

mgr inż. Mariusz Dobrowolski^{a)}, dr hab. inż. Tadeusz Maciak, prof. PB^{a)*}

^{a)}Politechnika Białostocka / Białostok University of Technology

*Autor korespondencyjny / Corresponding author: t.maciak@pb.edu.pl

Oprogramowanie CrowdSim do modelowania dynamiki pieszych

CrowdSim – Pedestrian Dynamics Modeling Software

Программное обеспечение CrowdSim для моделирования пешеходных потоков

ABSTRAKT

Cel: Celem artykułu jest zaprezentowanie autorskiego oprogramowania **CrowdSim** służącego do modelowania dynamiki pieszych. W oprogramowaniu zaimplementowano algorytmy bazujące na dwuwymiarowym modelu automatu komórkowego Cellular Automata (**CA**). Uwzględniono w nich strategiczne możliwości pieszych dotyczące wyboru wyjścia (**model CA-A**) oraz możliwości powstawania zakłóceń płynności ruchu i blokad (**model CA-B**). Zaimplementowanie różnych modeli pod względem złożoności dało możliwości porównania ich efektywności, funkcjonowania oraz stopnia realizmu otrzymanych wyników.

Projekty i metody: W celu lepszego i bardziej wiarygodnego wyznaczenia ścieżek poruszania się pieszych zaproponowano dwie modyfikacje usprawniające działanie algorytmów symulacyjnych. Pierwszą z nich było wprowadzenie zaawansowanych algorytmów rankujących komórki: **SRA** (Simple Ranking Algorithm), **BFS-RA** (Breadth First Search Ranking Algorithm) i **ARA** (Advanced Ranking Algorithm). Wartość rankingu danej komórki w procesie symulacji jest dla każdego z aktorów wyznacznikiem drogi, po której się porusza. Kolejnym autorskim usprawnieniem było wprowadzenie do definicji automatu komórkowego pojęcia **pamięci aktora**, dzięki czemu piesi podczas symulacji, pamiętając swoje ostatnie ruchy, poruszają się bardziej płynnie i realistycznie.

Wyniki: Na potrzeby porównania modeli przygotowana została specjalna mapa budynku o wymiarach kontyngencji 40 na 22,5 metra (900 m²) zdolnego pomieścić maksymalnie 3000 osób. Przeprowadzone symulacje wykazały, że w przypadku modelu **CA** aktorzy gromadzą się przy jednym z wyjść, podczas gdy pozostałe zostają w większości nieużywane. Wynika to z faktu, że w modelu **CA** brakuje funkcji kosztu, zatem aktor zawsze wybiera wyjście najbliższe. Inaczej sprawa ma się w przypadku modelu **CA-A**. Tu aktorzy w miarę równo obciążają każde z wyjść; są oni w stanie podjąć decyzje odnośnie do wyboru wyjścia nie tylko na podstawie odległości, ale również zagęszczenia osób przy danym wyjściu. W przypadku porównania modeli **CA-A** i **CA-B** praktycznie niezauważalne są różnice w otrzymanych statystykach obciążenia wyjść. Dopiero po bliższym przyjrzeniu się ruchowi aktorów podczas symulacji widać, że jeśli w przypadku czystego modelu **CA-A** ludzie poruszają według idealnego porządku, o tyle w modelu **CA-B** zauważalne są chwilowe blokady i zakleszczenia aktorów.

Wnioski: Stworzone i zaprezentowane w niniejszej pracy oprogramowanie **CrowdSim** może posłużyć zarówno projektantom, architektom lub konstruktorom, jak i służbom porządkowym, ochronie danego obiektu lub organizatorom imprez masowych. Symulacje wykonane wielokrotnie dla jednego scenariusza przy różnych parametrach wejściowych uwiadczyły problemy i wąskie gardła mogące pojawić się w trakcie ewakuacji z zagrożonego obiektu. Przeprowadzanie takich symulacji na etapie projektowym może bardzo pozytywnie wpłynąć na bezpieczeństwo osób użytkujących dany obiekt, jak również poprawić jego funkcjonalność.

Słowa kluczowe: ewakuacja, modelowanie, automat komórkowy

Typ artykułu: oryginalny artykuł naukowy

Przyjęty: 14.12.2018; Zrecenzowany: 19.12.2018; Zatwierdzony: 21.12.2018;

Identyfikatory ORCID autorów: M. Dobrowolski – 0000-0001-6496-0789; T. Maciak – 000-0001-9058-1272;

Autorzy wnieśli równy wkład merytoryczny w opracowanie artykułu;

Proszę cytować: BiTP Vol. 52 Issue 4, 2018, pp. 46–66, <https://dx.doi.org/10.12845/bitp.52.4.2018.3>;

Artykuł udostępniany na licencji CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>).

ABSTRACT

Aim: The aim of the article is to present **CrowdSim**, proprietary software for modeling the dynamics of pedestrians. The software offers algorithms based on a 2-dimensional model of a Cellular Automaton (**CA**). They include strategic pedestrian choices of the exit (**CA-A model**) and the formation of blockages (**Model CA-B**). Implementation of various models in terms of their complexity provided the opportunity to compare their efficiency, functioning and realism of the results obtained.

Project and methods: In order to better and more credibly determine pedestrian paths, two modifications were proposed to improve the operation of simulation algorithms. The first of these was the introduction of advanced algorithms ranking cells: **SRA** (Simple Ranking Algorithm), **BFS-RA** (Breadth First Search Ranking Algorithm) and **ARA** (Advanced Ranking Algorithm). The value of the ranking of a given cell in the simulation process is for each

of the actors the determinant of the path on which it moves. Another proprietary improvement was the introduction of the concept of *actor's memory* to the definition of a cellular automata, thanks to which pedestrians, while remembering their last movements, move more smoothly and realistically.

Results: For the purposes of model comparison, a special map of the building with the dimensions of 40 to 22.5 meters (900 m²) has been prepared, able to accommodate a maximum of 3000 people. The simulations showed that in the case of the **CA** model, the actors gather at one of the outputs while the others are mostly unused. This is due to the fact that the **CA** model lacks the cost function, so the actor always chooses the nearest output. Things are different in the case of the **CA-A** model. Here, the actors equally burden each exit, they are able to make decisions regarding the choice of the exit not only on the basis of the distance but also the concentration of actors at the given exit. When comparing the **CA-A** and **CA-B** models, there are practically no differences in the received output load statistics. Only after a closer look at the movement of the actors during the simulation can be seen that if in the case of a clean **CA-A** model people move according to the ideal order, in the **CA-B** model there are momentary blockages and deadlocks of actors.

Conclusions: **CrowdSim** software created and presented in this work can be used by designers, architects or constructors as well as law enforcement, protection of a given facility or organizers of mass events. Simulations made repeatedly for one scenario with different input parameters reveal problems and bottlenecks that may appear during evacuation from an object at risk. Conducting such simulations at the design stage can have a very positive impact on the safety of people using a given facility, as well as improve its functionality.

Keywords: evacuation, modelling, cellular automata

Type of article: original scientific article

Received: 14.12.2018; Reviewed: 19.12.2018; Accepted: 21.12.2018;

Authors' ORCID IDs: M. Dobrowolski – 0000-0001-6496-0789; T. Maciak – 000-0001-9058-1272;

The authors contributed equally to this article;

Please cite as: *BiTP* Vol. 52 Issue 4, 2018, pp. 46–66, <https://dx.doi.org/10.12845/bitp.52.4.2018.3>;

This is an open access article under the CC BY-SA 4.0 license (<https://creativecommons.org/licenses/by-sa/4.0/>).

АННОТАЦИЯ

Цель: Цель статьи – представить проприетарное программное обеспечение **CrowdSim** для моделирования пешеходных потоков. Это программное обеспечение имеет алгоритмы, основанные на двумерной модели клеточного автомата Cellular Automata (**CA**). Они включают в себя стратегические возможности пешеходов, относительно выбора выхода (**модель CA-A**), а также возможности возникновения препятствий и заторов на пути движения потока (**модель CA-B**). Внедрение различных моделей с точки зрения сложности предоставило возможность сравнить их эффективность, функционирование и реалистичность полученных результатов.

Проект и методы: Чтобы лучше и достовернее определить пешеходные маршруты, были предложены две модификации для улучшения работы алгоритмов моделирования. Первой из них было внедрение продвинутых алгоритмов, которые ранжируют клетки: **SRA** (Simple Ranking Algorithm), **BFS-RA** (Breadth First Search Ranking Algorithm) и **ARA** (Advanced Ranking Algorithm). Значение ранжирования данной клетки в процессе моделирования для каждого из участников является определяющим фактором пути, по которому он движется. Другим проприетарным усовершенствованием стало введение концепции памяти актера в определение клеточного автомата, благодаря которому пешеходы, помня последние движения, двигались более плавно и реалистично.

Результаты: Для целей сравнения моделей была подготовлена специальная карта здания размером от 40 до 22,5 метров (900 м²), рассчитанная на максимум 3000 человек. Моделирование показало, что в случае модели **CA** актеры собираются на одном из выходов, в то время как другие выходы, в основном, не используются. Это связано с тем, что в модели **CA** отсутствует функция стоимости, поэтому актер всегда выбирает ближайший выход. В случае с моделью **CA-A** ситуация совершенно иная. Здесь актеры одинаково пользуются каждым выходом, они могут принимать решения относительно выбора выхода не только исходя из расстояния, но и плотности людей на данном выходе. В случае сравнения моделей **CA-A** и **CA-B** практически нет различий в полученной статистике нагрузки на выход. Только после более пристального изучения движения актеров во время симуляции можно заметить, что если в случае чистой модели **CA-A** люди движутся в соответствии с идеальным порядком, то в модели **CA-B** присутствуют моментные блокировки и зажатия актеров.

Выводы: Программное обеспечение **CrowdSim**, созданное и представленное в этой работе, может использоваться дизайнерами, архитекторами или конструкторами, а также правоохранительными органами, службой охраны данного объекта или организаторами массовых мероприятий. Моделирование, сделанное неоднократно для одного сценария с различными входными параметрами, выявляет проблемы и пробки, которые могут возникать при эвакуации с объекта, подверженного риску. Проведение таких симуляций на этапе проектирования может оказать очень положительное влияние на безопасность людей, использующих данный объект, а также улучшить его функциональность.

Ключевые слова: эвакуация, моделирование, клеточный автомат

Вид статьи: оригинальная научная статья

Принята: 14.12.2018; Рецензирована: 19.12.2018; Одобрена: 21.12.2018;

Идентификаторы ORCID авторов: M. Dobrowolski – 0000-0001-6496-0789; T. Maciak – 000-0001-9058-1272;

Авторы внесли одинаковый вклад в создание этой статьи;

Просим ссылаться на статью следующим образом: *BiTP* Vol. 52 Issue 4, 2018, pp. 46–66, <https://dx.doi.org/10.12845/bitp.52.4.2018.3>;

Настоящая статья находится в открытом доступе и распространяется в соответствии с лицензией CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>).

Wstęp

Bardzo ważną metodą badania procesu ewakuacji osób z zagrożonych budynków jest konstrukcja odpowiednich modeli matematycznych opisujących przemieszczanie się pieszych. Symulacje komputerowe oparte na takich modelach umożliwiają

Introduction

A very important method of analyzing the evacuation of people from buildings at risk is developing appropriate mathematical models describing relocation of pedestrians. Computer simulations based on such models enable to define many

określenie wielu ważnych parametrów ewakuacji budynków o dowolnych wnętrzach (takich jak np. całkowity czas ewakuacji), a także wizualizacje całego procesu ewakuacji. Dane zebrane podczas tego typu symulacji są doskonałym źródłem informacji dla architektów i projektantów, którzy mogą zawczasu zweryfikować proponowane przez siebie rozwiązania pod kątem bezpieczeństwa. Symulacje komputerowe są również doskonałą alternatywą dla próbnych ewakuacji, które można przeprowadzić dopiero po oddaniu budynku do użytku. Problematyka zapewnienia sprawnego procesu ewakuacji jest na tyle ważna, że odbywają się w tym obszarze wiedzy cykliczne międzynarodowe konferencje. Przykładowo można tu wymienić konferencje *Pedestrian and Evacuation Dynamics* [1], *Crowd and Cellular Automata* [2] czy *Traffic and Granular Flow* [3].

Na rynku dostępnych jest kilka komercyjnych rozwiązań oprogramowania mogącego posłużyć do symulacji procesu ewakuacji ludzi. Zdecydowana większość z tych rozwiązań jest płatna. Można tu wymienić najważniejsze i najbardziej popularne: **MassMotion** [4], **Pathfinder** [5], **Exodus** [6], **Legion** [7], **Fire Dynamic Simulator (FDS) + Evac** [8], **Goalem Crowd** [9], **Massive** [10], **Miarmy** [11], **Quadstone Paramics** [12], **VISSIM** [13]. Wymienione aplikacje umożliwiają tworzenie map w pełni trójwymiarowych oraz możliwość importu oraz eksportu najpopularniejszych programów do obróbki grafiki 3D (np. 3DS Max). Większość z nich charakteryzuje się zaawansowanym modułem analitycznym, mogącym przedstawić kilkanaście różnych statystyk.

