




A-Evac: The Evacuation Simulator for Stochastic Environment

Adam Krasuski* , and Karol Krenski, *The Main School of Fire Service, Department of Fire Safety Engineering, Slowackiego 52/54, 01-629 Warsaw, Poland*

Received: 25 November 2017/**Accepted:** 11 February 2019

Abstract. We introduce an open-source software for fire risk assessment named Aamks. This article focuses on a component of Aamks—an evacuation simulator named a-evac. A-evac models evacuation of humans in the fire environment produced by a zone fire model simulator. In the article, we discuss the probabilistic evacuation approach, automatic planning of exit routes, the interactions amongst the moving evacuees and the impact of smoke on humans. The results consist of risk values based on fractional effective dose and are presented in the form of various probability distributions and evacuation animations. The intended scope of Aamks is buildings, e.g., offices, malls, factories rather than stadiums or streets. We present the need for such software based on the current state of research and existing engineering tools in the probabilistic risk assessment domain. Then we describe a-evac and its details: geometry, path-finding, local movement, interaction with fire, and visualization. Given the above scope, the article contributes to the domain of probabilistic risk assessment by proposing: (a) stochastic approach to evacuation, (b) velocity-based model for evacuation, (c) evacuation software that interacts with fire conditions and zone fire models.

Keywords: Evacuation modeling, Stochastic simulations, Risk management, Monte Carlo

1. Introduction

There is continuous progress in understanding the fire phenomenon, its impact on the structure and the reaction of humans and safety systems. Addressing these aspects deterministically attracts many researchers, while the uncertainty associated with these problems takes less attention. Since the selection of a proper fire scenario in the assessment of the safety of buildings may have a more significant impact on the final judgment than its underlying details, the related uncertainty should be addressed more systematically.

Currently, in most countries, fire safety is based on legal regulations [29]. However, many regulations have clauses for equivalency or alternate methods based on engineering calculations. Within this group, the most typical approach for assessing the safety of the building are computational fluid dynamics (CFD) simulations with a precise choice of the input parameters for a small number of lengthy,

* Correspondence should be addressed to: Adam Krasuski, E-mail: akrasuski@sgsp.edu.pl



detailed simulations. This procedure is managed by a practitioner, based on their experience. In most cases, the final judgments are supported with calculations of model uncertainty. However, there is lack of calculations of the uncertainty related to the selection of the parameters for these fire scenarios.

There are scientific methods and models available for addressing the uncertainty related to the fire and the emergency scene as well as the computer implementations. The most common approach for the uncertainty reflection related to fire phenomenon is *risk*. We define risk as a *a combination of the probability of a fire and a quantified measure of its consequence* following [6, 59]. Within the frame of this definition, two components should be assessed in order to calculate fire risk: (a) probability that fire can start in the building, and (b) some quantified measure of its consequence.

There are many approaches to assess these components. The probabilistic methods play a prominent part in the fire risk assessment domain. However, the complexity of the domain and the cost of computations impedes the more widespread use of the real probabilistic tools. Therefore approximative methods were proposed to assess the risk in a probabilistic way.

One of the broadly applied methods is to find these values in historical fire loss databases. The building under investigation is defined by a set of parameters (i.e., building category, area, and others). Then, a query based on these predicates is run against the database. The resulted buildings met the defined criteria and had at least one fire incident. Then the probability that fire can start in a given building is calculated using various models [2, 6, 52, 62]. The same method is applied to obtain the consequences of such fires—a measure of similarity and statistical data.

The method, however, has serious shortcomings that limit their application for fire safety engineering. Increasing the number of parameters describing the building in question allows for better matching the building to the query data. However, on the other hand, it reduces the statistical significance of the results—fewer buildings in the database fit the search criteria. Definition of more parameters goes eventually to the empty result of our query. Hence, in most approaches to the problem in question, the similarity is defined based on a minimal number of parameters: building category and its floor area [62].

Such a definition of similarity measures produces serious inaccuracies in risk calculations. Figure 1 illustrates an example of such inaccuracies.

As a result, by limiting the number of parameters and averaging the values of the given category of buildings, the statistical approach assesses the risk of an abstract building rather than the specific one.

One of the methods to overcome this problem quantitatively is the application of event tree analysis (ETA) [13] or fault tree analysis (FTA) [40, 69]. A simple intuition behind these methods is perceiving the building in question not by its architecture, but as a decomposition of its fire-related components (building blocks) which affect the ignition, support and spreading of fire. Since these components are very similar in the majority of buildings, it is much easier to find the related data and keep statistical significance on the required level. Moreover, application of ETA or FTA allows for systematical identification of these compo-

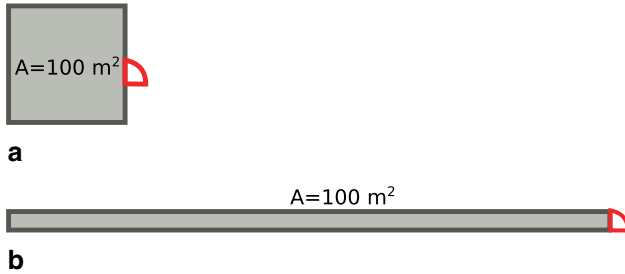


Figure 1. Two buildings with the same value of risk based on the building category and floor area. Grey rectangles illustrate projection of the building, red arcs—evacuation exits. It is clear that the real risk for occupants is greater in building b (Color figure online).

nents by logical exploration of the space of possible scenarios starting from a single event or a single failure.

Embracing FTA and ETA bring us closer to the analysis related to the specific building. The building in question is now perceived by elements that may have an impact on fire ignition and development. However, an amount of generalization remains: even though the impacting components and their uncertainties were identified, the spatio-temporal relationships among them were not.

