigm.

3. Adding Resilience to Cellular Automata towards Industry 5.0 Context

CA can also be used to model the resilience of complex systems. Resilience refers to the ability of a system to recover from disturbances or disruptions and maintain its functionality. Emergent Industry 5.0 [22], as a kind of "human – smart industrial infrastructure" resilient ecosystem, is an emerging paradigm that emphasizes the integration of humans, smart machines and infrastructure into the resilient manufacturing process. In addition to human-centricity [23], resilience is an important concern in the Industry 4.0 – Industry 5.0 transformation process [24] due to current global challenges. To model system resilience with CA, the target complex system with emergent properties, such as transportation network, energy grid, or social network, will be represented as a grid of cells, where each cell represents a component or unit of the system. The rules governing the behavior of the cells are designed to simulate the interactions and dependencies between the different components of the system. For example, the cells may represent different parts of a transportation network, such as roads, bridges, and tunnels, and the rules may govern how traffic flows through the network. The ability to model the interactions and dependencies between different components of the system can provide insights into the mechanisms that drive resilience and the factors that contribute to vulnerability. Resilience can be studied by simulating how a system responds to different types of disturbances or failures, such as the failure of a particular component or the disturbances within the connection logic between the components. In the Industry 5.0 context, failures associated with human factors (behavior, interaction, etc.) are also of great importance [25]. By studying the system's response to different types of disturbances, researchers can identify potential vulnerabilities and design strategies to improve the system's resilience. The key advantage of using CA for resilience modeling in Industry 5.0 is that it enables modeling and optimizing the performance of complex, dynamic systems in a way that is both flexible and efficient.

We believe that capabilities of traditional CA for better resilience modelling and simulation could be upgraded if to enable different kind of cells and their interaction, particularly: cells, which represent people, tools, and machines aka "productive forces"; designed products, developed infrastructure, etc. aka "manufacturing results"; adversarial cells as people and infrastructure capable to attack and disrupt/destroy other cells, etc. Rules in such CA will represent some specific relationships, which are important in the resilience context, such as: evolution and self-recovery of productive forces; evolution and recovery of manufacturing results depending on productive forces dynamics; attack and defence activity between adversarial cells representing different actors, etc. In this paper, we are going to represent a couple of dimensions for further evolution of CA towards resilience, particularly the "creativity" dimension (to handle relationships between producers and products) and the "adversarial" dimension (to handle adversarial activity of cells). As the basis to represent further CA upgrades towards simulating resilience, we take the most famous ever CA, i.e., Conway's "Game of Life" [26]. It is based on the concept of a two-dimensional grid of cells that can be either be "alive" or "dead", and are determined by the states of their neighboring cells. Such CA simulates "artificial life" by the rules, which are as follows:

"Game of Life" rules:

Rule 1: Any live cell with fewer than two live neighbors dies (becomes empty), as if caused by underpopulation.
Rule 2: Any live cell with two or three live neighbors continue living in the next generation.
Rule 3: Any live cell with more than three live neighbors dies (becomes empty), as if by overpopulation.
Rule 4: Any dead (or empty) cell with exactly three live neighbors becomes a live cell, as if by reproduction.

These rules are applied simultaneously to every cell in the grid in each generation, creating a new pattern of living and dead cells. This process is then repeated over and over, generating new patterns with each iteration.

3.1. Creativity component of the resilience

The simplest and basic creativity extension for CA would be introducing other than "living" but existing "material" cell, which is supposed to be the result of creative production, design or construction by the living cells. For example, reproduction *Rule 4* above describes how a new live cell appears ("born") replacing a dead cell due to the presence of three live cells (aka "parents") in its neighborhood. In a similar way, we can represent the rule on how a dead cell could be transformed into some kind of "product" (new type of cell, which is neither live nor dead) designed by the neighboring cells' (aka "creators") activity.

Let us introduce a new type of cell called "rock" to be used as an abstract label for existing (not dead) material but not live cells aka a product created (manufactured) by the live cells. Let us modify the basic rule set above, taking into account the new type of cell. Let us also name this extension to the basic "Game of Life" as "Game of Life and Creation". The updated rule set is as follows:

"Game of Life and Creation" ("rock creation" version) rules:

<u>Rule 1</u>: Any live cell with fewer than two live neighbors dies (becomes empty), as if caused by underpopulation.
<u>Rule 2</u>: Any live cell with two or three live neighbors continue living in the next generation.
<u>Rule 3</u>: Any live cell with more than three live neighbors dies (becomes empty), as if by overpopulation.
<u>Rule 4</u>: Any dead (or empty) cell with exactly three live neighbors becomes a live cell, as if by reproduction.
<u>Rule 5</u>: Any dead (or empty) cell with exactly two live neighbors becomes a rock cell, as if by creation (construction).

If applied, these rules are supposed to create at each iteration a new pattern of live, dead, and constructed rock cells. One may notice that the new *Rule 5* not only specifies how rock cells appear but also impact CA by taking away free spaces that could be otherwise used for the live cells reproduction. Rock cells themselves never die nor take part as parents in reproduction.

Consider an example of CA evolution following the new rule set in Fig. 1. Let us describe it using the "Game of Life" glossary (https://wiki.secretgeek.net/glossary-game-of-life). Fig. 1(a) shows the evolution of the "oscillator" (i.e., pattern that cycles through a repeating set of states) known as the "blinker". One may see that blinker's live cells in the "Game of Life and Creation" behave the same way as in the "Game of Life"; however, they created four rock cells during evolution. Fig. 1(b) shows the evolution of four living cells towards a stable position called a "beehive" with eight rocks constructed around. Fig. 1(c) shows the evolution of six living cells known as the "Latin cross", where all the living cells die finally but 12 constructed rocks remain. Fig. 1(d) shows the evolution of two "ants" configurations, which end up in a "block" configuration of four living cells like in the original "Game of Life". However, it produces 36 rocks during the evolution. All evolutionary chains in Fig. 1 lead to a stable configuration.