W artykule przedstawiono autorskie oprogramowanie **CrowdSim** służące do modelowania ruchu pieszych. Bazuje ono na znanych algorytmach ruchu jednostek wywodzących się z dwuwymiarowego modelu automatów komórkowych (model Cellular Automata **CA**) [14]. Poprzez modyfikacje bazowego modelu w oprogramowaniu uwzględniono strategiczne możliwości pieszych dot. wyboru wyjścia (model **CA-A**), jak też możliwości powstawania zakłóceń płynności ruchu oraz blokad (model **CA-B**) [15, 16]. Do przedstawionych modeli wprowadzono autorską modyfikację polegającą na tym, że pieszy zamiast losowego wyboru kierunku drogi przy przemieszczaniu się z komórki do komórki zachowuje swój poprzedni wektor ruchu. Wybór drogi pieszego odbywa się przez zastosowanie zaawansowanych algorytmów rankingowania komórek. Zaprezentowano również narzędzia umożliwiające zbieranie i wizualizacje danych obliczonych podczas procesu symulacji. Porównano wyniki zaproponowanych modeli dynamiki pieszych co pozwoliło na wskazanie mocnych i słabych stron każdego z nich oraz na wybór najbardziej optymalnych rozwiązań w zależności od założonych parametrów wejściowych.

Algorytmy ruchu pieszych wykorzystane w oprogramowaniu

Bazowy model Cellular Automata CA

Jak już wspomniano we wstępie w pracy wykorzystano algorytmy ruchu pieszych bazujące na dwuwymiarowym modelu automatów komórkowych **CA**. Rozpatruje się tu kwadratową siatkę komórek, które mogą przyjmować stan „pusty” bądź

essential parameters of evacuation of buildings of any interiors (such as, e.g. total evacuation time) and also to visualize the whole evacuation process. Data collected during this type of simulation are a perfect source of information for architects and designers who can then verify their solutions in terms of safety beforehand. Computer simulations are also a good alternative for trial evacuations which can be performed only after the building has been commissioned for use. The efficiency of an evacuation process and the related subject matter are significant to the point that international conferences are regularly held (*Pedestrian and Evacuation Dynamics* [1], *Crowd and Cellular Automata* [2] or *Traffic and Granular Flow* [3], etc.) to discuss these issues.

The market has several commercial software solutions available for the simulation of the evacuation process. The vast majority of these solutions is paid. The most essential and popular are: **MassMotion** [4], **Pathfinder** [5], **Exodus** [6], **Legion** [7], **Fire Dynamic Simulator (FDS) + Evac** [8], **Goalem Crowd** [9], **Massive** [10], **Miarmy** [11], **Quadstone Paramics** [12], **VISSIM** [13]. These applications allow one to create fully three-dimensional maps and to import to and export from the most popular programs for 3D graphical processing (e.g. 3DS Max). The majority of them is characterized by advanced analytical modules to browse through various statistical data.

The article presents proprietary software **CrowdSim** for modelling pedestrian traffic. It is based on the known algorithms of movement of units originating from a two-dimensional cellular automata model (**CA**). [14]. Through the modification of the base model, strategic possibilities of pedestrians concerning selection of exits (**CA-A** model) and the possible occurrence of disruptions to traffic smoothness and blockages (**CA-B** model) were taken into account in the software [15, 16]. The presented models featured a proprietary modification which made the pedestrian not choose directions randomly while moving from cell to cell, but follow the same movement vector. Pedestrian's path is selected through the advanced algorithms of ranking cells. Also tools allowing the collection and visualisation of data computed during the simulation process were presented. The results of the proposed pedestrian dynamics models were compared, which revealed the strengths and weaknesses of each of them and allowed us to select the most optimal solutions depending on the adopted input parameters.

Pedestrian traffic algorithms used in the software

Basic Cellular Automata CA model

As mentioned in the Introduction, the article discusses algorithms concerning pedestrian traffic basing on a two-dimensional model of cellular automata **CA**. A square mesh of cells is used here which can be “empty” or “occupied”. Mo-

„zajęty”. W rozważaniach zakłada się sąsiedztwo Moore’a [17]. Pojedyncza komórka reprezentować może kilka wartości: pieszych, przeszkody, wyjścia lub przestrzeń ruchu. Przestrzeń ruchu oznacza komórki siatki dostępne dla ruchu pieszych. Na początku symulacji na siatce znajdują się tylko komórki reprezentujące ściany (przeszkody), wyjścia oraz przestrzeń ruchu. W kolejnych krokach generowana jest populacja pieszych w miejscach niezajętych przez przeszkody i wyjścia. Ruch pieszych rejestrowany jest w dyskretnych przedziałach czasowych (np. odpowiadających 1 s). W kolejnych krokach symulacji piesi, według wybranego algorytmu, kierują się ku wyjściom.

Aby piesi mogli poruszać się po siatce komórek, muszą być one odpowiednio rankingowane. Najprostszą metodą rankingowania pól jest tzw. pole potencjalne Bouviera [18]. Koncepcja ta polega na tym, że do każdej z komórek siatki przypisuje się pewną wartość liczbowa zwaną potencjałem, uzależnioną od odległości od najbliższej komórki będącej wyjściem z pomieszczenia. Kiedy przypisze się tej reprezentującej wyjście komórce wartość 0, to wszystkie komórki przyległe (otoczenie Moore’a promień $R = 1$) będą miały przypisaną wartość potencjału równą 1. Analogicznie komórki należące do otoczenia Moore’a o promieniu $R = 2$ będą miały przypisaną wartość potencjału równą 2 itd. W ten sposób pole potencjalne propagowane jest od każdego z wyjść na całą siatkę automatu. Na podstawie wartości pola pieszy wybiera najbliższe mu wyjście i do niego podąża. W przypadku rozpatrywania propagacji pola potencjalnego od dwóch lub więcej komórek reprezentujących wyjście, przyjętą wartością dla danej komórki jest minimalna wartość potencjału (czyli odległość od najbliższego wyjścia w rozpatrywanej przestrzeni metrycznej). Natomiast w przypadku bardziej skomplikowanych modeli piesi mogą w trakcie symulacji zmieniać swoje preferencje, co do wyjść, do których podążają. Wówczas każda z komórek zawiera listę potencjałów. Na tej liście oddzielnie znajdują się wartości potencjałów dla każdego z dostępnych wyjść. Koncepcja Bouviera ma jednak swoje wady. Ten prosty mechanizm nienajlepiej sprawdza się w skomplikowanych pomieszczeniach, w których znajduje się wiele przeszkód jak np. ścianki działowe. Dlatego też, aby jak najwierniej odwzorować proces ewakuacji w stworzonym na potrzeby tej pracy programie komputerowym, zastosowano algorytm *Breadth-First Search (BFS)* [19]. Punktem wyjściowym są komórki oznaczone na siatce jako wyjścia z priorytetem 0, natomiast wierzchołkami są wszystkie komórki reprezentujące przestrzeń ruchu. Połączenia między wierzchołkami określane są poprzez sąsiedztwo Moore’a. Kolejność odwiedzania wierzchołków określa potencjał danych komórek. Algorytm ten komplikuje się trochę bardziej, jeżeli weźmiemy pod uwagę całe budynki, w których znajduje się wiele pomieszczeń, a oprócz drzwi wyjściowych znajdują się również drzwi przejściowe między tymi pomieszczeniami. Wówczas każde z tych pomieszczeń będzie posiadać swój lokalny ranking pól względem wyjść wewnętrznych, uwzględniający jednak bliskość do drzwi ewakuacyjnych.

W podstawowym modelu CA przyjmuje się losowe zajmowanie komórek przez pieszych. Z algorytmem ruchu pieszych bazującym na klasycznym modelu CA można zapoznać się np. w pracach H.L. Klüpfela [15], J. Wąsa [16] i M. Dobrowolskiego [20].

ore neighborhood is assumed in considerations [17]. A single cell can represent several values: pedestrians, obstacles, exits or traffic area. Traffic area means cells of the mesh available for pedestrian traffic. At the beginning of the simulation, the mesh only includes cells representing walls (obstacles), exits and traffic area. In the subsequent steps, a population of pedestrians is generated in places which are not occupied by obstacles and exits. Pedestrian traffic is recorded in discrete time intervals (e.g. of 1 s). In the further steps of the simulation, pedestrians move towards exits according to the selected algorithm.

In order for the pedestrians to move on the cellular mesh, the cells must be appropriately ranked. The simplest way of field ranking is by means of Bouvier’s potential field [18], a concept in which every cell of the mesh is assigned a number (the potential) the value of which depends on the distance from the nearest cell being the exit from the room. When the cell representing the exit is assigned value 0, all adjacent cells (Moore neighbourhood radius $R = 1$) will have the assigned potential value of 1; cells included in the Moore neighbourhood with radius $R = 2$ will be assigned the potential value of 2, and so forth. In this way, a potential field is propagated to every exit across the entire automaton mesh. Basing on the field value, the pedestrian chooses the nearest exit and moves towards it. In the case of analysing propagation of the potential field from two or more cells representing the exit, the assumed value for a given cell is constituted by the minimum potential value (i.e. distance from the nearest exit within the analysed metric area). In more complicated models pedestrians can change their preferences as to which exits they will move towards during the simulations. In such a case, each cell includes a list of potentials. This list in turn includes values of potentials for each of the available exits separately. Bouvier’s concept, however, has some disadvantages. Its simple mechanism proves rather insufficient in complex rooms with many obstacles, e.g. partition walls. Therefore, in order to accurately capture the evacuation process in the software developed for the purposes of this article, the *Breadth-First Search (BFS)* algorithm was used [19]. The starting point is constituted by the cells marked on the mesh as exits with priority 0, whereas peaks are all cells representing the traffic area. Connections between the peaks are defined based on the Moore neighborhood. The sequence in which peaks are visited is defined by the potential of particular cells. This algorithm becomes slightly more complex if we take whole buildings into account, with many rooms and passage doors, in addition to exit doors. In such case, each room will have its local ranking of fields in relation to internal exits, but accounting for close distance to evacuation doors.

In the basic CA model, cells are assumed to be occupied by the pedestrians on a random basis. The pedestrian traffic algorithm based on a classic CA model is described in papers by authors such as H.L. Klüpfel [15], J. Wąs [16] and M. Dobrowolski [20].

Modyfikacje modelu podstawowego

W modyfikacji modelu podstawowego proces decyzyjny został poszerzony o informację dotyczącą otoczenia. W przypadku grupy ludzi opuszczającej pomieszczenie, w którym znajduje się więcej niż jedno wyjście, poszczególne osoby podejmują decyzję o skierowaniu się do najkorzystniejszego z ich punktu widzenia wyjścia. W procesie podejmowania decyzji badana jest odległość od poszczególnych wyjść, liczona jako wartość pola potencjalnego generowanego od każdego wyjścia oraz gęstość tłumu wokół wyjścia wyrażana jako liczba osób w określonym polu wyjścia. Tak więc w stosunku do wariantu podstawowego, gdzie generowane jest jedno wspólne pole potencjalne dla wszystkich wyjść, tu dla każdego wyjścia generowane jest osobne pole potencjalne, czyli dla każdej komórki siatki rozpatrujemy pewien wektor wartości potencjałów (model CA-A).

Model klasyczny CA zakłada, że jeżeli w otoczeniu danej komórki znajduje się więcej niż jedna komórka o niższym priorytecie, to wybór między tymi komórkami jest całkowicie losowy. Podejście takie powoduje, że pieszy biorący udział w symulacji nienaturalnie często zmienia kierunek ruchu, przez co jego droga jest bardzo nieregularna. Dlatego też do rozpatrywanego algorytmu jako nowy element wprowadzono pojęcie **pamięci aktora**. Każdy z pieszych zapisuje w pamięci wektor ruchu dla swojego ostatniego kroku. Pieszy, podejmując decyzje o wyborze kolejnej wolnej komórki, w pierwszej kolejności bierze pod uwagę komórkę zgodną z ostatnio zapamiętanym wektorem ruchu. Dzięki tej modyfikacji ruch pieszego jest znacznie bardziej płynny i przewidywalny, co dodatnio wpływa na realizm całej symulacji.

Schemat blokowy zmodyfikowanego w powyższy sposób algorytmu uwzględniającego możliwości wyboru wyjścia przez pieszego jest przedstawiony w pracy [20].