The igniters, material supporting fire and control measures may result from the spatial configuration of the building. A fire may spread to another compartment within several minutes or a few hours. All these aspects have an impact on the fire, but FTA/ETA does not account for it.

The most advanced methods addressing this problem are known as stochastic simulations incorporating fire and evacuations models [20, 25, 65]. The other approaches to the problem in question, are either simplified state-transition [8], object-behaviour [16] or semi-quantitative models [38].

The idea of stochastic simulations is to use deterministic fire and evacuation models and the uncertain or variate input variables based on their probability distributions. The stochastic processes are mostly based on Monte Carlo method [42, 45, 63].

Risk assessment based on Monte Carlo simulations is powerful, flexible and very direct. It is simple and sometimes the only feasible way to solve a problem [45]. However, it has also shortcomings: high computational costs, lack of data defining probability distributions for input variables and lack of tools which make the process of risk assessment feasible.

The existing tools for stochastic simulations of fire risk assessment are hardly able to assess the risk for humans. For example, in [25] evaluation of the consequences of the fire are limited to the probability of occurrence of an undesired event, i.e., exceeding operational temperature for a cable. The impact of undesired events on humans is left for separate calculations. The model presented in [65] is more advanced and allows for calculating consequences for humans by fractional effective dose (FED) [50]. Consecutive simulations can then generalize the single

value of output FED into a probability distribution. However, the model lacks evacuation module, and FED is simplified by the definition *time frames—room* (position of the agent in a given room in a given time frame) tuple where occupants are expected to be.

Currently, only zone fire models (as opposed to CFD) are computationally fast enough for stochastic simulations. They are robust, well documented and readily available. The evacuation simulators are also easily available and well-trying. However, as far as we know, there is no evacuation simulator integrated with a zone fire model as a complete stochastic simulator as we describe it and we are trying to fill this gap.

In this article, we introduce Aamks—an engineering tool for fire risk assessment based on stochastic simulations. We focus on describing the evacuation module of Aamks, a-evac—an evacuation simulator under fire conditions for the stochastic environment.

Given the above goals, our work makes two main contributions:

- (a) velocity-based model for evacuation,
- (b) evacuation module that interacts with fire conditions.

2. Aamks, the Multisimulations Platform

We created Aamks—the platform for running simulations of fires and then running the evacuation simulations, but thousands of them for a single project. This is the stochastic simulation approach based on Monte Carlo method. We use CFAST, which is a zone, but a fast fire simulator. This allows us to explore the space of possible scenarios and assess the probability of them. The second component of risk—consequences—is taken from an evacuation simulator capable to model evacuation in the fire environment. We use a-evac as the evacuation simulator which we have built from scratch. The *multisimulation* is a convenient name for what we are doing. Aamks tries to assess the risk of failure of humans evacuation from a building under fire. We applied methodology proposed in [6, 11, 25]—stochastic simulations based on the Simple Monte-Carlo (SMC) approach as defined in [10, 45].

Our primary goal was to bridge the gap between scientific achievements in fire safety engineering as well as computer science and available engineering tools. This resulted in: AutoCAD plugin for creating geometries, web-based interface, predefined setups of materials and distributions of various aspects of buildings features. The workflow is as follows: The user draws a building layout or exports an existing one. Next, the user defines a few parameters including the type of the building, the safety systems in the building and others. The user can also alter the default parameters of variable distributions. Finally, they launch a defined number of stochastic simulations. As a result, they obtain the distributions of the safety parameters, namely: available safe egress time (ASET), required safe egress time

(RSET), fractional effective dose (FED), hot layer height and temperature and F-N curves as well as the event tree and risk matrix.

We code Aamks in python and javascript which are modern and comfortable languages due to a rich collection of libraries. We decided that borrowing FDS's Evac Fortran code and integrating it with Aamks would be harder for us than to code our evacuation simulator. Hence we decided to create our own python based evacuation module – a-evac. There's also a higher chance of attracting new python/javascript developers than Fortran developers for our project, based on the popularity of the above three languages.¹

Aamks consists of the following modules:

- a-geom, geometry processing: AutoCAD plugin, importing geometry, extracting topology of the building, navigating in the building and others.
- a-evac, directing evacuees across the building and altering their states
- a-fire, CFAST binaries and processing of their outputs
- a-gui, web application for user's input and for the results visualisation
- a-montecarlo, stochastic producer of thousands of input files for CFAST and a-evac
- a-results, post-processing the results and creating the content for reports,
- a-manager, managing computations on the grid/cluster of computers
- a-installer

3. A-Evac, the Evacuation Simulator

In the following subsections we describe the assumptions for a-evac, sometimes with the necessary Aamks context.

3.1. Geometry of the Environment

The Aamks workflow starts with a 3D geometry where fires and evacuations will be simulated. We need to represent the building, which contains one or more floors. Each floor can consist of compartments and openings in them, named COMPAS

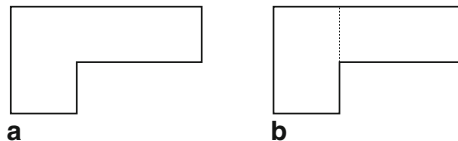


Figure 2. The concept of a HOLE: (a) the room in reality, (b) the room representation in CFAST: two rectangles for separated calculations, but open to each other via HOLE.

¹ <https://insights.stackoverflow.com/survey/2018/>.