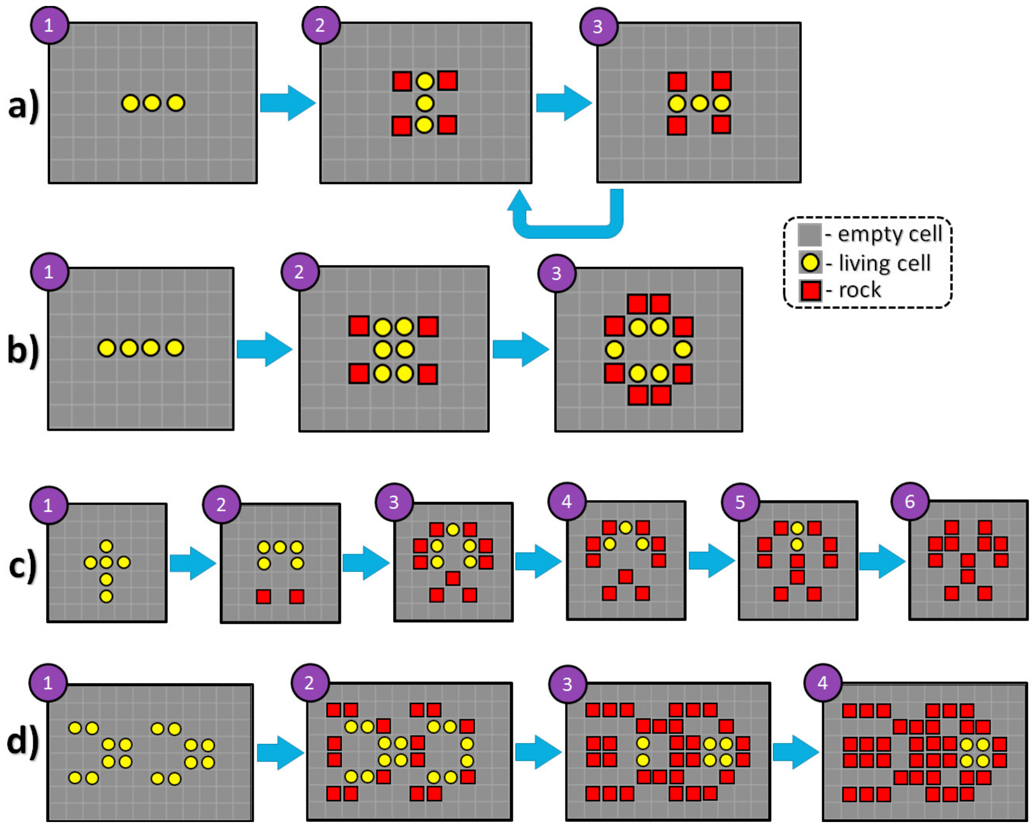

Fig. 1. Evolution within the "rock creation" version of the "Game of Life and Creation": (a) "Blinker" – cycling with four rocks created; (b) Four living cells evolve into "Beehive" and create rocks around; (c) evolution and death of "Latin cross" but with remaining rocks; (d) Evolution of "Ants" into "Block" with many rocks created.

Some configurations of live cells in the Game of Life perform oscillations that recreate the same shape and move at the same time across the grid. Among those, the most famous one is "Glider". Fig. 2 illustrates the evolution of "Glider" in the traditional Game of Life and in the Game of Life and Creation ("rock creation" version). In the former case, the trace of created rocks left by the glider takes away space needed for own recreation, and the glider dies.

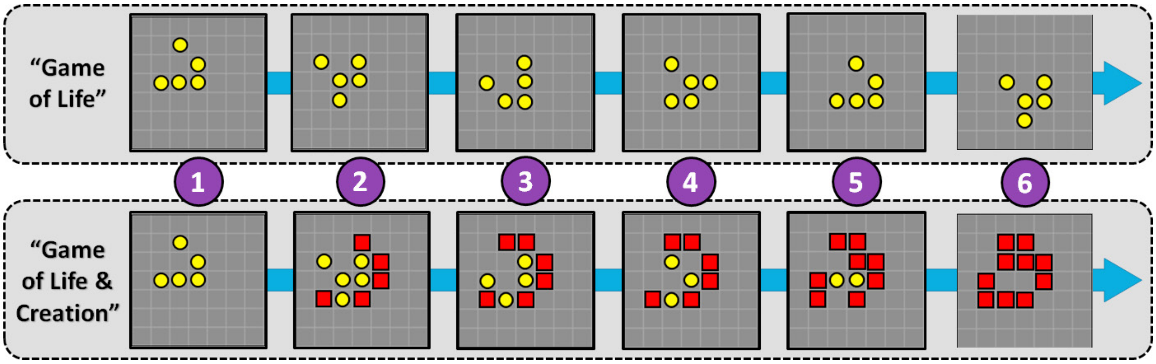

Fig. 2. Evolution ("continuous travel") of the "Glider" in the Game of Life and evolution of it in the "Game of Life and Creation" ("rock creation" version) resulted to death and creation of 10 rocks.