Opisywane wcześniej oba modele zakładają płynny przepływ mas ludzi zgodnie z ich uporządkowaną kolejnością względem odległości od wyjścia. Jednak w realnych sytuacjach ewakuacji zachowanie takiej kolejności bywa często zachwiane. Zwłaszcza podczas zaistnienia zjawiska paniki zaobserwować można zaistniałe między pieszymi konflikty, blokady oraz wzajemne przepychanie się [15]. Wobec tego należało zmodyfikować dalej istniejący algorytm o możliwość lokalizacji miejsc mogących spowodować tworzenie się blokad w ruchu pieszych. Można założyć, że iloraz liczby komórek wyjściowych do liczby wejściowych wskazuje dokładnie na miarę przepustowości danej komórki. W przypadku, gdy iloraz ten jest mniejszy od 1, mamy do czynienia z wąskim gardłem. W ten sposób wyznaczone są na siatce miejsca szczególnie narażone na blokady. Zatem im mniejszy współczynnik przepustowości, tym większe prawdopodobieństwo wystąpienia w tym miejscu blokady. Tak jak w przypadku poprzedniego modelu algorytm ruchu zakłada tu, że pieszy na podstawie funkcji kosztu dokonuje wyboru najbardziej odpowiadającego mu w danym momencie wyjścia z pomieszczenia. Jeżeli wyznaczona zostaje kolejna komórka jako cel ruchu w danym kroku czasowym, to algorytm sprawdza, czy należy ona do zbioru wąskich gardeł. Jeżeli tak jest to z wyznaczonym prawdopodobieństwem ustalone jest, czy wystąpiła blokada. Algorytm przewiduje w takim przypadku, że pieszy, który chce zająć taką komórkę, traci możliwość ruchu w tym kroku czasowym. Natomiast komórka ta zostaje zablokowana aż do końca

Modifications of the basic model

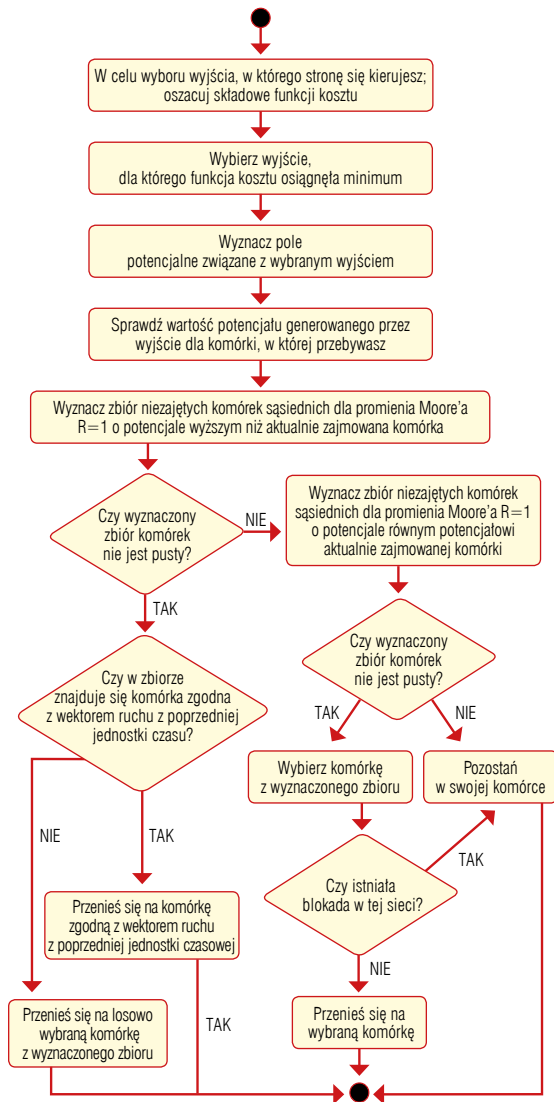
In the modified version of the basic model, the decision-making process was expanded to include information on the surrounding. In the case of a group of people leaving a room with more than one exit, individual persons decide on moving towards that exit which is the most favourable from their point of view. In the decision-making process, distances from particular exits are analysed, calculated as the value of a potential field generated from each exit and density of the crowd around the exit expressed as the number of persons within the defined exit field. Therefore, in relation to the basic option where one potential field is generated for all the exits, here, a separate potential field is generated for each exit, i.e. a certain vector of potential value is analysed for each cell in the mesh (CA-A model).

The classic CA model assumes that if there is more than one cell of lower priority around a particular cell, selection between these cells is totally random. Such an approach results in the fact that the pedestrian taking part in the simulation in an unnatural manner often changes the direction of movement, and therefore their path is highly irregular. Consequently, a new element, the concept of **actor's memory** concept was introduced to the algorithm in question. Each pedestrian records the movement vector for his last step in the memory. By making a decision concerning selection of the next free cell, the pedestrian first takes into account the cell compliant with recently remembered movement vector. Due to this modification, pedestrian's movement is considerably smoother and predictable, what positively affects realism of the whole simulation.

Flow chart of the such modified algorithm considering options of selection of the exit by the pedestrian is shown in paper [20].

The two models described earlier assume a smooth flow of masses of people in an orderly sequence in relation to the distance from the exit. However, in real-life evacuation situations, such a sequence is often disturbed. Especially during panic, conflicts between pedestrians, blockages and jostling can be observed [15]. Accordingly, the algorithm had to be modified even further by adding the possibility of locating sites which can cause blockages in pedestrian traffic. It can be assumed that the quotient of the number of exit cells to the number of entry cells precisely indicates the measure of the cell's throughput. If this quotient drops below 1, a bottleneck occurs. In this way, sites particularly exposed to blockages can be located on the mesh. The lower the throughput coefficient, the higher the probability of blockage at this specific spot. Similarly as with the previous model, the movement algorithm assumes that the pedestrian, basing on the cost function, chooses that exit which at the moment is most suitable for him. If another cell is selected as the destination of movement within a given time step, the algorithm verifies whether it belongs to the bottleneck group. If it does, then, subject to the assumed probability coefficient, it is checked whether a blockage has occurred. In such a case, the algorithm envisages that the pedestrian who wants to occupy such a cell, loses the possibility of movement during this time step. The cell, in turn, is blocked until the round ends. The pedestrian may move no sooner than during the next time step, and the blockage is removed from the cell constituting a bottleneck

ture. Pieszy zyskuje możliwość ruchu dopiero w następnym kroku czasowym, a z komórki będącej wąskim gardłem zdejmowana jest blokada (model CA-B). Schemat omawianego algorytmu zaimplementowanego w stworzonym oprogramowaniu został zaprezentowany na rycinie 1.



Rycina 1. Diagram opisujący algorytm ruchu pojedynczego pieszego w modelu uwzględniającym możliwość pojawienia się wąskich gardeł
Źródło: Opracowanie własne.

CROWDSIM

Oprogramowanie symulacyjne CrowdSim składa się z modułu symulacyjnego, wizualizacji oraz edytora map dwuwymiarowych. Daje ono możliwość przygotowania i przeprowadzenia symulacji ewakuacji osób z dość skomplikowanych pomieszczeń.

Edytor map

Pierwszym elementem, z którym styka się użytkownik oprogramowania, jest edytor map dwuwymiarowych. Okno edytora map zostało przedstawione na rycinie 2. Edytor pozwala zarów-

(the CA-B model). A diagram of the discussed algorithm, as implemented in the developed software, is presented in Figure 1.

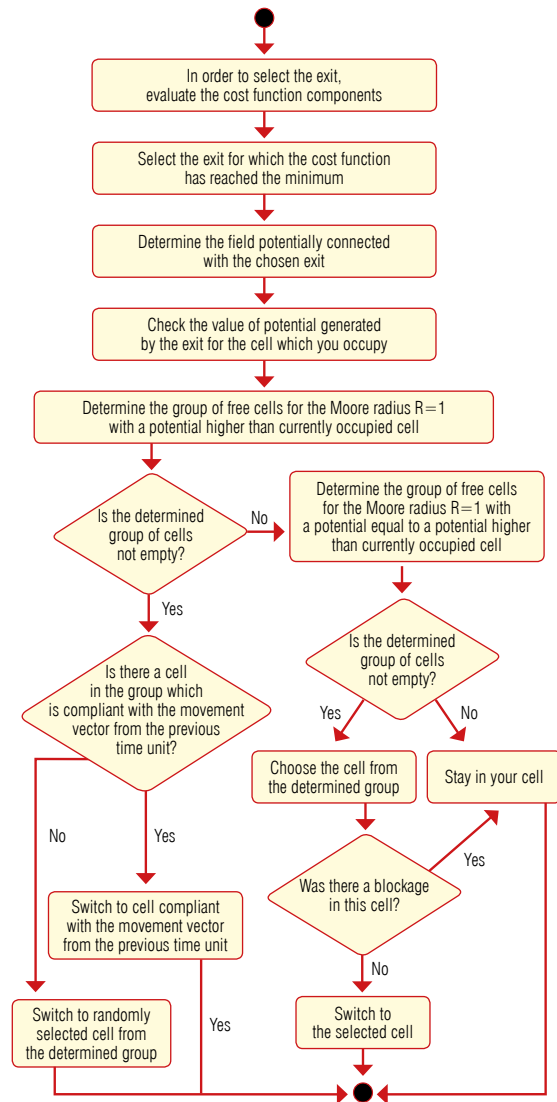


Figure 1. Diagram describing the algorithm of single pedestrian traffic in a model that takes into account the possibility of bottlenecks
Źródło: Own elaboration.

CROWDSIM

The CrowdSim simulation software consists of a simulation module, a visualisation section, and a two-dimensional map editor. It offers the possibility to design and carry out an evacuation simulation of people from relatively complex spaces.

Map editor

The first element the user has contact with is the two-dimensional map editor. The map editor window is shown in Figure 2.

no na zaaranżowanie i odtworzenie pomieszczeń wewnątrz budynków, jak i przestrzeni zewnętrznej, miast, placów itd.

The editor enables both to arrange and reproduce rooms inside buildings, outside areas, cities, squares, etc.



Rycina 2. Widok edytora map oraz zakładki „Design” górnego menu aplikacji CrowdSim

Figure 2. The map editor view and the "Design" tab of the top menu of the CrowdSim application

Źródło: Opracowanie własne.

Source: Own elaboration.

Edytor umożliwia odwzorowanie przestrzeni w odpowiedniej skali. Liczba opcji i możliwości jest na tyle duża, że bez problemu można odwzorować nawet bardzo skomplikowane obiekty i przestrzenie. Bardziej złożone konstrukcje architektoniczne można zapisywać w postaci definiowalnych obiektów, aby potem używać je ponownie bez konieczności projektowania ich na nowo. Edytor umożliwia zapisanie nowo utworzonej mapy i późniejszą jej edycję. Dzięki temu po wykonanej symulacji można modyfikować rozkład wyjść i pomieszczeń w kontekście procesu ewakuacji ludzi. Wszystkie mapy utworzone za pośrednictwem programu, zapisywane są w postaci wektorowej w plikach xml, zrozumiałych dla technologii WPF i .NET Framework. Dla projektanta udostępnionych jest wiele predefiniowanych kształtów i obiektów, takich jak kolumny, słupki itd. W programie dostępne są trzy rodzaje wyjść:

1. wyjścia wewnętrzne – przejścia między pomieszczeniami;
2. wyjścia ewakuacyjne – po przejściu przez nie uczestnik symulacji jest usuwany z mapy;

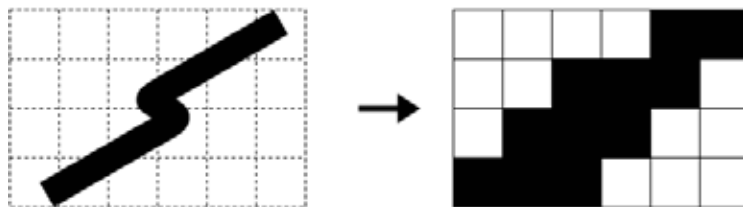
The editor enables the reproduction of a given space in user-selected scale. The number of options and possibilities is large enough to easily reproduce even highly complex objects and spaces. More complex architectural structures can be saved as definable objects and then reused without the need to be redesigned. Once created, the map can be saved as well and edited at a later time. This means that after a simulation is completed, the user can modify the project in order to design the layout of exits and rooms as effectively as possible in the context of evacuation of people. All maps created in the program are saved in a vectoral format as xml files compatible with the WPF and .NET Framework technologies. The designer has many predefined shapes and objects to choose from, including columns, posts, etc. There are three types of exits available in the program:

1. internal exits – passages between rooms;
2. evacuation exits – on passing it, the simulation participant is removed from the simulation
3. evacuation zones – in an open-area simulation, zones in which the simulation participant can be safely removed from the map, provided that he enters such a zone.

3. strefy ewakuacyjne – w przypadku symulowania terenów otwartych taka strefa oznacza miejsce, po dotarciu do którego uczestnik symulacji może zostać bezpiecznie usunięty z mapy.

Tak jak wspomniano powyżej, wszystkie mapy utworzone w edytorze zapisywane są w pamięci komputera w postaci wektorowej. Algorytm oparty o automaty komórkowe i systemy wieloagentowe przyjmuje, jednakże jako wejście jedynie macierz komórek. W związku z tym kolejną rolą edytora jest konwersja mapy wektorowej na mapę rastrową.

Pojedyncza komórka może oznaczać puste pole, przeszkodę, aktora lub wyjście. Wszystkie elementy architektoniczne takie jak ściany, kolumny, słupki, barierki oznaczane są jako komórki reprezentujące stan zajęty – przeszkodę, wyjścia oznaczane są oddzielnym typem, natomiast wszystkie pozostałe komórki definiowane są jako wolne. W procesie konwersji cała mapa dzielona jest na kwadraty o boku 40 cm. Jest to najmniejsza jednostka powierzchni brana pod uwagę w procesie symulacji, reprezentuje ona powierzchnie zajmowaną przez pojedynczego aktora [15]. Jeżeli w danym kwadracie choćby częściowo znajduje się jakiś element architektoniczny oznaczający przeszkodę komórka automatycznie zostaje oznaczona jako zajęta, podobnie w przypadku wyjść (w przypadku niektórych ustawień programu następuje również rozróżnienie na komórki oznaczające wyjścia wewnętrzne oraz ewakuacyjne). Na rycinie 3 przedstawiona została przykładowa wizualizacja takiej konwersji, gdzie kolorem czarnym oznaczone zostały komórki zajęte (przeszkoda).



Rycina 3. Przykład wizualizacji konwersji między mapą wektorową a macierzą komórek

Figure 3. Sample visualisation of conversion between a vector map and a cell matrix

Źródło: Opracowanie własne.