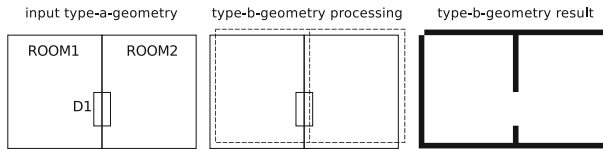


Figure 3. The conversion from A-areas to walkable areas.

and VENTS respectively in CFAST. Our considerations are narrowed to rectangular geometries (curved areas need to be approximated by rectangles). There are two basic ways of representing architecture geometries: (a) cuboids can define the insides of the rooms (A-areas) or (b) cuboids can define the walls/obstacles (walkable areas). CFAST uses the A-areas. We create CFAST geometries from the input files of the following format (there are more entities than presented here):

```
{
  "FLOOR 1":
  {
    "ROOM": [
      [ [ 3.0 , 4.8 , 0.0 ] , [ 4.8 , 6.5 , 3.0 ] ] ,
      [ [ 3.0 , 6.5 , 0.0 ] , [ 6.8 , 7.4 , 3.0 ] ]
    ] ,
    "COR": [
      [ [ 6.2 , 0.2 , 0.0 ] , [ 7.6 , 4.8 , 3.0 ] ]
    ] ,
    "D": [
      [ [ 3.9 , 3.4 , 0.0 ] , [ 4.8 , 3.4 , 2.0 ] ]
    ] ,
    "W": [
      [ [ 1.2 , 3.4 , 1.0 ] , [ 2.2 , 3.4 , 2.0 ] ]
    ] ,
    "HOLE": [
      [ [ 3.0 , 6.5 , 0.0 ] , [ 4.8 , 6.5 , 3.0 ] ]
    ]
  }
}
```

ROOM and COR(RIDOR) belong to COMPAS. D(OOR), W(INDOW) and HOLE belong to VENTS. HOLE is a result of CFAST restrictions [46, 68] – it is an artificial entity which serves to merge two compartments into a single compartment as shown on Fig. 2.

All the entities in the example belong to the same FLOOR 1. The triplets are (x_0, y_0, z_0) and (x_1, y_1, z_1) encoding the beginning and the end of each entity in 3D space. In practice, we obtain these input files from AutoCAD, thanks to our plugin which extracts data from AutoCAD drawing. In order to have a fully open source, platform independent solution we are working on an AutoCAD alternative (in Aamks context), a web-based graphics authoring tool.²

In later sections, we will introduce problems of guiding evacuees throughout the building. Those modules require the walkable areas. We convert from an A-areas

² <https://aamks.szach.in/?node=demo>.

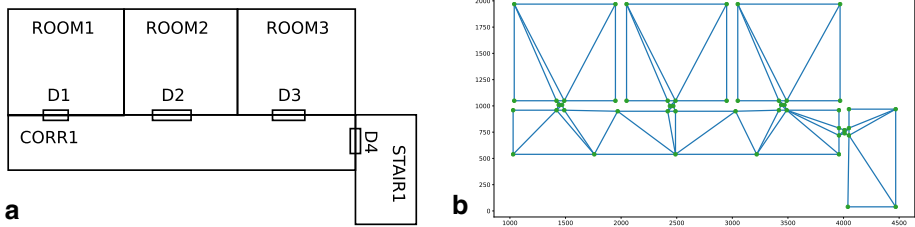


Figure 4. The idea of triangulation. (a) Original geometry, (b) triangulation.

to walkable areas by duplicating the geometry, translating the geometry and applying some logical operations. Figure 3 shows the idea.

There are three aspects of movement when it comes to evacuation modeling [66]: (a) path-finding—for the rough route out of the building, (b) local movement—evacuees interactions with other evacuees, with obstacles and with environment, and (c) locomotion—for “internal” movement of the agent (e.g. body sway). A-evac models only (a) and (b).

3.2. Path-Finding (Roadmap)

The simulated evacuees need to be guided out of the building. The walkable areas provide the input for path-finding. The coordinates define each of the cuboids—representing obstacles—in walkable areas. These coordinates represent corners of the shapes. Since we model each of the floors of a building separately, we flatten 3D geometry into 2D and represent obstacles as rectangles. Therefore walkable areas in path-finding module is represented as set of 4-tuple coordinates $[(x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3)]$.

The set of 4-tuple elements is then flattened to the set of coordinates—bag-of-coordinates. Since the majority of the obstacles share the coordinates, we remove duplicates from the set (for the sake of performance). Then, this bag-of-coordinates is the input for triangulation. We apply algorithm for triangulation presented in [43] and implemented as a Blender³ library. The triangulation represents space as a set of triangles. Figure 4 depicts the idea of triangulation.

The triangles are used as navigation meshes for the agents. The navigation meshes define which areas of an environment are traversable by agents.

After triangulation of bag-of-coordinates, some of the triangles are located inside the obstacles – those (by definition) are not traversable, so we remove them. What is left is a traversable-friendly space.

We create then the graph of spatial transitions for the agents, based on the adjacency of triangles obtained from the triangulation. Spatial transition means that an agent can move from one triangle to another.

³ <https://www.blender.org/>.

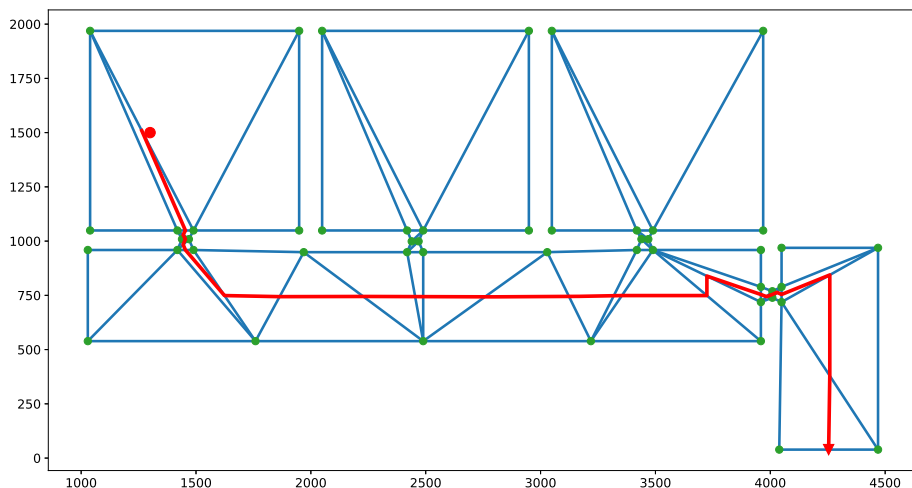


Figure 5. The roadmap defined by the graph for an example query. The red line crosses centers of edges that an agent needs to travel to reach the exit (Color figure online).