Let us consider another possible and more dynamic option of the Game of Life and Creation named as "egg creation" version, in which the live cells, instead of creating "rocks", lay "eggs", from which, after a strictly defined time $\boldsymbol{T}$, new live cubs hatch. The rules will be as follows:

"Game of Life and Creation" ("egg creation" version) rules:

Rule 1: Any live cell with fewer than two live neighbors dies (becomes empty), as if caused by underpopulation.
Rule 2: Any live cell with two or three live neighbors continue living in the next generation.
Rule 3: Any live cell with more than three live neighbors dies (becomes empty), as if by overpopulation.
Rule 4: Any dead (or empty) cell with exactly three live neighbors becomes a live cell, as if by reproduction.
Rule 5: Any dead (or empty) cell with exactly two live neighbors becomes an "egg" cell with the "delay" parameter t initialized as $t = \boldsymbol{T}$.
Rule 6: Any "egg" cell at each iteration decrements its "delay" parameter by 1. If, as a result, t reaches zero, then the "egg" cell is replaced by the live cell, as if by delayed reproduction.

If applied, these rules are supposed to create at each iteration a new pattern of live, dead, and laid egg cells. Here we have two types of reproduction, the immediate one with three "parents" specified by *Rule 4* and the delayed one with two "parents" (*Rule 5* and *Rule 6*). One may notice that *Rule 5* not only specifies how egg cells appear but also impacts CA by taking away free space that could be otherwise used for immediate live cell reproduction. Egg cells themselves never die until a newborn appears, according to *Rule 6*. The choice of the delay time $\boldsymbol{T}$ can be used to control the game dynamics. If $\boldsymbol{T} \to \infty$, then the "egg creation" version of the game will behave exactly as the "rock creation" version, which means that the latter version can be considered as a special case of the former one.

Example in Fig. 3 illustrates the evolution of "Blinker" in the "Game of Life and Creation" ("egg creation" version) with parameter $\boldsymbol{T} = 2$. One may see that the live population in this game with such settings grows and spreads faster than in the original "Game of Life". Bigger number for $\boldsymbol{T}$ will slow down the evolution.

3.2. Adversarial component of the resilience

The "adversarial" component of the CA-driven resilience simulation addresses special simulation options when different types of cells compete with each other for space and resources, attack each other, recover after attacks, etc. The competition between cells can lead to the emergence of complex patterns and behaviors that are not present in traditional CA models. The rules that govern the behavior of mutually adversarial cells will be more complex than in traditional CA models. For example, the rules may include interactions between neighboring cells, such as competition for space and resources, or physical collisions leading to the emergence of new interesting spatial patterns and structures. The ability to model the dynamics of competition, adversarial attacks, hybrid threats or wars between entities can provide insights into the emergence of patterns and behaviors that are not observable in traditional models.

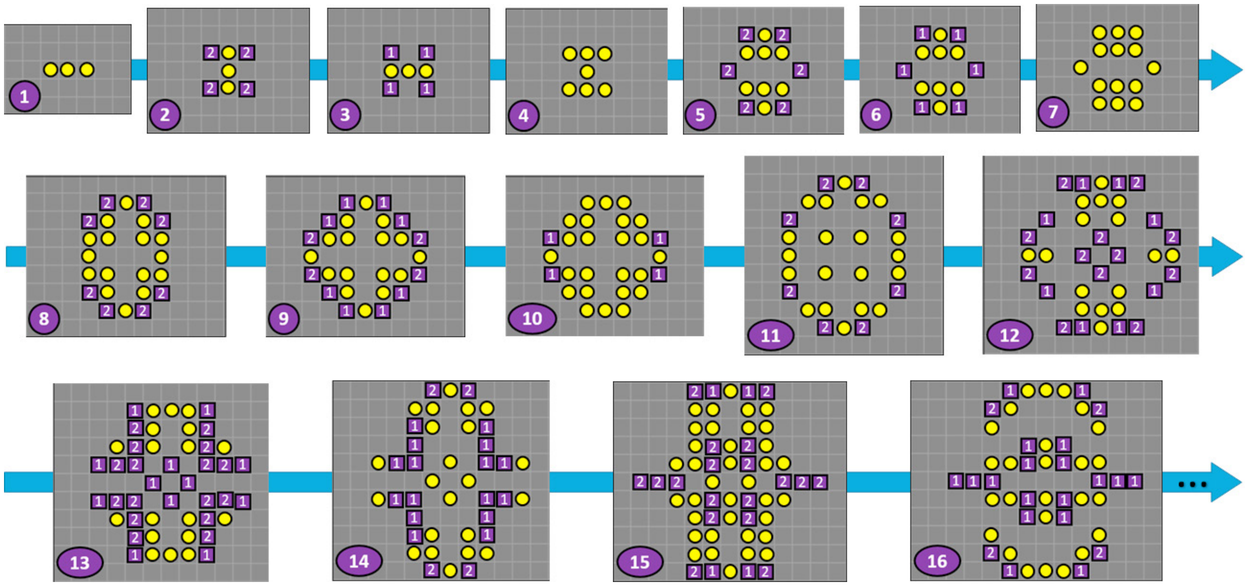

Fig. 3. "Blinker" evolution in the "Game of Life and Creation" ("egg creation" version). "Egg" cells are marked with the violet squares with current value of the delay parameter indicated. Initialization of the delay parameter ***T***=2 makes spreading evolution of the live cells faster.