Source: Own elaboration.

Po dokonaniu konwersji mapy wektorowej przechodzi się do rozmieszczenia aktorów na mapie. Każdy aktor zajmuje dokładnie jedną komórkę. Edytor umożliwia dwa sposoby rozlokowania uczestników symulacji. Pierwszym z nich jest użycie kursora. Po przytrzymaniu lewego przycisku myszy i przeciągnięciu kursora na jego ścieżce ruchu pojawiają się aktorzy. Drugim sposobem jest zaznaczanie danego obszaru i podanie dokładnej liczby osób, która ma się tam znaleźć. Program sam losowo rozlokuje wtedy aktorów na zadanym obszarze.

Symulator

Symulator pełni kluczową rolę w całym systemie. Oprócz przeprowadzania samej symulacji odpowiedzialny jest również w pewnej części za przygotowanie macierzy komórek oraz za rankingowanie pól i zbieranie statystyk w czasie samej symulacji. Symulator składa się z trzech podstawowych modułów:

As mentioned above, all maps created in the editor are recorded in computer memory as vectors. However, the algorithm based on cellular automata and multi-agent systems only assumes a cell matrix as the entrance. Therefore, the editor's next task is to convert the vectorial map to a raster map.

A single cell can denote an empty field, an obstacle, an actor, or an exit. All architectural elements such as walls, columns, posts, railings are marked as cells representing occupied state – an obstacle, exits are marked as a separate type, and all the remaining cells are defined as unoccupied. During conversion, the entire map is divided into 40 cm x 40 cm squares being the smallest units included in the simulation process and representing the space occupied by a single actor [15]. If a square, in any part, includes any architectural element being an obstacle, the cell is automatically marked as occupied. The same applies to exits (depending on the program settings, the cells can be differentiated into internal and evacuation exits). Figure 3 shows an example of visualisation of such conversion, where occupied cells (obstacle) are marked with black.

Once the map conversion is complete, actors can be positioned on the map. Each actor occupies exactly one cell. The editor offers two ways to locate the simulation participants. The first one involves the mouse pointer. While holding the left mouse button and dragging it along, actors will appear along its path of movement. The second method requires the user to select an area and provide the exact number of people to be placed there. The program will then randomly place the actors within this particular area.

Simulator

The simulator serves a key role in the whole system. Apart from the simulation process itself, it is also responsible for preparing a cell matrix, ranking fields, and collecting statistical data during the simulation. The simulator consists of three basic modules:

- modułu rankingowania pól,
- modułu symulacyjnego,
- modułu zbierania statystyk.

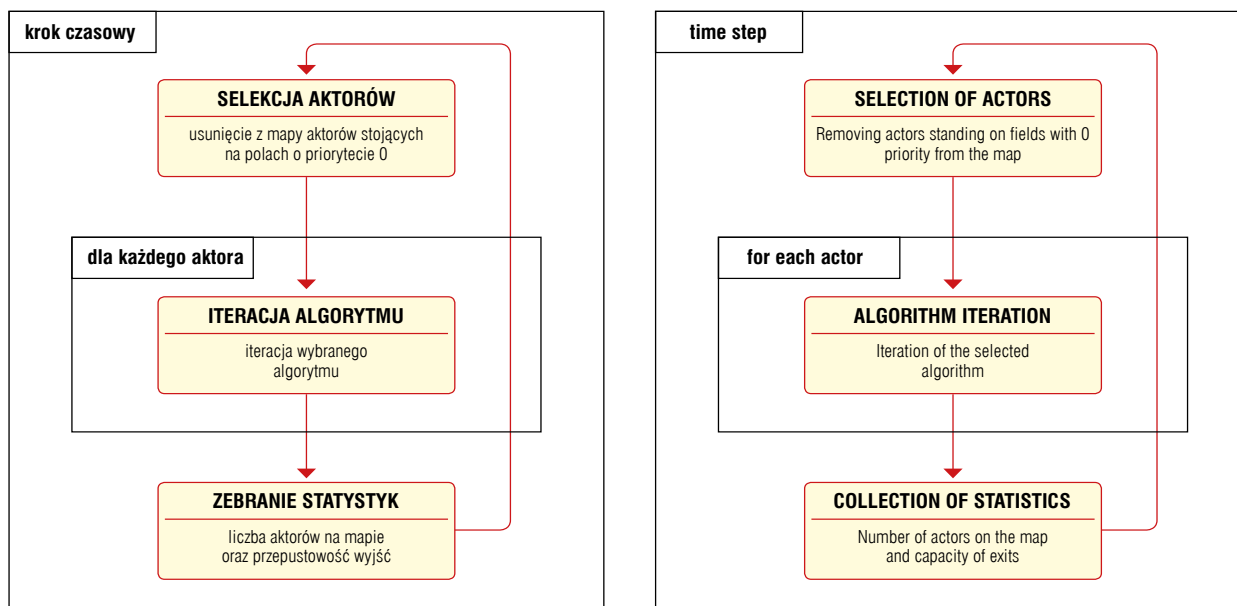
Moduł rankingowania pól odpowiedzialny jest za przypisanie każdej komórce odpowiednio wyliczonego priorytetu w postaci liczby całkowitej. Wartość ta jest wykorzystywana przez algorytm symulacyjny w procesie podejmowania decyzji o ruchu. Ogólna zasada zakłada, że im niższa wartość, tym pole znajduje się bliżej wyjścia. W związku z tym w procesie symulacji każdy aktor powinien dążyć do zajmowania pól z coraz to niższym priorytetem. Jako priorytet zerowy oznaczane są wyjścia i strefy ewakuacyjne. W programie symulacyjnym zaimplementowano trzy różne algorytmy rankingowania pól. Każdy z nich jest odpowiedni dla innego rodzaju map (pomieszczenia zamknięte lub tereny otwarte). Poprzez interfejs użytkownik ma możliwość wybrać samodzielnie rodzaj algorytmu rankującego lub też ustawić opcję automatycznego wyboru – wówczas program analizuje strukturę mapy oraz sam dokonuje wyboru najbardziej odpowiedniego algorytmu.

Moduł symulacyjny realizuje trzy opisane we wcześniejszej części artykułu algorytmy które użytkownik może wybrać podczas przygotowywania symulacji: algorytm modelu podstawowego **CA**, algorytm uwzględniający strategiczne możliwości pieszych **CA-A** oraz algorytm uwzględniający dodatkowo możliwości pojawiania się wąskich gardeł **CA-B**. Co więcej, użytkownik ma możliwość konfiguracji takich parametrów jak: współczynnik paniki (wykorzystywany w trzecim opisywanym algorytmie jako prawdopodobieństwo blokady) oraz wartość decydującą, czy podczas swojej decyzji odnośnie do wyboru wyjścia ewakuacyjnego pieszy będzie kierował się w większym stopniu zagęszczeniem osób przy danym wyjściu czy odległością. Po wyborze odpowiednich algorytmów oraz parametrów program jest gotowy, aby przejść do obliczania symulacji. Proces symulacji jest podzielony na kroki czasowe, których długość zależy od współczynnika paniki (im wyższy współczynnik paniki tym krótsze kroki czasowe) i wynosi od 0,8 do 1,2 sek. Parametry ustawione na wejściu pozostają niezmiennie przez cały czas działania symulacji. W każdym kroku czasowym odbywa się pojedyncza iteracja wybranego przez użytkownika algorytmu dla każdego z aktorów oddzielnie. Po obliczeniu iteracji dla wszystkich aktorów na mapie w danym kroku czasowym (rycina 4), następuje zebranie danych takich jak liczba aktorów na mapie w danej chwili oraz obciążenie wyjść (czyli liczba osób przechodzących przez dane wyjście w perspektywie czasu). Za to działanie odpowiedzialny jest **moduł zbierania statystyk** opisany w następnej części pracy.

- **the field ranking module,**
- **the simulation module,**
- **the statistical data collection module.**

The field ranking module assigns to each cell an appropriately calculated priority in the form of an integer value. This value is used by the simulation algorithm during the movement decision-making process. As a rule, the lower the value, the closer the field is to the exit. Accordingly, during the simulation process, each actor should aim at occupying fields with increasingly lower priority. Exits and evacuation zones are marked as having zero priority. The simulation program uses three field ranking algorithms. Each of them is appropriate for a different type of map (closed rooms or open areas). Through the interface, the user can choose the ranking algorithm or configure the automatic selection option to allow the program to analyse the structure of the map and select the most appropriate algorithm.

The simulation module implements the three algorithms described earlier, which the user can select while preparing the simulation: the CA basic model algorithm; the **CA-A** algorithm accounting for pedestrians' strategic capabilities; and the algorithm additionally accounting for the possible occurrence of bottlenecks **CA-B**. Furthermore, the user has the possibility of configuring such parameters as the panic coefficient (used by the third algorithm as the probability of blockages), and a value deciding whether in his decision on which evacuation exit to choose, the pedestrian will be motivated more by the concentration of people around an exit or by the distance. After selecting appropriate algorithms and parameters, the program is ready to perform the simulation. The simulation process is divided into time steps the length of which depends on the panic coefficient (the higher the coefficient, the shorter the time steps) and which can take between 0.8 and 1.2 seconds. The parameters set at the beginning remain unchanged throughout the simulation process. Each time step includes a single iteration of the algorithm selected by the user for each actor separately. After calculating the iteration for all actors on the map in a particular time step (Fig. 4), data are collected, such as number of actors on the map at a given moment and the load of exits (i.e. the number of people passing through the exits over time). This task rests with **the statistical data collection module** described in the further part of the article.



Rycina 4. Struktura pojedynczego kroku czasowego w procesie symulacji

Figure 4. The structure of a single time step in the simulation process

Źródło: Opracowanie własne.

Source: Own elaboration.

Wizualizacja symulacji oraz przedstawienie statystyk

W wyniku działania algorytmu symulacyjnego powstają dwa pliki, jeden z nich reprezentuje schemat ruchów poszczególnych aktorów, tak aby można było go wizualizować, a drugi zawiera statystyki w ujęciu czasu. Moduł wizualizacji jest częścią oprogramowania i wykorzystuje on oba pliki w swoim działaniu.

Visualising the simulation and presenting statistical data

The simulation algorithm generates two files – the first one represents actors' movement patterns to be visualised later; the second contains statistical data distributed over time. The visualisation module constitutes part of the software and uses both files in its operation.



Rycina 5. Zrzut ekranu procesu symulacji programu

Figure 5. A screenshot from the simulation process

Źródło: Opracowanie własne.

Source: Own elaboration.

RESEARCH AND DEVELOPMENT

Sama wizualizacja odtwarzana jest z wcześniej przygotowanego pliku, więc można ją dowolnie przewijać i zatrzymywać. Dodatkowo aktorzy rozróżnieni są kolorami od zielonego po czerwony. Odcienie kolorów zielonego i żółtego oznaczają swobodnie poruszających się ludzi, natomiast pomarańczowy i czerwony na mapie oznacza duże zagęszczenie aktorów, najczęściej wąskie gardło. Są to miejsca potencjalnie niebezpieczne, mogą one zamienić się w ogniska paniki i szybko rozprzestrzenić się na pozostałych aktorów. W trakcie działania algorytmów symulacyjnych duży wpływ na wynik końcowy ma element losowości, dlatego też każda symulacja, nawet przy identycznych danych wejściowych różni się od siebie. Na rycinie 5 przedstawiono obraz z przykładowej symulacji. Czarnym kolorem oznaczone są ściany i przeszkody, wyjścia oznaczone jaśniejszym zielonym oznaczają drzwi wewnętrzne, natomiast ciemniejszym wyjścia ewakuacyjne.

Widok statystyk umożliwia podgląd pięciu różnych kategorii:

1. Pozostali piesi (Remaining actors) – wykres pokazuje liczbę aktorów na mapie w perspektywie czasu;
2. Obciążenie wyjść (Exits load) – pokazuje liczbę osób opuszczających teren przez dane wyjście w perspektywie czasu;
3. Użycie wyjść (Exits usage) – pokazuje na wykresie kołowym ile osób procentowo opuściło teren przez dane wyjście;
4. Przepustowość wyjść (Exits capacity) – pokazuje w formie wykresu słupkowego, ile osób maksymalnie może opuścić teren przez dane wyjście;
5. Wydajność wyjść (Exits performance) – pokazuje na wykresie w jakim procencie dane wyjście było wykorzystane w trakcie trwania ewakuacji. Jeżeli wartość wynosi 100% oznacza to, że przez cały okres symulacji przez wyjście wydostawała się maksymalna możliwa ilość osób.

Przykładowe zobrazowanie statystyk zostało przedstawione na rycinie 6.

Do oprogramowania dodany został eksperymentalnie moduł wizualizacji 3D za pomocą silnika graficznego Unity3D [21]. Moduł ten między innymi umożliwia cofanie i zatrzymywanie symulacji. Bazuje on na tym samym pliku wynikowym z zapisem ruchu co wizualizacja 2D. Udostępnia jednakże kilka dodatkowych opcji, takich jak oznaczenie szybkości ruchu oraz linie narysowane na mapie oznaczające ścieżki ruchu pieszych (rycina 7).