An agent on the edge of a triangle can always reach the other two edges. For triangles which share edges, it allows an agent to travel from one triangle to another.

The pairs of all adjacent edges are collected. We use python networkx module [5] which creates a graph made of the above pairs. For further processing, we add agents positions to the graph, by pairing them with the adjacent edges.

The graph represents all possible routes from any node to any other node in the graph. We can query the graph for the route from the current agent's position to the closest exit. It means that an agent will walk through the connected nodes and will finally reach the exit door. We instruct networkx that we need the shortest distances in our routes (default is the least hops on the graph) and we obtain a set of edges the agent should traverse in order to reach the exit. Figure 5 depicts the set of edges returned by the graph for an example query.

The set of edges returned by the graph cannot be used directly for path-finding. Neither the vertices of the edges nor the centers of them, do define the optimal path that would be naturally chosen by evacuees during a real evacuation. Therefore an additional algorithm should be used to smooth the path. For this purpose, we apply the funnel algorithm defined in [9]. The funnel is a simple algorithm finding straight lines along the edges.

The input for the funnel consists of a set of ordered edges (named portals) from the agent origin to the destination. The funnel always consists of 3 entities: the origin (apex) and the two vectors from apex to vertices on edges—the left leg and the right leg.

The apex is first set to the origin of the agent, and the legs are set to the vertices of the first edge. We advance the left and right legs to the consecutive edges

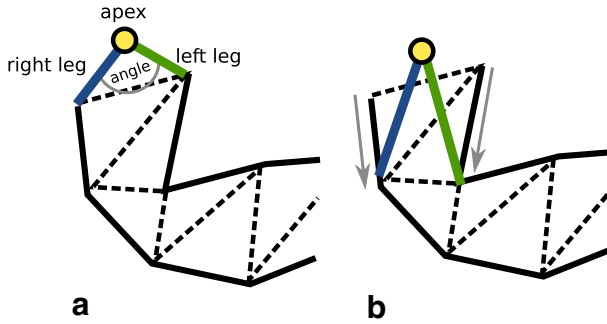


Figure 6. The idea of the funnel algorithm.

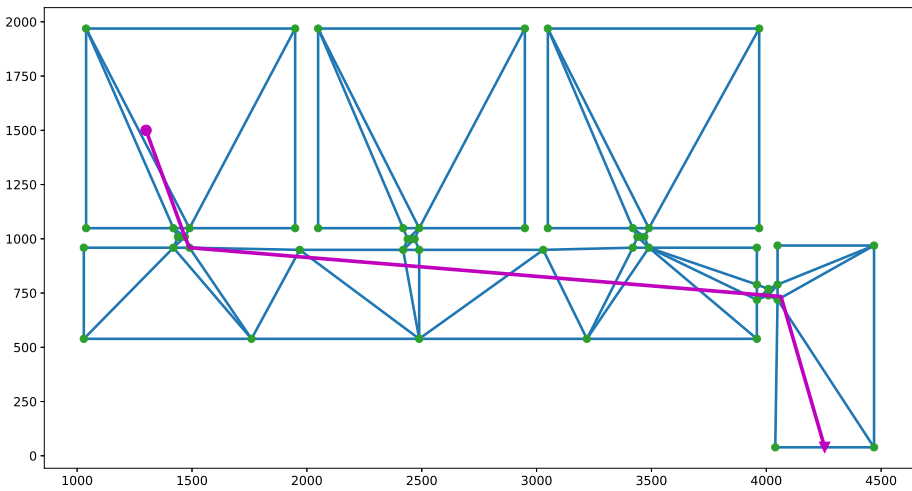


Figure 7. The roadmap from starting point to exit smoothed by funnel algorithm.

in the set and observe the angle between the legs. When the angle gets smaller, we accept the new vertex for the leg. Otherwise, the leg stays at the given vertex. After some iteration one of the legs should cross the other leg defining the new position of the apex. The apex is moved, and we restart the procedure (Fig. 6).

As a result, the path is smoothed and defined only by the points where the changes in velocity vector are needed. Moreover, we used an improved version of the funnel algorithm that allows for defining points keeping a distance from the corners reflecting the size of the evacuee. What allows for modeling the impaired evacuees on wheeled chairs or beds in the hospitals. Figure 7 depicts the smoothed path by funnel algorithm.

3.3. Local Movement

The local movement focuses on the interaction with (a) other agents (b) static obstacles (walls) and (c) environmental conditions. There are two approaches to local movement: (a) force-based models and (b) velocity-based models. Force-based models represent humans as particles and model their interactions using physical forces.

Two of the most popular force-based methods are boids model proposed by Reynolds [53] and Helbing's et al. social force model [22]. Reynolds' model captures flocking behavior using separation, alignment, and cohesion forces, while the model of Helbing's uses a mixture of sociological and physical forces to describe pedestrian interactions.

In both approaches, the forces depend only on the separation of the agents and can lead to simulation artifacts such as oscillations and backward movement, but there are ways to prevent these issues [36, 37, 47, 54, 56].