Let us introduce a simple update of the basic "Game of Life" towards adversarial CA where two (potentially could be more) "teams" of living cells compete with each other. The lattice space will be divided into two "countries" by a border between them, which can be used to locate the original configurations of both teams in each own country. After applying rules, such CA may result in the configuration when one of the teams "occupies" part of the other team's territory and, therefore, moves the border accordingly. We have named such a CA as "War and Peace" and suggested the following rules for it:

"War and Peace" (two teams) rules:

Rule 1: Any live cell from either team with fewer than two live teammate neighbors dies (becomes empty), as if caused by underpopulation.

Rule 2a: Any live cell from either team with two or three live teammate neighbors will continue living in the next generation if the number of adversary neighbors is not more than the number of teammate neighbors.

Rule 2b: Any live cell from either team with two or three live teammate neighbors dies (becomes empty) if the number of its adversary neighbors is more than the number of teammate neighbors, as if by "killing".

Rule 3: Any live cell from either team with more than three live teammate neighbors dies (becomes empty), as if by overpopulation.

Rule 4: Any dead (or empty) cell with exactly three live neighbors ("parents") from either team becomes a live cell and a teammate of its parents, as if by reproduction. If both teams pretend to get a "child" within the same cell, the advantage is given to team, which belongs to the same "country" as the empty cell (defined by the border).

One may see that any configuration of cells will evolve according to the "Game of Life" rules in the case of "peace" (absence of adversaries around). In the situation of teams' collisions ("war"), rules 2 and 4 resolve the conflicts.

An example is shown in Fig. 4 where two gliders (one from "Yellow" team and another from "Blue" team) collide on the border between the corresponding countries. The collision ends up in a stable configuration: two adversarial "Blocks" (aka border posts) on opposite sides of the border.

Consider the example in Fig. 5. Here, the yellow "Glider" violates the border of the blue team and collides with the ship ("Boat" in terms of the Game of Life). After CA stabilizes, the yellow team (transformed into "Block") appears to occupy two cells of the enemy's territory, leaving the "Tub" configuration instead of the "Boat".

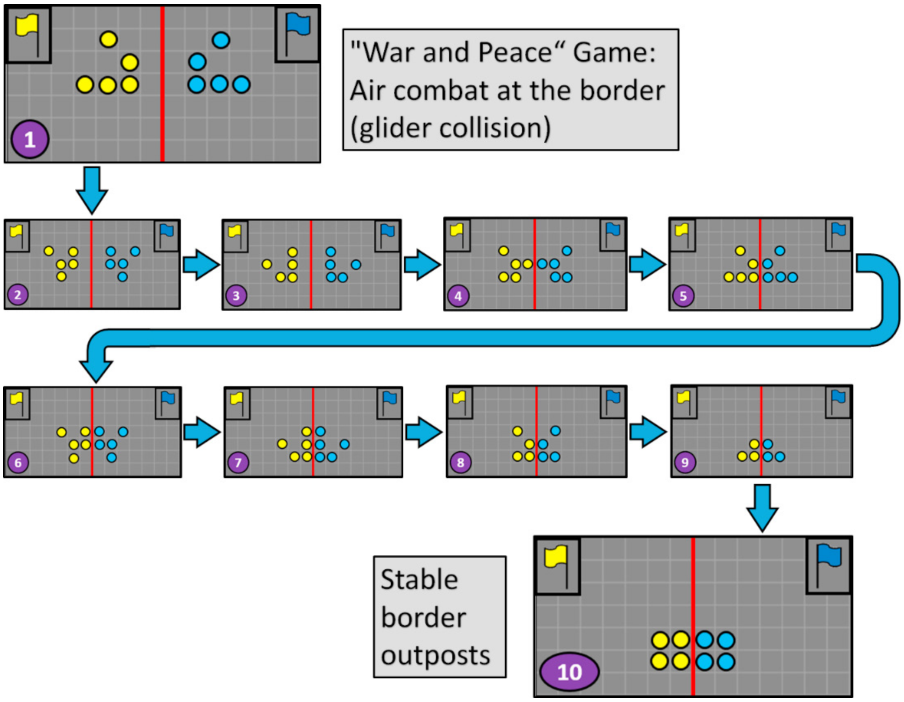

Fig. 4. The collision between two mutually adversarial gliders, which happened close to the border in the “War and Peace” CA.