Since the visualisation is reproduced from a pre-prepared file, it can be rewound and stopped as needed. In addition, actors are marked with various colours, from green to red. Shades of green and yellow denote freely moving people, whereas orange and red on the map mean a high concentration of actors, i.e. usually a bottleneck. These are potentially dangerous areas which can potentially turn into panic points and rapidly proliferate over the remaining actors. While the simulation algorithms are in operation, the final result is largely affected by the randomness aspect. Every simulation is therefore different, even if the input data remain unaltered. Figure 5 shows a screen from a sample simulation. The walls and obstacles are marked in black, the exits marked with lighter green indicate internal doors, whereas a darker shade of green indicates evacuation exits.

The user can browse through five different categories in the Statistics section:

1. Remaining Actors – a chart which shows a number of actors on the map in time;
2. Exits Load – shows the number of people leaving the area through a particular exit over time;
3. Exits Usage – shows, on a circular chart, how many people (in percent) left the area through a particular exit;
4. Exits Capacity – shows, in the form of a bar chart, how many people maximally can leave the area through a particular exit;
5. Exits performance – in the chart it shows in what percent the particular exit was used during the evacuation. If this value equals 100%, it means that a maximum possible number of people got through the exit during the evacuation.

A sample illustration of statistics is shown in Fig. 6.

The software was experimentally equipped with a 3D visualisation module using the Unity3D graphical engine [21]. This module enables, e.g. rewinding and stopping the simulation. It relies on the same output file with movement recording as 2D visualisation. However it provides several additional options, such as determination of movement speed and lines drawn on the map indicating pedestrian movement paths (Fig. 7).

Porównanie modeli dynamiki pieszych

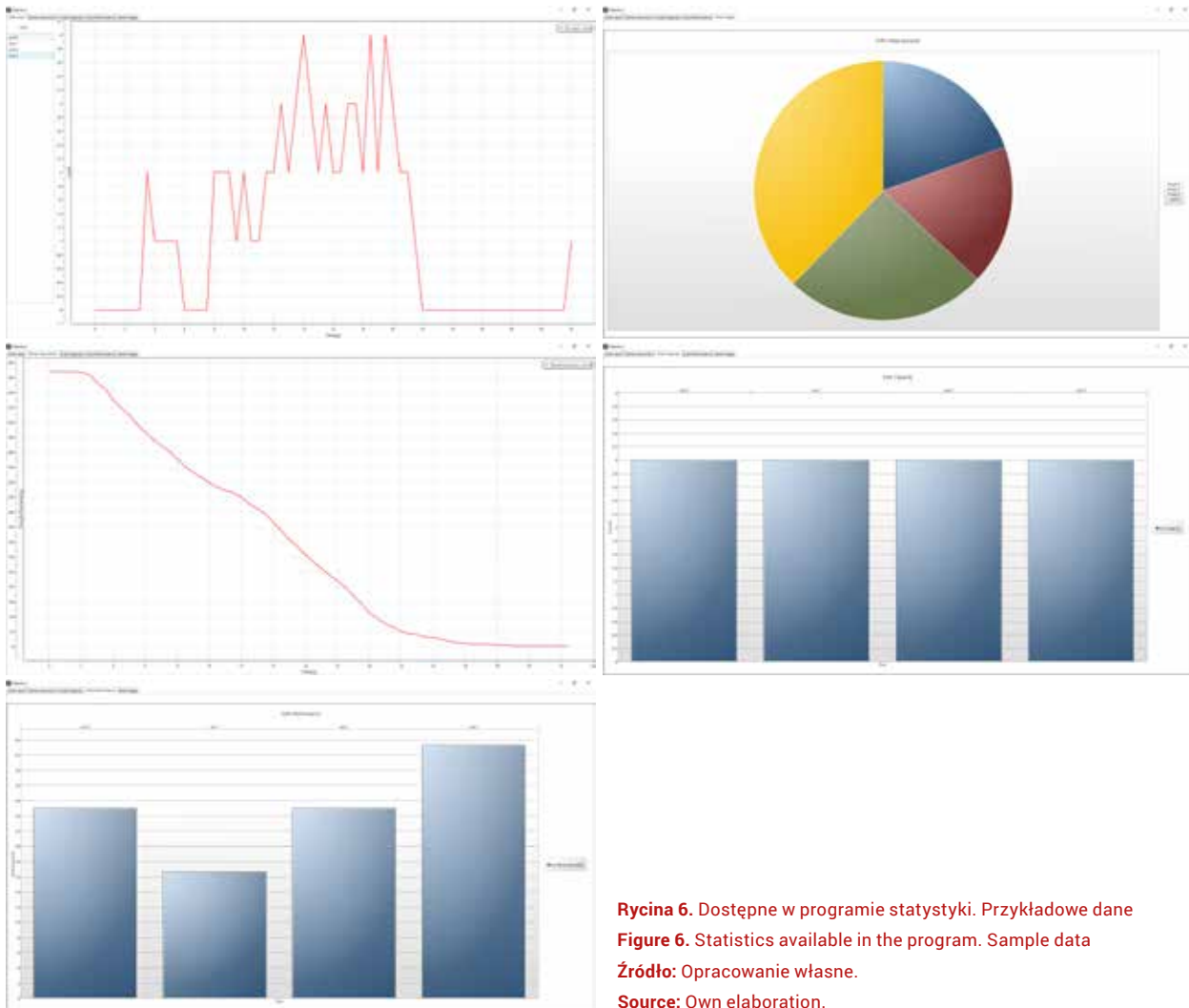
Porównanie algorytmów rankingowania komórek macierzy

Jednym z najważniejszych elementów przygotowania do symulacji jest nadanie każdej komórce odpowiedniego rankingu. To od tej właśnie wartości zależeć będzie charakterystyka ruchu aktorów w procesie ewakuacji. Wartość rankingu danej komórki w procesie symulacji jest dla każdego z aktorów wyznacznikiem drogi, każdy z nich dąży do przemieszczenia się na pola o coraz niższym rankingu. W przypadku bardziej skomplikowanych scenariuszy, gdy na mapie znajduje się więcej niż jedno wyjście ewakuacyjne komórka może posiadać wiele wartości rankingu,

Comparison of pedestrian dynamics models

Comparison of algorithms for ranking matrix cells

One of the most important elements of preparation for the simulation process is to assign appropriate ranking to each cell. This value will affect the characteristics of movements of actors in the evacuation process. Ranking value of a particular cell in the simulation process constitutes a path indicator. Each one of them aims at fields on lower and lower ranking. In the case of more complex scenarios, when more than one evacuation exit is present on the map, the cell can have various ranking



Rycina 6. Dostępne w programie statystyki. Przykładowe dane
Figure 6. Statistics available in the program. Sample data
Źródło: Opracowanie własne.
Source: Own elaboration.



Rycina 7. Obraz wizualizacji 3D
Figure 7. 3D image visualisation
Źródło: Opracowanie własne.
Source: Own elaboration.

oddzielnych dla każdego wyjścia. Wówczas to aktor decyduje o wyborze wyjścia zgodnie z zaimplementowanym algorytmem.

W zależności od rodzaju mapy oraz rozkładu pomieszczeń dobór metody rankingowania pól może się różnić. W przypadku otwartych przestrzeni sprawa wygląda dosyć prosto, natomiast komplikuje się znacząco w przypadku mapy z rozbudowaną strukturą pomieszczeń. W obu tych przypadkach nie można użyć tego samego algorytmu, aby uzyskać zadowalający skutek. Dlatego też w przypadku opisywanego oprogramowania symulacyjnego zaimplementowano trzy zupełnie różne algorytmy rankingowania pól. Za dobór odpowiedniej metody odpowiada specjalny moduł oprogramowania symulacyjnego. Jest on w stanie przeanalizować strukturę mapy, rodzaj terenu oraz poziom skomplikowania siatki pomieszczeń i dobrać najbardziej optymalny sposób rankingowania komórek.

Wszystkie przedstawione poniżej algorytmy rankingowania komórek są algorytmami autorskimi napisanymi na potrzeby oprogramowania symulacyjnego [20]:

- **SRA** (*Simple Ranking Algorithm*) jest to najprostszy i najbardziej naiwny ze wszystkich zaimplementowanych algorytmów. Wartość rankingu danej komórki obliczana jest dokładnie tak jak odległość między dwoma punktami układu współrzędnych z tym, że wartości x oraz y są odpowiednio wartościami indeksu wiersza i indeksu kolumny danej komórki. Metoda ta jest bardzo prosta i szybka oraz łatwa do zrównoleglenia. Program wykorzystują ją do rankingowania przestrzeni otwartych nie zawierających oddzielnych pomieszczeń lub stref oddzielonych barierkami. Algorytm ten natomiast zupełnie nie sprawdza się w przypadku przestrzeni zamkniętych podzielonych na oddzielne pomieszczenia ponieważ nie bierze pod uwagę żadnych przeszkód, znajdujących się na drodze między daną komórką a wyjściem. Na rycinie 8 przedstawiony został wynik działania algorytmu **SRA** na przykładowym pomieszczeniu składającym się z trzech pokoi oddzielonych wewnętrznymi przejściami. Jak widać, algorytm błędnie ponumerował pola znajdujące się w pomieszczeniu o numerze 1. Komórki tego pokoju faktycznie licząc wektorowo znajdują się blisko wyjścia jednak aby do niego dotrzeć należy iść okrężną drogą przez pokój oznaczony numerem 2.
- **BFS-RA** (*Breadth First Search Ranking Algorithm*) jest to metoda oparta na popularnym algorytmie grafowym przeszukiwania wszerz [19]. Mapa komórkowa potraktowana jest w tym przypadku jak graf, w którym poszczególne komórki są wierzchołkami, z czego każdy z wierzchołków ma połączenie jedynie z otaczającymi go innymi komórkami (wierzchołkami). Korzeniem grafu jest każda komórka reprezentująca wyjście ewakuacyjne. Dla każdej z komórek mapy obliczana jest na podstawie algorytmu najkrótsza ścieżka prowadząca do korzenia (wyjścia ewakuacyjnego). Dzięki takiemu podejściu można odpowiednio oznaczyć wszystkie komórki, nawet w przypadku bardzo skomplikowanej siatki pomieszczeń. Oczywiście koszt obliczeniowy tego algorytmu, w porównaniu z pierwszą metodą, jest znacznie wyższy i znacznie trudniejszy do zrównoleglenia. Dlatego jest on wybierany dopiero wtedy, gdy ukształtowanie mapy nie pozwala na zastosowanie algorytmu **SRA**.

values, separate for each exit. In such a case the actor decides which exit to choose according to the implemented algorithm.

Selection of a ranking method can differ depending on the type of map and layout of rooms. The case is quite simple for open areas, whereas it gets considerably complex in the case of maps with an extended structure of rooms. In both these cases one cannot use the same algorithm in order to obtain a satisfying result. Therefore, in the case of the discussed simulation software, three different field ranking algorithms were applied. Selection of the appropriate method is performed by the special simulation software module. It can analyse the map structure, the type of area, and the level of complexity of the room network, and then select an optimal cell ranking method.

All cell ranking algorithms shown below are proprietary algorithms written for simulation software purposes [20]:

- **SRA** (*Simple Ranking Algorithm*). The simplest and most “naïve” of all the implemented algorithms. The ranking value of a particular cell is calculated exactly as the distance between both points of the coordinate system; however, x and y values are the row index and column index values, respectively. This method is very simple, fast and easy to parallel. The program uses it to rank the open areas not including separate rooms nor areas separated with railings. Whereas this algorithm is totally unfit for closed areas divided into separate rooms because it does not take into account any obstacles between the cell and the exit. Figure 8 shows the result of operation of the **SRA** algorithm on a sample room consisting of three compartments separated by internal passages. As can be seen, the algorithm incorrectly numbered the fields included in the room number 1. Cells in this room, when counting by vectors, are actually located near the exit. However, in order to approach it, one has to go via a circuitous route through the room number 2.
- **BFS-RA** (*Breadth First Search Ranking Algorithm*) A method based on a popular graph algorithm of breadth search [19]. The cell map is treated in this case as a graph in which particular cells are peaks and each peak is connected only to the surrounding cells (peaks). The root of the graph is constituted by each cell representing the evacuation exit. For each cell of the map, the shortest path leading to the root (evacuation exit) is calculated. With such an approach, all cells can be marked, even in the case of highly complex room mesh. Understandably, the calculation cost of this algorithm, as compared to the first method, is considerably higher and harder to parallel. Therefore it is selected only when the form of the map does not allow to use the **SRA** algorithm.