The main advantage of all the above force-based approaches is the simplicity in their formulation. However, they often require careful parameter tuning to generate desired simulation results. Besides, they suffer from numerical stability issues, since forces can take large values and vary quickly. Currently, most of the available evacuation software uses force-based models.

As such, velocity-based models work directly in the velocity space by selecting at each simulation step a new velocity for each agent according to a given cost function. Numerous velocity-based formulations have been proposed in the past decade, including time-to-collision approaches [3], minimal predicted distance techniques [44, 48], as well as geometrical methods based on linear programming [64].

Recently, vision-based approaches were proposed that can more closely match human behaviour [12, 27] as well as approaches that account for the holonomic nature of the human motion [26, 57], non-linear motions [67], and behavioral realism [35].

Due to instabilities of force-based models—what is crucial for stochastic multi-variate and often large number of unrevised simulations—we decided that our model will be velocity-based. We also excluded other recent approaches due to the lack of availability of well-tried software libraries.

A-evac handles collision avoidance with agents and obstacles via RVO2⁴ which is an implementation of the time-to-collision approaches based on linear programming [64]. Later in this next section, we describe how we model the third aspect of local movement—interaction with the environment.

RVO2 aims at avoiding the *velocity obstacle* [14]. The velocity obstacle is the set of all velocities of an agent that will result in a collision with another agent or an obstacle. Otherwise, the velocity is collision-avoiding. RVO2 aims at asserting that none of the agents collides with other agents within time τ .

The overall approach is as follows: each of the agents is aware of other agents parameters: their position, velocity, and radius (agent's observable universe). Besides, the agents have their private parameters: maximum speed and a preferred velocity which they can auto-adjust granted there is no other agent or an obstacle colliding. With each loop iteration, each agent responses to what he finds in his/

⁴ <http://gamma.cs.unc.edu/RVO2/>.

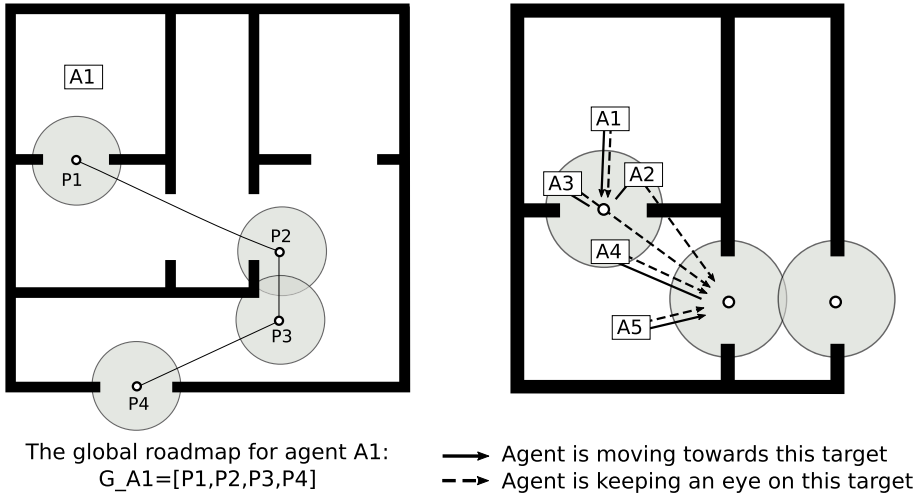


Figure 8. The roadmap and local movement.

her surroundings, i.e., his/her own and other agents radiuses, positions and velocities. The agent updates their velocity if it is the velocity obstacle with another agent. For each pair of colliding agents, the set of collision-avoiding velocities is calculated. RVO2 finds the smallest change required to avert collision within time τ and that is how an agent gets his/her new velocity. The agent alters up to half of his/her velocity while the other colliding agent is required to take care of his/her half. Figure 9a, b depicts the idea of velocity collision avoidance.

The algorithm remains the same for avoiding static obstacles. However, the value of τ is smaller concerning obstacles as agents should be more 'brave' to move towards an obstacle if this is necessary to avoid other agents.

It turned out problematic how to pick the local target from the roadmap. Local targets need to be updated (usually advanced, but not always) near points defined by the funnel algorithm during path-finding phase – the *disks* on Fig. 8, after they become visible to the agent. However, the disks can be crowded, and agents can be driven away from the correct courses by other agents. We carefully inspected all possible states that agents can find themselves in. In order to have a clearer insight and control over the agents inside the disks, we use the Finite State Machine instead of just plain algorithm block in our code. Four binary features define the state of the agent: (a) is agent inside the disk? (b) are agent's walking target, and an agent is observing target the same thing? (c) can agent see what he is looking at (or are there obstacles in-between)? (d) has agent reached the final node?

Within each iteration of the main loop, we check the states of the agents. Agents themselves can change the states—e.g., an agent has crossed the border of the disk, or by our commands—e.g., the agent is ordered to walk to another target. Consider these circumstances: the agent has managed to see his/her next target, and now he walks towards this next target—he is in state S1. However, now he loses the eye contact with this new target and finds himself in state S2. The

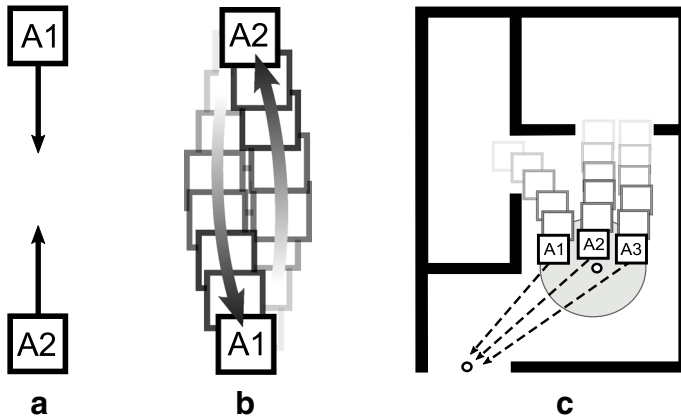


Figure 9. RVO2 at its work of resolving collisions: (a) agents on direct collision courses and (b) their calculated collision-avoiding courses, (c) three agents crossing a HOLE in parallel.

program logic reacts to such a state by transiting to the state S3: start looking at the previous target and walk towards this previous target. Based on what happens next we can order the transition to another state or wait for the agent to change the state himself. By careful examination of all possible circumstances, we can make sure that our states and their transitions can handle all possible scenarios.