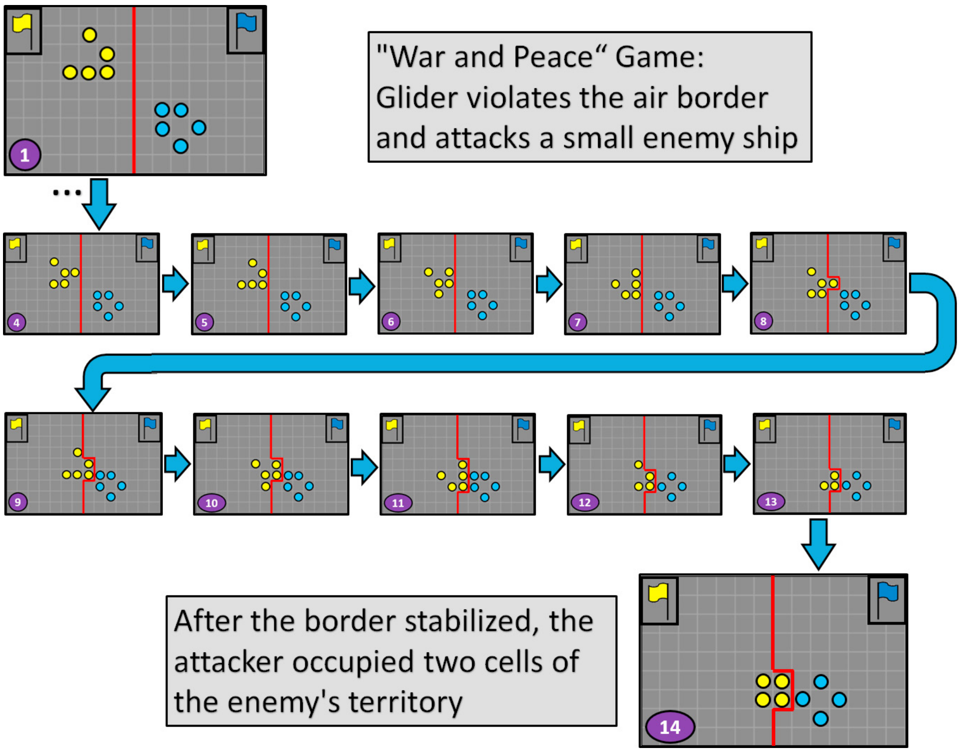

Fig. 5. The collision between the yellow “Glider” and the blue ship (“Boat”), which happened after the glider violated the border and appeared on the territory of blue team in the “War and Peace” CA. After the “fight”, the yellow “Glider”, which finally takes the “Block” configuration, occupied two cells of foreign territory destroying the blue “Boat” into the form of “Tub”.

3.3. Combining Creativity and Adversarial components

CA extensions defined above (i.e., “Life and Creation” and “War and Peace”) can be combined together into a hybrid set of rules giving more simulation flexibility to CA to address the resilience issues.

Consider two types of a hybrid named “War and Creation”, in which the first option would be a combination of the “Life and Creation” (“rock creation” version) CA with the “War and Peace” CA and the second option would be a combination of the “Life and Creation” (“egg creation” version) CA with the “War and Peace” CA.

“Game of War and Creation” (“rock creation” version) rules (see example in Fig. 6):

Rule 1: Any live cell from either team with fewer than two live teammate neighbors dies (becomes empty), as if caused by underpopulation.

Rule 2a: Any live cell from either team with two or three live teammate neighbors will continue living in the next generation if the number of adversary neighbors is not more than the number of teammate neighbors.

Rule 2b: Any live cell from either team with two or three live teammate neighbors dies (becomes empty) if the number of its adversary neighbors is more than the number of teammate neighbors, as if by “killing”.

Rule 3: Any live cell from either team with more than three live teammate neighbors dies (becomes empty), as if by overpopulation.

Rule 4: Any dead (or empty) cell with exactly three live neighbors (“parents”) from either team becomes a live cell and a teammate of its parents, as if by reproduction. If both teams pretend to get a “child” within the same cell, the advantage is given to team, which belongs to the same “country” as the empty cell (defined by the border).

Rule 5: Any dead (or empty) cell with exactly two live neighbors (“creators”) from either team becomes a “rock” cell and inherits the color from its parents’ team, as if by creation. If both teams pretend to create their “rock” within the same cell, the advantage is given to team, which belongs to the same “country” as the empty cell (defined by the border).

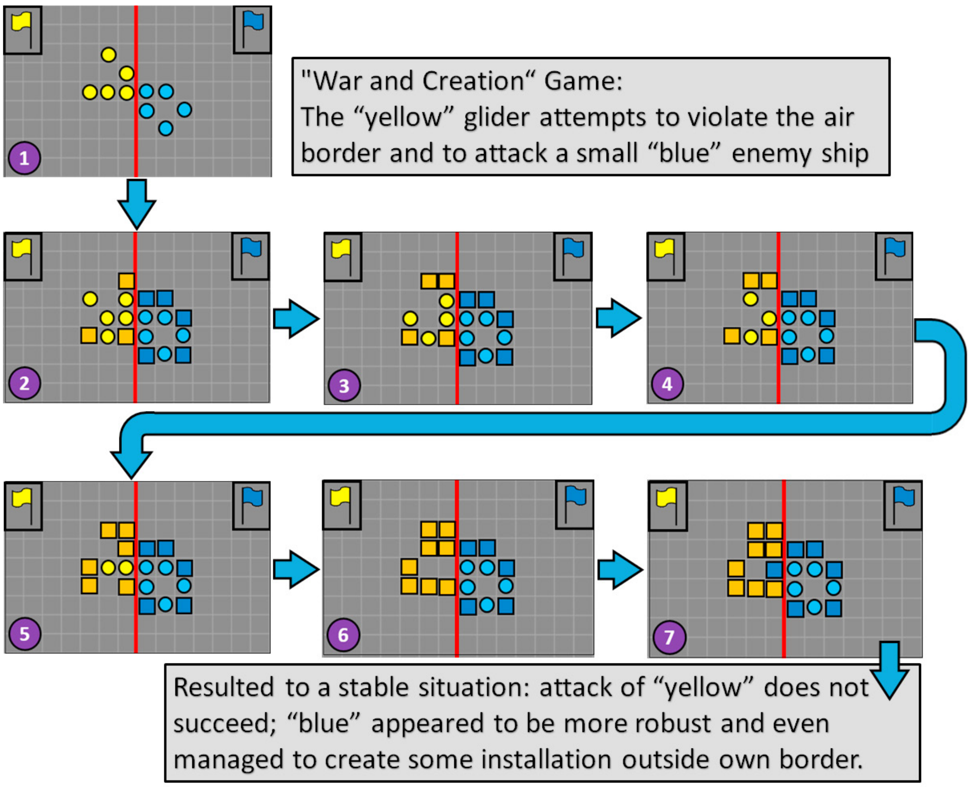

Fig. 6. The collision between the yellow “Glider” and the blue ship (“Boat”), which happened after the glider attempted to cross the border in the “War and Creation” (“rock creation” version) CA. After the “fight”, the yellow live team failed (only rocks remain). The blue team not only preserved its configuration but also created some installations on the yellow’s territory.