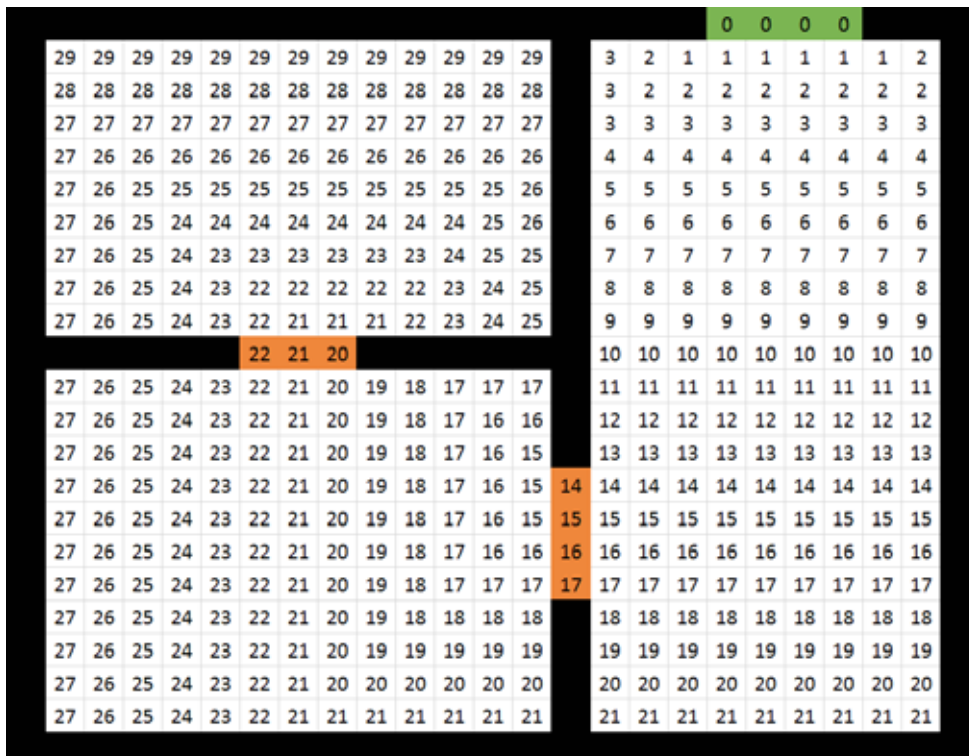


Rycina 8. Przykładowy wynik działania algorytmu SRA

Figure 8. An example of the result of the SRA algorithm

Źródło: Opracowanie własne.

Source: Own elaboration.



Rycina 9. Reprezentacja metody rankingowania BSF-RA

Figure 9. Representation of the BSF-RA ranking method

Źródło: Opracowanie własne.

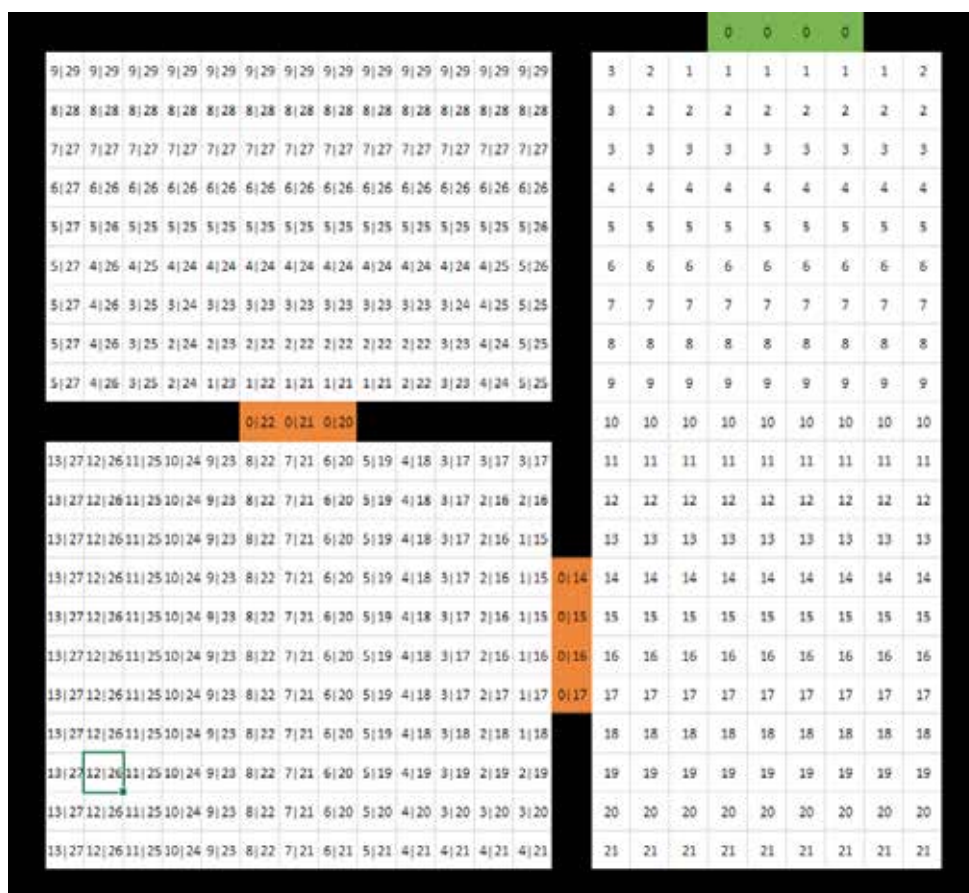
Source: Own elaboration.

Jak przedstawiono na rycinie 9, pola zostały ponumerowane zupełnie inaczej niż w pierwszym przypadku. Skrajne lewe górne pomieszczenie posiada największe wartości rankingu, ponieważ w tym przypadku nie liczy się odległość wektorowa lecz rzeczywista droga, jaką musi przejść aktor, aby dotrzeć do wyjścia. Nie ma tu możliwości zablokowania się aktora na mapie, tak aby nie był on w stanie opuścić pomieszczenia, jak to miało miejsce w przypadku metody SRA (oczywiście pod warunkiem poprawnego zaprojektowania mapy).

- **ARA (Advanced Ranking Algorithm).** Poprzedni model **BFS-RA** umożliwił wyznaczenie dokładnej ścieżki między wyjściem a dowolną komórką na mapie. Można wysnuć twierdzenie, że jest on zupełnie wystarczający do przeprowadzenia dowolnie skomplikowanej symulacji. Jednak w praktyce okazuje się, że aktorzy podczas ewakuacji bardzo nieefektywnie wykorzystują dostępną przestrzeń. Każdy z aktorów podąża idealnie najkrótszą drogą do wyjścia, tak jak by był świadom planu pomieszczeń. W rzeczywistych warunkach natomiast ewakuacja sprowadza się do opuszczania kolejno następujących po sobie pomieszczeń na drodze do wyjścia. Ścieżki ewakuacyjne najczęściej oznaczone są tabliczkami, które kierują przechodniów do kolejnych drzwi wewnętrznych (przejściowych) na drodze do wyjścia ewakuacyjnego.

As shown in Fig. 9, fields were numbered differently than in the first case. The top left room has the highest ranking values because in this case the vector distance does not count but the actual distance which has to be made by the actor in order to reach the exit. There is no possibility of blocking the actor on the map without a possibility of leaving the room as in SRA method (of course assuming that the map has been correctly designed).

- **ARA (Advanced Ranking Algorithm).** The previous **BFS-RA** model enabled to determine the exact path between the exit and any cell on the map. It can be stated that it is totally sufficient to execute simulation with any level of complexity. In practice, it turns out that actors during the evacuation use the available area very inefficiently. Each actor takes an ideally shortest path to the exit as if he knew the layout of rooms. In real-life conditions, however, evacuation consists in leaving subsequent rooms on the road to exit. Evacuation paths are most often marked with plates which lead pedestrians to next internal (connecting) doors on the road to evacuation exit.



Rycina 10. Graficzne przedstawienie efektu działania algorytmu ARA

Figure 10. Graphical representation of the effect of the ARA algorithm

Źródło: Opracowanie własne.

Source: Own elaboration.

Według tej wiedzy każdy pokój należałoby traktować jako oddzielną mapę a drogę ewakuacyjną jako kolejność pokoi, które należy przejść aby wydostać się z pomieszczenia. Dlatego też powstała konieczność stworzenia kolejnej metody rankingowania, która oprócz globalnego rankingu całej mapy udostępni również ranking każdego z pomieszczeń osobno. Algorytm **ARA** stworzony na potrzeby tej pracy wprowadza pojęcie rankingu lokalnego oraz globalnego, gdzie lokalny odnosi się do konkretnego pomieszczenia, a globalny do całej powierzchni budynku. **ARA** oparty jest również na metodzie przeszukiwania grafu w szerz z tą różnicą, że ranking tworzony jest dla całej mapy i dla każdego z poszczególnych pomieszczeń oddzielnie. Opuszczając poszczególne pomieszczenia, aktorzy kierują się lokalnym rankingiem. Dzięki temu cały proces symulacji wygląda znacznie realistyczniej i naturalniej [20]. Przystawienie efektu działania algorytmu **ARA** zostało zaprezentowane na rycinie 10.

Porównanie modeli CA

Podstawową różnicą między modelami **CA** i **CA-A** jest wprowadzenie w modelu **CA-A** pojęcia funkcji kosztu, dzięki której aktor może zdecydować o wyborze najkorzystniejszego wyjścia. Aby w pełni dostrzec różnicę między tymi dwoma modelami, należy odpowiednio przygotować mapę. Powinno na niej znajdować się więcej niż jedno wyjście ewakuacyjne, tak aby widoczne mogło być działanie funkcji kosztu. Zaprezentowane poniżej mapy zostały przygotowywane specjalnie w celu uwidocznienia różnic między rozpatrywanymi modelami. Model **CA-B** jest modyfikacją modelu **CA-A** uwzględniającą mechanizm blokad i w związku z tym nie należy spodziewać się, że dodanie mechanizmu blokad w znaczący sposób wpłynie na wyniki symulacji. Należy zatem spodziewać się, że różnice między modelami **CA-A** i **CA-B** będą raczej niewielkie.

Na potrzeby porównania modeli przygotowana została specjalna mapa budynku o wymiarach kontyngencji 40 na 22,5 metra (900 m²) zdolnego pomieścić maksymalnie 3000 osób. Składa się on z 6 dużych pokoi po 75 m² każdy, dwóch korytarzy oraz obszernego holu. Budynek posiadał trzy wyjścia ewakuacyjne na ścianie południowej. Cztery pokoje miały po jednych drzwiach wyjściowych natomiast dwa posiadały po dwa wyjścia po obu stronach pokoju. Każde z wyjść ewakuacyjnych miało szerokość 1,5 m. Rycina 11 przedstawia plan opisywanego pomieszczenia.

Zgodnie z przedstawionym planem budynku aktor mógł wybierać między jednym z trzech wyjść ewakuacyjnych, dzięki temu będzie można zauważyć różnice między modelami **CA** i **CA-A**. Kolejną istotną kwestią było rozmieszczenie aktorów na mapie. Celem pierwszego eksperymentu numerycznego było pokazanie różnicy między oboma modelami poprzez wykorzystanie funkcji kosztu. Stąd też aktorzy zostali rozmieszczeni tak, aby dla zdecydowanej większości z nich najbliższym wyjściem było wyjście numer 1 (rycina 12).

Można zauważyć, że w przypadku modelu **CA** aktorzy gromadzą się przy jednym z wyjść, podczas gdy pozostałe zostają w większości nieużywane (rycina 13). Wynika to z faktu, że w modelu **CA** brakuje funkcji kosztu, zatem aktor zawsze wybiera wyjście najbliższe. Inaczej sprawa ma się w przypadku modelu **CA-A**. Rycina 14 pokazuje, że aktorzy w miarę równo obciążają każde z wyjść. Są oni w stanie podjąć decyzje odnośnie do wyboru wyjścia nie tylko na podstawie odległości, ale

According to this knowledge, each compartment should be treated as a separate map, and the evacuation route as a sequence of compartments which need to be passed in order to get out of the room. Therefore, there was a necessity to create the next ranking method which apart from the global ranking of the whole map, also makes the rank of each room individually available. The **ARA** algorithm created for the purposes of this paper introduces a term of local and global ranking where the local one relates to the particular room and global – to the whole area of the building. **ARA** is also based on the breadth search of the graph with such a difference that the rank is created for the whole map and for each of the rooms separately. By leaving subsequent rooms, the actors are directed by the local ranking. Due to this, the whole simulation process looks much more realistic and natural [20]. The effect of **ARA** algorithm operation effect is presented in Fig. 10.

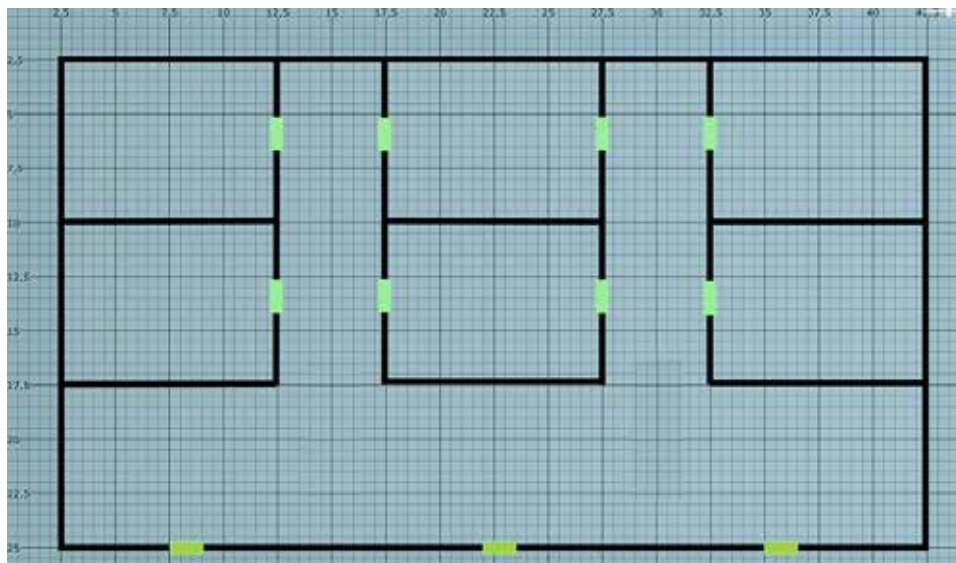
Comparison of CA models

The main difference between **CA** and **CA-A** models is the introduction of the concept of “cost function” in the **CA-A** model due to which the actor can decide which exit is the most favourable. In order to fully observe the difference between these two models, the map should be prepared properly and include more than one evacuation exit in order to make the cost function operation visible. The maps below were prepared specifically for the purpose of demonstrating differences between the analysed models. The **CA-B** model constitutes a modification of the **CA-A** model taking into account the blockage mechanism and, therefore, one should expect that any differences between **CA-A** and **CA-B** models will be rather small.