On Fig. 9c we show how agents are passing through a HOLE. Due to our concept of the disks (where searching for new targets takes place) and due to the internals of RVO2 we gain the desired effect of agents not crossing the very center of the disk. Instead, the agents can walk in parallel and advance to another target which looks natural and doesn't create an unnecessary queue of agents eager to cross the very center of the HOLE.

3.4. Evacuation Under Fire and Smoke

Fire simulations precede each a-evac simulation. We have only tested a-evac with CFAST [34, 46]. CFAST writes its output to CSV files. We need to query these CFAST results quite a bit. Therefore we transform and store these results in a fast in-memory relational database.⁵ For each frame of time we are repeatedly asking the same questions: (a) given the agent's coordinates, in which room are they? (b) what are the current conditions in this room?

When it comes to (b), the environmental effects on the agent can be (b.1) limited visibility (eyes), (b.2) poisonous gases (nose) and (b.3) temperature in the room (body). Both (b.1) and (b.2) are read from the default (but configurable) height of 1.8 m. There are always two zones in CFAST, which are separated at a known height, so we need to read the conditions from the correct zone, based on where our 1.8 m belongs.

⁵ <https://www.sqlite.org/>.

The value of visibility (*OD*—optical density) affects agent’s speed. We use the relation proposed in [15] following the FDS + Evac [39]:

$$v_n^{pref}(K_s) = \max \left\{ v_{n,min}, v_n^{pref} \left(1 + \frac{\beta}{\alpha} \cdot K_s \right) \right\} \tag{1}$$

where: K_s is the extinction coefficient ($[K_s] = m^{-1}$) calculated as $OD/\log_{10}e$ according to [32, 33], $v_{n,min}$ is the minimum speed of the agent A_n and equals $0.1 \cdot v_n^{pref}$ (agent’s preferable velocity), and α, β are the coefficients defined in [15].

Setting the minimal value of speed means that the agent does not stop in thick smoke. They continue moving until the value of incapacitated Fractional Effective Dose (FED) is exceeded, which is fatal to the agent. FED is calculated from CFAST-provided amounts of the following species in the agent environment: carbon monoxide (CO), hydrogen cyanide (HCN), hydrogen chloride (HCl), carbon dioxide (CO_2) and oxygen (O_2) by the equation [39, 50]:

$$FED_{total} = (FED_{CO} + FED_{HCN} + FED_{HCl}) \times HV_{CO_2} + FED_{O_2} \tag{2}$$

where HV_{CO_2} is the hyperventilation induced by the concentration of CO_2 . Following are the formulas for the terms in the above equation. FEDs are given in ppm and time t in minutes. C stands for concentration of the species in volumetric % [28]:

$$FED_{CO} = \int_0^t 2.764 \times 10^{-5} (C_{CO}(t))^{1.036} dt \tag{3}$$

$$FED_{HCN} = \int_0^t \frac{\exp\left(\frac{C_{HCN}(t)}{43}\right)}{220} - 0.0045 dt \tag{4}$$

In contrast to the model applied in FDS + Evac, CFAST does not allow for proactive correction of effect of nitrogen dioxide— $C_{CN} = C_{HCN} - C_{NO_2}$. Therefore this effect is not included in the calculations.

$$FED_{HCl} = \int_0^t \frac{C_{HCl}(t)}{1900} dt \tag{5}$$

Based on [28, 58]

$$FED_{O_2} = \int_0^t \frac{dt}{60 \cdot \exp[8.13 - 0.54(20.9 - C_{O_2}(t))]} \tag{6}$$

$$HV_{CO_2} = \frac{\exp(0.1903 \cdot C_{CO_2}(t) + 2.0004)}{7.1} \tag{7}$$

Table 1
FED Effects on Human Health in Aamks

| FED | Effect on human health |
|--------------|---|
| < 0.01 | Minor or negligible |
| [0.01 – 0.3[| Low—short period of hospitalization |
| [0.3 – 1[| Heavy—lingering health problems or permanent disability |
| ≥ 1 | Lethal |

The general idea of FED (proposed by Purser) is an evaluation of the consequences of inhalation of toxic product in fire atmosphere by evacuees [50]. The emphasis was put at what point in the time during the fire exposure, the evacuee will have inhaled a toxic dose. This time is then related to the tenability criteria. The value of FED is calculated by integrating the area under fire profile curve for toxicant under consideration. When the integral is equal to toxic dose, the incapacitating or lethal effect is assumed. Thus, the FED equation can be represented as dose received at the time divided by an effective dose to cause incapacitation or death [21]. This way, the original concept does not consider the sublethal effect of exposure. However, in order to have a broader spectrum of consequences in risk assessment, we decided to consider also sublethal effects.

There are few quantitative data from controlled experiments concerning the sublethal effect of the smoke on people. In works [7, 18, 50, 58, 60] sublethal effect in a form of incapacitation (IC_{50}), escape ability (EC_{50}), *lingering health problems* and *minor effects* were reported. Incapacitation was inferred from lethality data, to be about one-third to one-half of those required for lethality. The mean value of the ratios of the IC_{50} to the LC_{50} and the standard deviation were 0.50 and 0.21, respectively. In [18] a scale for effects based on FED was introduced. The three ranges were proposed: 1 FED indicating lethality, 0.3 FED indicating incapacitation and 0.01 FED indicating no significant sublethal effects should occur. We propose based on this data a scale for sublethal effects of smoke for evacuees as presented in Table 1.