"Game of War and Creation" ("egg creation" version) rules (see example in Fig. 7):

Rule 1: Any live cell from either team with fewer than two live teammate neighbors dies (becomes empty), as if caused by underpopulation.

Rule 2a: Any live cell from either team with two or three live teammate neighbors will continue living in the next generation if the number of adversary neighbors is not more than the number of teammate neighbors.

Rule 2b: Any live cell from either team with two or three live teammate neighbors dies (becomes empty) if the number of its adversary neighbors is more than the number of teammate neighbors, as if by "killing".

Rule 3: Any live cell from either team with more than three live teammate neighbors dies (becomes empty), as if by overpopulation.

Rule 4: Any dead (or empty) cell with exactly three live neighbors ("parents") from either team becomes a live cell and a teammate of its parents, as if by reproduction. If both teams pretend to get a "child" within the same cell, the advantage is given to team, which belongs to the same "country" as the empty cell (defined by the border).

Rule 5: Any dead (or empty) cell with exactly two live neighbors ("delayed parents") from either team becomes an "egg" cell (of the same color as its parents' team) with the "delay" parameter t initialized as $t = \boldsymbol{T}$. If both teams pretend to create their "egg" within the same cell, the advantage is given to team, which belongs to the same "country" as the empty cell (defined by the border).

Rule 6: Any "egg" cell at each iteration decrements its "delay" parameter by 1. If, as a result, t reaches zero, then the "egg" cell is replaced by the live cell (preserving its color or team affiliation), as if by delayed reproduction.

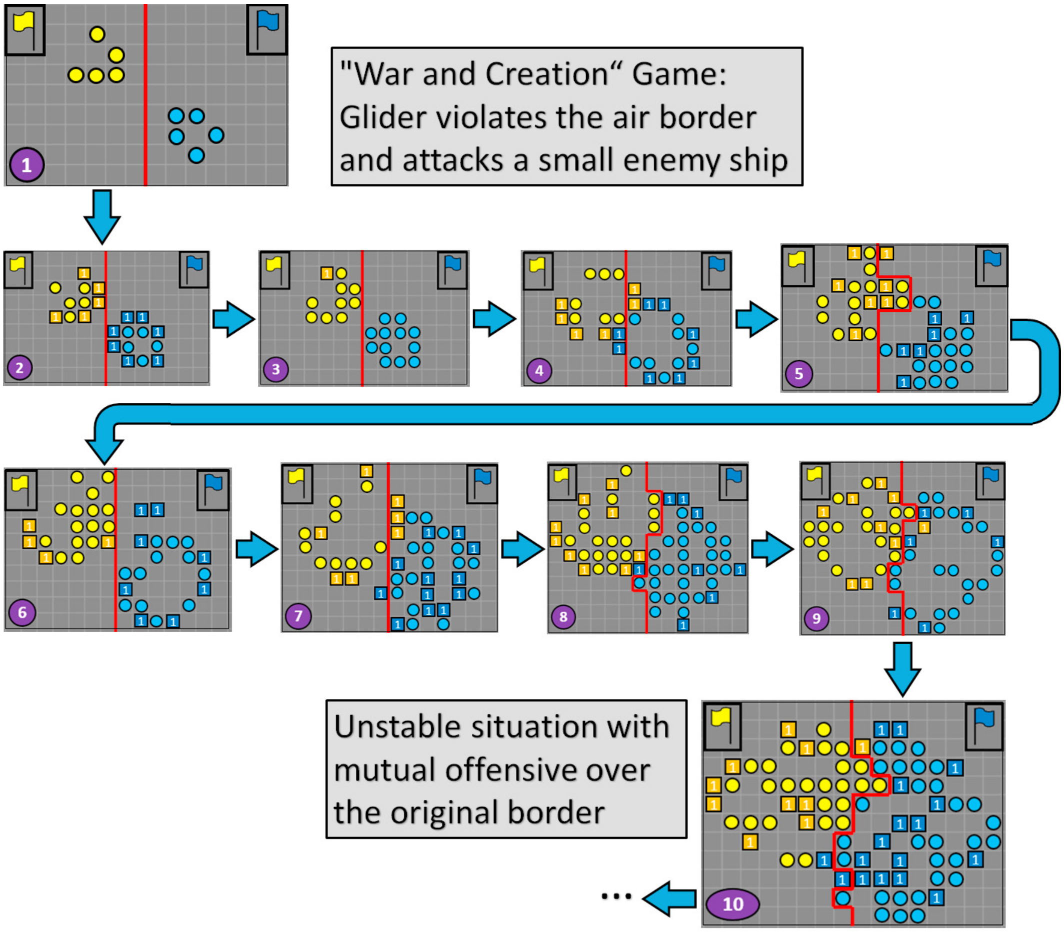

Fig. 7. The collision between the yellow "Glider" and the blue ship ("Boat"), which happened after the glider attempted to cross the border in the "War and Creation" ("egg creation" version) CA. The choice for a small delay parameter ($T = 1$) improved survivability of the life cells and provided additional dynamics to the CA evolution. The situation after ten iterations remains unstable and both teams seem to have certain progress in occupying the enemy's territory and good chances to continue their mutual offensive.