For the purposes of comparison of the models, a special map of the building was prepared with 40 m x 22.5 m (900 m²) floor dimensions which is able to contain maximum 3000 people. It consists of 6 large rooms, 75 m² each, two corridors and a spacious hall. The building had three evacuation exits on the southern wall. Four rooms were equipped with one exit door each, whereas two had two exits on both sides of the room. Each evacuation exit had the width of 1.5 m. Figure 11 shows the plan of the building.

According to the presented plan of the building, the actor could select between one of three evacuation exits. Due to this, differences between **CA** and **CA-A** models will be visible. The next essential issue was to distribute the actors on the map. The aim of the first numerical experiment was to show the difference between both models by using the cost function. This is why the actors were distributed in such way as to make the exit No. 1 the nearest exit (Fig. 12).

It can be observed that in the case of the **CA** model, the actors gather near one of the exits, while other exits remain mostly unused (Fig. 13). This stems from the fact that in the **CA** model there is no cost function and, therefore, the actor always selects the nearest exit. The situation is different in the case of **CA-A** model. Figure 14 shows that the actors occupy each exit quite equally. They are able to make a decision on which exit to choose not only basing on the distance but also the concentration of

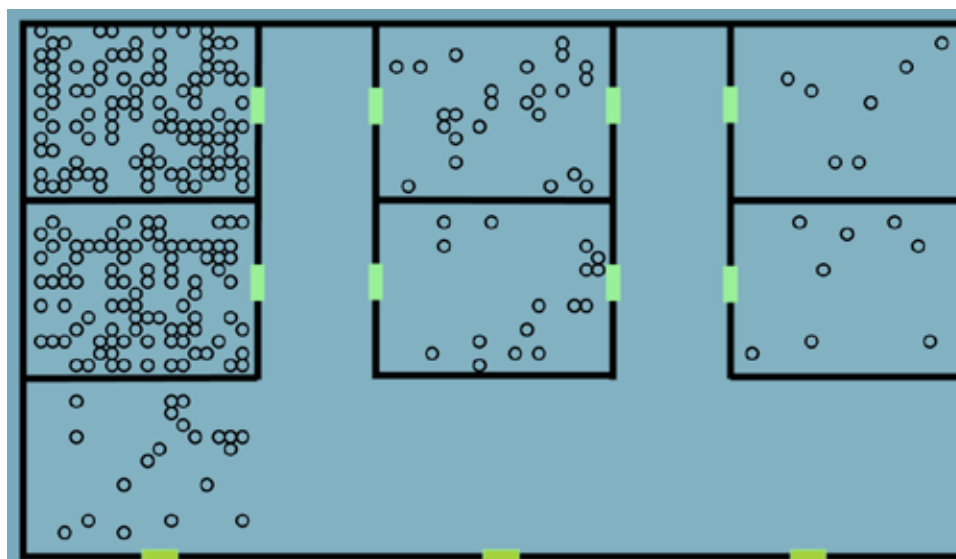


Rycina 11. Plan budynku przygotowanego na potrzeby symulacji

Figure 11. Plan of the building prepared for the needs of simulation

Źródło: Opracowanie własne.

Source: Own elaboration.



Rycina 12. Rozkład aktorów na przykładowej mapie

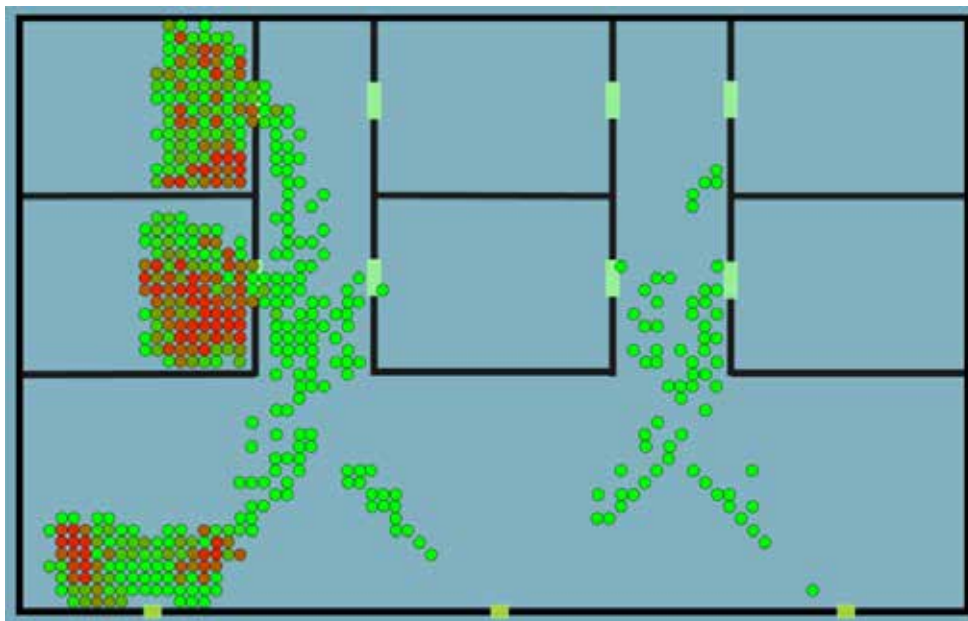
Figure 12. Distribution of actors on a sample map

Źródło: Opracowanie własne.

Source: Own elaboration.

również zagęszczenia aktorów przy danym wyjściu. Obserwację tę potwierdzają diagramy obciążenia wszystkich wyjść w ujęciu procentowym. W modelu **CA** jedno z wyjść (skrajne lewe) jest zdecydowanie najbardziej obciążone – ewakuowało się przez nie ponad 66% wszystkich aktorów na mapie. Natomiast w przypadku modelu **CA-A** rozkład obciążenia jest prawie równy pomiędzy wszystkimi trzema wyjściami. Jeżeli chodzi o czasy symulacji dla obu modeli, to są one bardzo zbliżone i wynoszą odpowiednio: **CA** 97 sekund a **CA-A** 85 sekund. Różnica na korzyść **CA-A** wynika z faktu, iż aktorzy zdecydowanie optymalnie wykorzystują wszystkie dostępne wyjścia, jednakże tracą tym samym cenne sekundy na zmianę swojej decyzji i kierunku

aktorów near a particular exit. This observation is confirmed by the load diagrams of all exits on a percentage basis. In the **CA** model, one exit (the leftmost) is definitely loaded the most – over 66% of all the actors on the map evacuated through this exit. Whereas in the case of the **CA-A** model, load distribution is nearly equal between all three exits. Simulation times for both models are very similar and equal: **CA** 97 seconds and **CA-A** 85 seconds, respectively. The difference in favour of **CA-A** results from the fact that the actors use all the available exits optimally, while losing valuable seconds for changing their decision and movement direction. By evaluating the whole simulation visually it can be observed that the **CA-A** model provides much

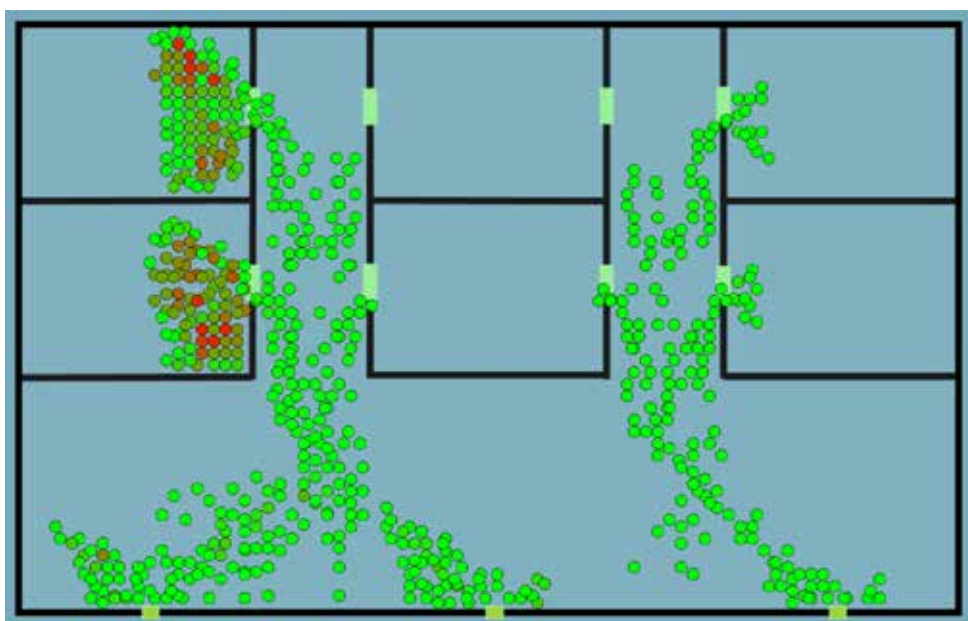


Rycina 13. Przebieg symulacji na przykładowej mapie dla modelu CA

Figure 13. The simulation process on a sample map for the CA model

Źródło: Opracowanie własne.

Source: Own elaboration.



Rycina 14. Przebieg symulacji na przykładowej mapie dla modelu CA-A.

Figure 14. The simulation process on a sample map for the CA-A model

Źródło: Opracowanie własne.

Source: Own elaboration.

ruchu. Oceniając wizualnie całą symulację, można zauważyć, że model CA-A daje znacznie bardziej realistyczne wyniki i lepiej symuluje zachowania ludzi w prezentowanym scenariuszu.

Przy porównaniu modeli CA-A i CA-B praktycznie niezauważalne są różnice w przypadku statystyk. Dopiero po bliższym przyjrzeniu się ruchowi aktorów podczas symulacji widać, że jeśli w przypadku czystego modelu CA-A ludzie poruszają się według idealnego porządku, o tyle w modelu CA-B zauważalne są

more realistic results and better simulates people's actions in the presented scenario.

When comparing the CA-A and CA-B models, there are practically no differences in statistics. Only after a closer look on the movement of actors during the simulation, it can be observed that in the case of the pure CA-A model, people move according to an ideal order. However, in the CA-B model, there are some

chwilowe blokady i zakleszczenia aktorów co nadaje całej wizualizacji jeszcze bardziej rzeczywistego charakteru.

Kolejną ważną kwestią wpływającą na przebieg symulacji są następujące współczynniki:

- **współczynnik dystansu**, który określa poziom ważności dystansu aktora od danego wyjścia potrzebny do określenia funkcji kosztu;
- **współczynnik zagęszczenia**, który określa ważność poziomu zagęszczenia tłumu wokół wyjścia w otoczeniu Moore'a dla zadanego promienia potrzebny dla określenia funkcji kosztu;
- **globalne prawdopodobieństwo blokady** zwane również czynnikiem paniki, wykorzystywane w modelu CA-B do obliczenia prawdopodobieństwa blokady w momencie, gdy dwóch aktorów wybrało docelowo tę samą komórkę w danym kroku czasowym.

Współczynniki dystansu i zagęszczenia są ze sobą ściśle powiązane. W aplikacji użytkownik określa je, przesuując suwak i określając, czy aktor przy wyborze wyjścia ma się kierować bardziej dystansem czy poziomem zagęszczenia.

Na kolejnej rycinie nr 15 przedstawione zostały wykresy pozostałych aktorów dla dwóch skrajnych przypadków zagęszczenia i dystansu. W pierwszym przypadku współczynnik dystansu został ustawiony na maksymalną wartość (rycina 15 linia a). Natomiast w drugim współczynnik zagęszczenia (rycina 15 linia b).

temporary blockages and jams of actors, which makes the visualisation even more real.

The next important issue affecting the simulation process is the following coefficients:

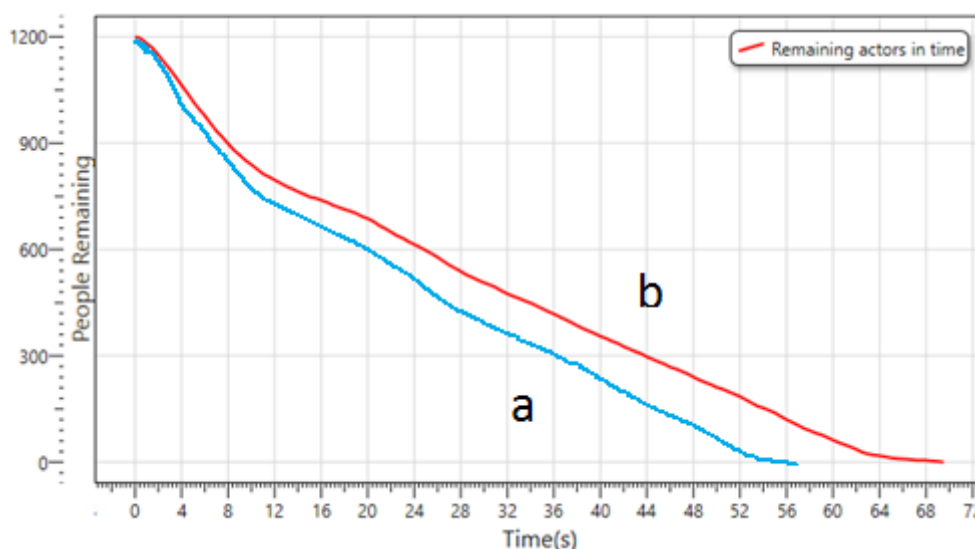
Distance coefficient, defining the level of importance of a distance between the actor and an exit needed to define the cost function;

Density coefficient, defining the importance of the level of crowd density around the exit in Moore neighbourhood for the particular radius necessary to define the cost function;

Global blockage probability also referred to as the panic factor, used in the CA-B model to calculate the probability of the blockage at the moment when two actors chose the same cell in a particular time step.