FED_{total} affects the agent's movement in the smoke. For $FED_{total} > 0.3$, the smoke inhalation leads to sublethal effects [7]—the agent is not able to find safety from the fire and stays where he is. For $FED_{total} > 1$ we model lethal effects. We later use these effects in the final risk assessment.

Table 1 summarizes the FED effects on human health, what is our original proposition for evaluation of the sublethal effect of smoke. These ranges are incorporated in Aamks. It is based on the following works: [7, 18, 50, 58, 60]

3.5. Probabilistic Evacuation Modeling

This section presents the internals of our probabilistic evacuation model, which we find distinct across the available, similar software.

Table 2 presents the distributions of the input parameters used in Aamks. Each of the thousands of simulations in a single project is initialized with some random

Table 2
Parameters of the Distributions for the Exemplary Scenario

| | Parameter | Distribution | μ/min | σ/max |
|-----|--|--------------|------------------|---------------------|
| 1. | Density in rooms [m^2/humans] | Normal | 5 | 2 |
| 2. | Density on corridors [m^2/humans] | Normal | 20 | 3 |
| 3. | Density in stairways [m^2/humans] | Normal | 50 | 3 |
| 4. | Human location in the compartment x | Uniform | 0 | Room width |
| 5. | Human location in the compartment y | Uniform | 0 | Room depth |
| 6. | Time of the alarm | Log-normal | 0.7 | 0.2 |
| 7. | Pre-evacuation time in the room of fire origin | Uniform | 0 | 30 |
| 8. | Pre-evacuation time in other compartments | Log-normal | 3.04 | 0.142 |
| 9. | Horizontal speed | Normal | 1.2 | 0.2 |
| 10. | Vertical speed | Normal | 0.7 | 0.2 |
| 11. | α for speed in the smoke | Normal | 0.706 | 0.069 |
| 12. | β for speed in the smoke | Normal | -0.057 | 0.015 |
| 13. | Humans taking an alternative evacuation route | Binomial | 0.03 | 0.97 |



Figure 10. The results of evacuation modeling as F-N curves.

input setup according to these distributions. Aamks has a library of the default parameters values for important building categories (schools, offices, malls, and many others). The Aamks users should find it convenient to have all the distributions in a library, but they may choose to alter these values, which is possible.

Since A-evac primary goal is stochastic simulations, most of the data in Table 2 are probability distributions, and come from the standards and research papers. The minority of them (deterministic) come from other models, mostly FDS+E-vac. Following are some comments on Table 2.

Aamks puts much attention to the pre-evacuation time [66], which models how people lag before evacuating after the alarm has sounded. Positions 7. and 8. in Table 2 are separated, because the behavior of humans in the room of fire origin is distinct. We compile two regulations *C/VM2 Verification Method: Framework for Fire Safety Design* [61] and *British Standard PD 7974-6:2004* [6] in order to get the most realistic, probability-based pre-evacuation in the room of fire origin and the rest of the rooms. These regulations are the defaults in Aamks. However, those can be altered by the users.

The Horizontal/Vertical speed (unimpeded, walking speed of an agent) is based on [17, 23, 30, 49].

Formula 1 defines the speed in the smoke.

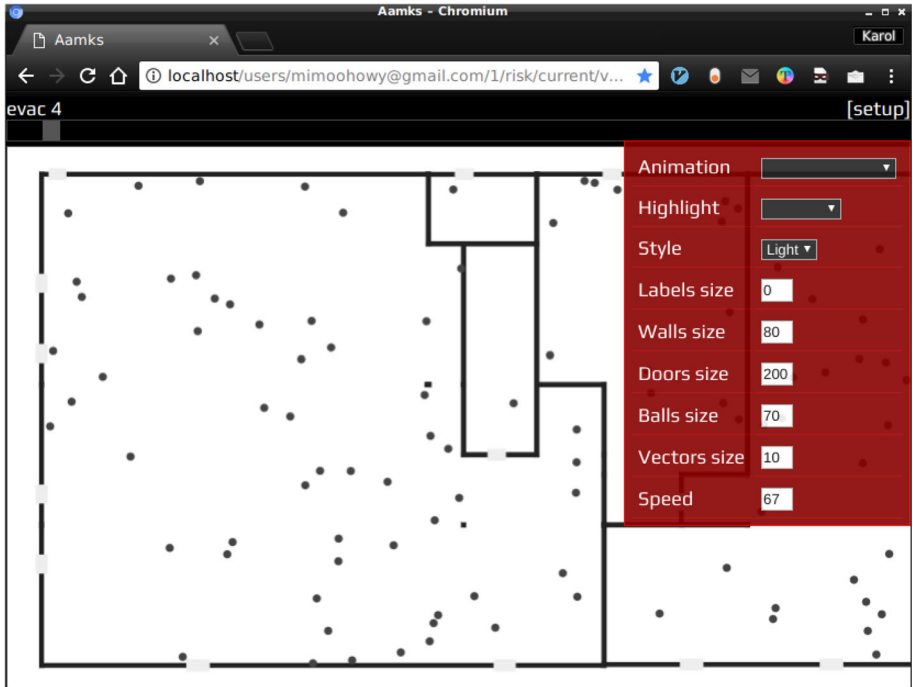


Figure 11. 2D visualization: animation of evacuees.

The randomness of the simulations comes from the random number generator's *seed*. We save the seed for each simulation so that we can repeat the very same simulation, which is useful for debugging and visualization.