One may see that various CA population/configuration creativity, survivability and resilience capabilities largely depend on the rules and their parameters (e.g., number of "parents" for the "newborn child" or the "delayed born" parameter among many others). Manual control over the rules during the simulation process is quite a complicated and time-consuming task. The best way would be automatic adaptation of CA (particularly the rules behind CA) to the desired behavior. One possible approach has been suggested in [27] where the authors studied the use of genetic algorithms for this purpose. They show how the CA rules can be found as an outcome of genetic evolution so that these rules can enforce a desired behavior for the CA. Our preliminary experiments show that the efficiency impact of genetic algorithms used for adaptive (self-learning) options of our "Life and Creation", "War and Peace", and "War and Creation", i.e., CA with a greater simulation scope, is greater than in the case of traditional CA.

Yes, it is true that CA as a simulation tool and its modifications are a very high-level abstraction and applying it to real-life problems, particularly in industry, requires a certain effort. One reason for this difficulty is that real-life systems are often very complex, with many interacting components and processes that are difficult to model with simple rules. CA are based on simple local rules that determine the behavior of each cell, and it can be challenging to find the right set of rules to capture the complexity of a real-life system. Another challenge is that real-life systems often involve continuous processes, such as the flow of fluids or the transfer of heat, which are difficult to model using discrete cellular automata. Despite these challenges, it is still possible to use CA to model real-life phenomena, including industrial processes. A thorough combination of empirical data, theoretical models, and computational simulations will lead to CA models that capture the essential features of a system. In any case, the combination of CA advantages, such as simplicity, emergent behavior, nonlinear dynamics, parallelism, and adaptability, makes CA a powerful simulation instrument that can be used to model a wide range of complex Industry 5.0 systems that involve humans with different roles, smart infrastructure, their complex interaction and logistics, safety, and resilience.

4. Conclusions

Resilience is significant for sustaining operations in the context of Industry 5.0 due to its ability to navigate unforeseen disruptions, ranging from cyber-attacks to supply chain interruptions and natural disasters, which is pivotal for sustainable operations. As systems and processes become more complex and interconnected, the effects of disruptions spread widely, highlighting the importance of using advanced modeling methods. The Creative CA models, exemplified by "Life and Creation", lay the foundation for secure interactions among humans, technology, and products within organizational settings. Conversely, the Adversarial CA models, such as "War and Peace", hold the promise of optimizing systems to deftly detect, respond to, and recover from a spectrum of adversarial threats. The Hybrid CA model, including the innovative "War and Creation", emerges as a versatile structure that easily integrates aspects of collaboration, competition, destruction, and recovery. This hybridization introduces a novel dimension to the pursuit of resilience, enabling dynamic adaptations within Industry 5.0 contexts. Enriching this approach, the integration of evolutionary computation empowers the refinement of rules, enhancing infrastructure resilience and ensuring effective performance under various adversities.

We performed a preliminary experimental testing of these CA by simulating resilient and adaptive industrial logistics processes within the project IMMUNE: "Cyber-Defence for Intelligent Systems", which is NATO SPS project (http://recode.bg/natog5511) [28]. Due to the sensitive nature of our industrial partners' data, we are unable to provide specific case descriptions. However, we have validated our CA instrument on a variety of simulated scenarios, and the results demonstrate its potential for real-world applications. While grappling with real-world data complexities and privacy concerns, we acknowledge the necessity of quantifying our methods' efficacy. To this end, our planned future efforts are focused on creating and sharing synthetic benchmark datasets, capturing multifaceted dimensions of resilience within Industry 5.0 systems. We also envision integration of CA with agent-based paradigms or machine learning methods, which will potentially enhance the ability to model complex human-machine interactions in the dynamic environment of Industry 5.0. In this way, we want to bridge the core concept of resilience with the transformative realm of digital evolution in Industry 5.0 and potentially contribute to setting the trajectory towards a sustainable, resilient and efficient future of Industries 4.0, 5.0, and smart manufacturing.

References

[1] Nahavandi, S. (2019). "Industry 5.0—A human-centric solution". *Sustainability*, **11(16)**: 4371.

[2] Sindhwani, R., Afridi, S., Kumar, A., Banaitis, A., Luthra, S., and Singh, P. L. (2022). “Can industry 5.0 revolutionize the wave of resilience and social value creation? A multi-criteria framework to analyze enablers”. *Technology in Society*, **68**: 101887.

[3] Kari, J. (2005). “Theory of cellular automata: A survey.” *Theoretical Computer Science*, **334(1-3)**: 3-33. https://doi.org/10.1016/j.tcs.2004.11.021

[4] Wąs, J., and Sirakoulis, G. C. (2015). “Cellular automata applications for research and industry”. *Journal of Computational Science*, **11**: 223-225. https://doi.org/10.1016/j.jocs.2015.10.005

[5] Sarkar, P. (2000). “A brief history of cellular automata”. *ACM Computing Surveys*, 32(1): 80-107. https://doi.org/10.1145/349194.349202

[6] Wootton, J. T. (2001). “Local interactions predict large-scale pattern in empirically derived cellular automata”. *Nature*, **413(6858)**: 841-844. https://doi.org/10.1038/35101595

[7] Li, Z., Sim, C. H., and Low, M. Y. H. (2006). “A survey of emergent behavior and its impacts in agent-based systems”. In: *Proceedings of the 4th IEEE International Conference on Industrial Informatics* (pp. 1295-1300). IEEE. https://doi.org/10.1109/INDIN.2006.275846

[8] Folino, G., Giordano, A., and Mastroianni, C. (2016). “Scalable asynchronous execution of cellular automata”. In: *AIP Conference Proceedings* (Vol. 1776, No. 1, p. 080006). AIP Publishing LLC. https://doi.org/10.1063/1.4965363