Distance and density coefficients are closely related to each other. Within the application, the user defines them by moving the slider and defining whether the actor has to take the distance or rather density level into account when selecting the exit.

Figure 15 includes charts of the remaining actors for two extreme cases of density and distance. In the first case, the distance coefficient was set to a maximum value (Fig. 15 line a). In the second case, it is the density coefficient (Fig. 15 line b).



Rycina 15. Wykres pozostałych na mapie aktorów w funkcji czasu dla maksymalnego współczynnika odległości (linia a) oraz dla maksymalnego współczynnika zagęszczenia (linia b)

Figure 15. A graph of the remaining actors on the map as a function of time for the maximum distance factor (line a) and for the maximum density factor (line b)

Źródło: Opracowanie własne.

Source: Own elaboration.

Jak można zauważyć czas ewakuacji przy maksymalnym współczynniku zagęszczenia wyniósł około 72 sekundy dla 1200 aktorów. Wykres pozostałych na mapie aktorów ma przebieg zbliżony do liniowego. Natomiast diagram użycia wyjść rozkłada się idealnie po równo na każde z trzech po 33%. Dla maksymalnego współczynnika odległości czas symulacji jest znacząco krótszy i wynosi 54 sekundy. Wynikać to może z faktu,

As can be observed, the evacuation time at the maximum density factor approximated 72 seconds for 1200 actors. The chart showing the remaining actors on the map is similar to a line. On the other hand, the exit use diagram distributes ideally equal per each of three at 33%. For the maximum distance coefficient, simulation time is considerably shorter and equals 54 seconds. This may result from the fact that the map prepared

iż przygotowana do symulacji mapa miała równomiernie rozłożone wyjścia względem powierzchni budynku, oraz że aktorzy na mapie zostali rozłożeni bardzo równomiernie, pokrywając całą mapę. Dzięki temu osoby konsekwentnie podążając do najbliższego wyjścia nie traciły czasu na zmiany decyzji jak w przypadku symulacji z maksymalną wartością współczynnika zagęszczenia. Co do użycia wyjść to, w przeciwieństwie do pierwszego przypadku rozkład nie jest już tak idealny (mniejsze użycie wyjście 3), choć obciążenie wyjść było dość równomierne, ale wynikało to bardziej z równomiernego rozłożenia wyjść i aktorów na mapie niż z samego algorytmu.

Podsumowanie

Celem pracy było przedstawienie i porównanie klasycznych modeli dynamiki pieszych symulujących proces ewakuacji ludzi z różnego typu pomieszczeń. Jak zademonstrowano, każdy scenariusz ewakuacji wymaga indywidualnego doboru metod i parametrów. Niestety nie ma możliwości stworzenia uniwersalnego algorytmu i doboru współczynników, tak aby optymalnie opisać każdą sytuację.

Przedstawione wyniki symulacji skłaniają do postawienia ważnego pytania dotyczącego możliwości wiarygodnego symulowania inteligentnego zachowania się ludzi w procesie ewakuacji przy pomocy oprogramowania komputerowego. Odpowiedź na tak zadane pytanie wcale nie okazuje się prosta ponieważ pomimo poprawnego zaimplementowania algorytmów **CA**, **CA-A** oraz **CA-B** pozostaje duży niedosyt w kwestii wiarygodności oraz realizmu przeprowadzonych tymi metodami symulacji. Mechanizmy te okazują się często zbyt proste i niewystarczające do opisanego skomplikowanego procesu ruchu pieszych.

Kolejnym problemem jest optymalizacja. Algorytmy oparte o automaty komórkowe nie są tak proste do zrównoleglenia, zatem przy dużej liczbie aktorów podczas symulacji pojawić się mogą problemy wydajnościowe. Mając na uwadze opisane wyżej problemy, zaproponowano w pracy znaczące usprawnienia algorytmów symulacyjnych. Pierwszym z nich jest wprowadzenie zaawansowanych algorytmów rankingujących **ARA** i **SRA** w celu lepszego i bardziej wiarygodnego wyznaczania ścieżek poruszania się pieszych. Kolejnym autorskim usprawnieniem było wprowadzenie do definicji automatu komórkowego i reguły przejścia pojęcia **pamięci aktora**, dzięki czemu piesi podczas symulacji, pamiętając swoje ostatnie ruchy, poruszają się bardziej płynnie i realistycznie.

Stworzone i zaprezentowane w niniejszej pracy oprogramowanie **CrowdSim** może posłużyć zarówno projektantom, architektom lub konstruktorom, jak i służbom porządkowym, ochronie danego obiektu lub organizatorom imprez masowych. Symulacje wykonane wielokrotnie dla jednego scenariusza przy różnych parametrach wejściowych uwiadcniają problemy i wąskie gardła mogące pojawić się w trakcie ewakuacji z zagrożonego obiektu. Przeprowadzanie takich symulacji na etapie projektowym może bardzo pozytywnie wpłynąć na bezpieczeństwo osób użytkujących dany obiekt jak również poprawić jego funkcjonalność.

for simulation had equally distributed exits in relation to the building area and that the actors on the map were distributed very evenly, covering the whole map. Due to this, persons consequently walking towards the nearest exit did not waste time on changing decisions as in the case of simulation with the maximum value of the density coefficient. As for the use of exits, in contrast to the first case, distribution is not so ideal (lower use at exit 3). Although the load of exits was quite even, it resulted from an even distribution of exits and actors on the map rather than from the algorithm itself.

Summary

The aim of this paper was to present and compare classic models of pedestrian dynamics simulating the process of evacuation of people from various types of rooms. As it was demonstrated, each evacuation scenario requires individual selection of methods and parameters. Unfortunately, there are no possibilities of creating a universal algorithm and selecting coefficients in such a way as to optimally describe all possible situations.

The presented simulation results lead to the important question concerning a possibility of a reliable simulation of the intelligent behaviour of people in the evacuation process with the use of computer software. The answer to such a question turns out not that simple because despite of the properly implemented **CA**, **CA-A** and **CA-B** algorithms, there is a large insufficiency in terms of reliability and realism of simulations conducted with the use of these methods. These mechanisms often turn out to be too simple and insufficient to describe the complex process of pedestrian traffic.

Another problem is posed by optimisation. Algorithms based on cell automata are not so simple to parallel and therefore, in the case of a large number of actors during the simulation, some efficiency issues can occur. Taking the aforementioned issues into account, considerable improvements in simulation algorithms were proposed in the paper. The first one is the introduction of the **ARA** and **SRA** advanced ranking algorithms to determine pedestrian paths in a better and more reliable manner. Another proprietary improvement was the introduction of the concept of **actor memory** to the cell automaton and passage rule definitions. Due to this, pedestrians move more smoothly and more realistically during the simulation by remembering their previous moves.

The **CrowdSim** software created and presented in this work can serve both designers, architects and construction engineers but also for security services or organisers of mass events. Simulations conducted multiple times for one scenario at various output parameters show the issues and bottlenecks which can occur during the evacuation from the building at risk. Such simulations conducted in the design stage can very positively affect safety of people using the building and also improve its functionality.

Opracowanie wykonano w ramach realizacji pracy statutowej S/WI/1/2013 finansowanej ze środków przeznaczonych na naukę MNiSW, prowadzonej na Wydziale Informatyki Politechniki Białostockiej.

The article was written as part of the S/WI/1/2013 statutory work financed from the funds of the Ministry of Science and Higher Education conducted at the Faculty of Computer Science, the Bialystok University of Technology.

Literatura / Literature

- [1] The 9th international conference on Pedestrian and Evacuation Dynamics August 2018 in Lund, Sweden, [dok. elektr.] <https://www.conferencemanager.dk/PED2018/event.html> [dostęp: grudzień 2018].
- [2] Cellular Automata for Research and Industry, The 13th International Conference and School – Como Italy, September 2018, [dok. elektr.] <http://acri2018.disco.unimib.it/> [dostęp: grudzień 2018].
- [3] The Conference on Traffic and Granular Flow, Washington, DC, USA, July 2017, [dok. elektr.] <https://tgf17.gwu.edu/> [dostęp: grudzień 2018].
- [4] Oasys-software, www.oasys-software.com [dostęp: grudzień 2018].
- [5] Pathfinder: Thunderhead, www.thunderheadeng.com/pathfinder [dostęp: grudzień 2018].
- [6] Exodus: Fire Safety Engineering Group, www.fseg.gre.ac.uk/exodus [dostęp: grudzień 2018].
- [7] Legion Software: Legion, www.legion.com/legion-software [dostęp: grudzień 2018].
- [8] FDS+Evac tools: Ruggiero Lovreglio, www.lovreglio.altervista.org [dostęp: grudzień 2018].
- [9] Goalem Crowd: Goalem, <http://golaem.com/content/product/goalaem> [dostęp: grudzień 2018].
- [10] Massive Software, <http://www.massivesoftware.com> [dostęp: grudzień 2018].
- [11] Miarmy Basefount, <http://www.basefount.com/miarmy.html> [dostęp: grudzień 2018].
- [12] Quadstone Paramics, <http://www.paramics-online.com/index.php> [dostęp: grudzień 2018].
- [13] VISSIM, PTV Group, <http://vision-traffic.ptvgroup.com/en-uk/products/ptv-vissim/> [dostęp: grudzień 2018].
- [14] Burstedde C.K., Klauck K., Schadschneider A., Zittartz J., *Simulation of pedestrian Dynamics using a 2-dimensional cellular automata*, „Physica A” 2001, 295, 507–525, [dok. elektr.] <https://pdfs.semanticscholar.org/8afc/062897038b24a0060b7bc2ca62d046b7a82c.pdf> [dostęp: grudzień 2018].
- [15] Klüpfel H.L., *A Cellular Automaton Model for Crowd Movement and Egress Simulation*, PhD thesis, University of Duisburg-Essen, 2003, [dok. elektr.] <https://duepublico.uni-duisburg-essen.de/servlets/DocumentServlet/Document-5477/Disskluepfel.pdf> [dostęp: grudzień 2018].
- [16] Wąs J., *Algorytmy modelowania inteligentnych zachowań dynamiki pieszych z zastosowaniem niehomogenicznych automatów komórkowych*, Praca doktorska, AGH, Kraków 2006, [dok. elektr.] <https://docplayer.pl/45682913-Akademia-gorniczo-hutnicza-im-stanislawa-staszica-w-krakowie-wydzial-elektrotechniki-automatyki-informatyki-i-elektroniki-katedra-automatyki.html> [dostęp: grudzień 2018].
- [17] Barański M., Maciak T., *Automaty komórkowe w modelowaniu ewakuacji, Bezpieczeństwo i technika pożarnicza*, BiTP Vol. 43, Issue 3, 2016, pp. 127–142, DOI: 10.12845/bitp.43.3.2016.12.
- [18] Bouvier E., Cohen E., Najman L.; *From crowd simulation to airbag employment: particle systems, a new paradigm of simulation*, “Journal of Electronic Imaging” 1997, 6, 1, s. 94–107.
- [19] Breadth first search, https://en.wikipedia.org/wiki/Breadth-first_search, [dostęp: grudzień 2018].
- [20] Dobrowolski M.; *Automaty komórkowe w modelowaniu ewakuacji ludzi*, praca dyplomowa, Politechnika Białostocka, Wydział Informatyki 2017, Promotor T. Maciak.
- [21] Unity3D: Microsoft, <https://unity3d.com/> [dostęp: grudzień 2018].

MGR INŻ. MARIUSZ DOBROWOLSKI – absolwent Wydziału Informatyki Politechniki Białostockiej. Obecnie pracuje jako programista w firmie informatycznej w Białymstoku.

DR HAB. INŻ. TADEUSZ MACIAK – profesor Politechniki Białostockiej, zatrudniony w Katedrze Mediów Cyfrowych i Grafiki Komputerowej Wydziału Informatyki Politechniki Białostockiej, emerytowany profesor Szkoły Głównej Służby Pożarniczej w Warszawie, były kierownik Zakładu Informatyki i Łączności w SGSP. Autor wielu publikacji związanych z problematyką szeroko pojętej obronności wewnętrznej kraju.

MARIUSZ DOBROWOLSKI M.SC. ENG. – Graduate of the Faculty of Computer Science at the Bialystok University of Technology. He currently works as a software developer in a Bialystok-based IT company.

TADEUSZ MACIAK D.SC. ENG. – Professor of the Bialystok University of Technology, working in the Department of Digital Media and Computer Graphics of the Faculty of Computer Science, the Bialystok University of Technology. Professor emeritus of the Main School of Fire Service (SGSP) in Warsaw, former head of the SGSP Computer Science and Communication Department. Author of many publications on internal state defense.