[9] Giordano, A., De Rango, A., D’Ambrosio, D., Rongo, R., and Spataro, W. (2019). “Strategies for parallel execution of cellular automata in distributed memory architectures”. In: *Proceedings of the 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing* (pp. 406-413). IEEE. https://doi.org/10.1109/EMPDP.2019.8671639

[10] Sen, B., Goswami, M., Mazumdar, S., and Sikdar, B. K. (2015). “Towards modular design of reliable quantum-dot cellular automata logic circuit using multiplexers”. *Computers & Electrical Engineering*, **45**: 42-54. https://doi.org/10.1016/j.compeleceng.2015.05.001

[11] Fu, S. C., and Milne, G. (2004). “A flexible automata model for disease simulation”. In: *Proceedings of the 6th International Conference on Cellular Automata for Research and Industry* (pp. 642-649). Springer. https://doi.org/10.1007/978-3-540-30479-1_66

[12] Raabe, D. (2002). “Cellular automata in materials science with particular reference to recrystallization simulation”. *Annual Review of Materials Research*, **32(1)**: 53-76. https://doi.org/10.1146/annurev.matsci.32.090601.152855

[13] Elbasheer, M., Longo, F., Nicoletti, L., Padovano, A., Solina, V., and Vetrano, M. (2022). “Applications of ML/AI for decision-intensive tasks in production planning and control”. *Procedia Computer Science*, **200**: 1903-1912. https://doi.org/10.1016/j.procs.2022.01.391

[14] Yan, W., and Feng, X. (2011). “Simulation of strategic decision for supply chain on Cellular Automata”. In: *Proceedings of the International Conference on Computer Science and Service System* (pp. 731-735). IEEE. https://doi.org/10.1109/CSSS.2011.5972180

[15] Chen, K. S., Wu, S. L., and Yeh, H. M. (2015). “Modeling of chemical mechanical polishing processes by cellular automata and finite element/matlab integration methods”. *Microsystem Technologies*, **21**: 1879-1892. https://doi.org/10.1007/s00542-014-2302-9

[16] Yasin, J. N., Mahboob, H., Haghbayan, M. H., Yasin, M. M., and Plosila, J. (2022). “Cellular formation maintenance and collision avoidance using centroid-based point set registration in a swarm of drones”. In: *Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference* (Vol. 1, pp. 391-408). Springer. https://doi.org/10.1007/978-3-030-82193-7_26

[17] Mitra, A. (2022). “On boundary-effects at cellular automata-based road-traffic model towards uses in smart city”. In: *Proceedings of International Conference on Intelligent and Cloud Computing* (pp. 111-120). Springer Nature Singapore. https://doi.org/10.1007/978-981-16-9873-6_10

[18] Cook, M. (2004). “Universality in elementary cellular automata”. *Complex Systems*, **15(1)**: 1-40. https://doi.org/10.25088/ComplexSystems.15.1.1

[19] Gordon, D. (1987). “On the computational power of totalistic cellular automata”. *Mathematical Systems Theory*, **20**: 43-52. https://doi.org/10.1007/BF01692058

[20] Arrighi, P., Schabanel, N., and Theyssier, G. (2013). “Stochastic cellular automata: Correlations, decidability and simulations”. *Fundamenta Informaticae*, **126(2-3)**: 121-156. https://doi.org/10.3233/FI-2013-875

[21] Seck-Tuoh-Mora, J. C., Hernandez-Romero, N., Lagos-Eulogio, P., Medina-Marin, J., and Zuñiga-Peña, N. S. (2021). “A continuous-state cellular automata algorithm for global optimization”. *Expert Systems with Applications*, **177**: 114930. https://doi.org/10.1016/j.eswa.2021.114930

[22] Akundi, A., Euresti, D., Luna, S., Ankobiah, W., Lopes, A., and Edinbarough, I. (2022). “State of Industry 5.0—Analysis and identification of current research trends”. *Applied System Innovation*, **5(1)**: 27. https://doi.org/10.3390/asi5010027

[23] Adel, A. (2022). “Future of industry 5.0 in society: human-centric solutions, challenges and prospective research areas”. *Journal of Cloud Computing*, **11(1)**: 1-15. https://doi.org/10.1186/s13677-022-00314-5

[24] Golovianko, M., Terziyan, V., Branytskyi, V., & Malyk, D. (2023). ”Industry 4.0 vs. Industry 5.0: Co-existence, transition, or a hybrid”. *Procedia Computer Science*, **217**: 102-113. https://doi.org/10.1016/j.procs.2022.12.206

[25] Longo, F., Nicoletti, L., and Padovano, A. (2019). “Modeling workers’ behavior: A human factors taxonomy and a fuzzy analysis in the case of industrial accidents”. *International Journal of Industrial Ergonomics*, **69**: 29-47. https://doi.org/10.1016/j.ergon.2018.09.002

[26] Gardner, M. (1970). “Mathematical games”. *Scientific American*, **222(6)**: 132-140. https://www.jstor.org/stable/24925832

[27] Bäck, T., and Breukelaar, R. (2005). “Using genetic algorithms to evolve behavior in cellular automata”. In: *Unconventional Computation. Lecture Notes in Computer Science* (vol. 3699, pp. 1-10). Springer. https://doi.org/10.1007/11560319_1

[28] Kaikova, O., Terziyan, V., Tiihonen, T., Golovianko, M., Gryshko, S., and Titova, L. (2022). ”Hybrid threats against Industry 4.0: adversarial training of resilience”. *E3S Web of Conferences,* **353**: 03004. https://doi.org/10.1051/e3sconf/202235303004