We register all the random input setups and the corresponding results in the database. We are expecting to research at some point the relationships in these data with data mining or sensitivity analysis.

The final result of Aamks is the compilation of multiple simulations as a set of distributions, i.e. F-N curves. The F-N curves were created as in [15]. Figure 10 depicts the exemplary results.

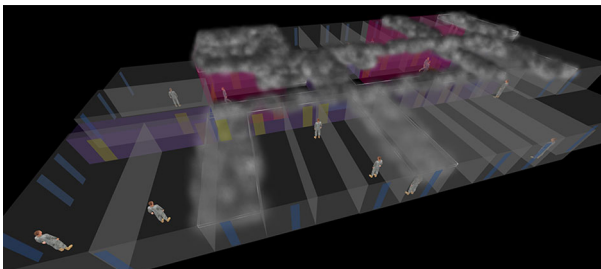


Figure 12. 3D visualization.

Due to the application of SMC approach [45], there is no need to predefine the number of simulations before the process of calculation starts—contrary to for example Latin Hypercube Sampling. Each next simulation in the stochastic process shrinks the error of the approximation. Hence, users can stop the simulation process at any moment obtaining the given value of an error as discussed in Sect. 4.1. Therefore the uncertainty related to the data presented on Fig. 10 are strictly related to the number of simulations performed.

However, our recommended practice is to perform at least such many simulations for which the resulted significance intervals prevent switching to another risk category defined by SFPE risk matrix [19].

3.6. Visualization

In Aamks we use a 2D visualization for supervising the potential user's faults in his/her CAD work (e.g., rooms with no doors Fig. 11), for the final results, and our internal developing needs. We use a web-based technology which allows for displaying both static images and the animations of evacuees.

We also have a web-based 3D visualization made with WebGL `Three.js`. This subsystem displays realistic animations of humans during their evacuation under fire and smoke. (Fig. 12).

4. Quality and the Performance of A-Evac

Below we evaluate the quality of a-evac as described in [55, 66] as well as its computer performance.

4.1. Verification of A-Evac

Verification and validation deal with how close the results of the simulations to the reality are. We took care to be compliant with the general development recommendations [66] by (1) obeying good programming practices, (2) verifying intermediate simulation outputs, (3) comparing simulation outputs against the analytical results, and (4) creating debugging animations.

There are three types of errors that can be generated by our software: (a) error in deterministic modeling of a single scenario, (b) error of Monte Carlo approximation, (c) distributions related errors.

For the first type of error, we applied the methods proposed in [55]. The proposed tests are organized in five core components: (1) pre-evacuation time, (2) movement and navigation, (3) exit usage, (4) route availability, and (5) flow conditions/constraints. For each category, there are detailed tests for the geometry, the scenario, and the expected results. The results are in Table 3 and visualized on the project's web page.⁶ We also performed validation against real building fire evacuation scenarios, and we compared the results with Pathfinder.⁷ The results are also presented on the project's web page.

⁶ <https://aamks.szach.in/?node=tests>.

⁷ <https://www.thunderheadeng.com/pathfinder>.

Table 3
The Results of Aamks Tests

| Id. | Name of the tests | Test code | Results |
|-----|-------------------------------------|------------|---------|
| 1. | Pre-evacuation time distributions | Verif.1.1 | OK |
| 2. | Speed in a corridor | Verif.2.1 | OK |
| 3. | Speed on stairs | Verif.2.2 | – |
| 4. | Movement around the corner | Verif.2.3 | OK |
| 5. | Assigned occupant demographics | Verif.2.4 | OK |
| 6. | Reduced visibility vs walking speed | Verif.2.5 | OK |
| 7. | Occupant incapacitation | Verif.2.6 | OK |
| 8. | Elevator usage | Verif.2.7 | – |
| 9. | Horizontal counter-flows (rooms) | Verif.2.8 | OK |
| 10. | Group behaviours | Verif.2.9 | – |
| 11. | People with movement disabilities | Verif.2.10 | – |
| 12. | Exit route allocation | Verif.3.1 | OK |
| 13. | Social influence | Verif.3.2 | – |
| 14. | Affiliation | Verif.3.3 | – |
| 15. | Dynamic availability of exits | Verif.4.1 | OK |
| 16. | Congestion | Verif.5.1 | OK |
| 17. | Maximum flow rates | Verif.5.2 | – |

RVO2, the core library of a-evac which drives the local movement, was evaluated in [37], where the various inadequacy to real evacuation scenarios were reported. The tests on Aamks website also include the comparison of RVO2 against Helbing's [22] and Karamouzas' models [36]. The results of these tests show that RVO2 can hardly meet density-speed criteria defined by Zhang and Seyfried experiment for bidirectional flow [70]. The comparisons show that for various scenarios RVO2 may be too much or too little efficient. We were also unable to obtain Dynamic Multi-Line phenomenon for RVO2. However, Helbing's and Karamouzas's model tests revealed the instabilities reported in the literature. The stability was crucial for us, while a-evac is designed for performing thousands of simulations with random parameters and unreviewed results. Karamouzas's model also needs more computer power what results in longer simulation time. The conclusions are that RVO2 is of the quality comparable with the lattice gas models [24, 31, 41].

The second type of error is related to the compilation of the whole collection of such single simulations—this is how we get the big picture of the safety of the inquired building. The picture is meant to present a risk. A probability of risk is calculated as a share of simulations resulted in fatalities, in the total number of simulations.

The accuracy of this evaluation depends on the method applied—stochastic simulations. In order to evaluate the error of the SMC approximation, let us assume that random variable Y (the expected value $\hat{\mu}_n$ we approximate) has a finite variance $Var(Y) = \sigma^2 < \infty$. In independent identical sampling, $\hat{\mu}_n$ is also a random variable and it has a mean and variance. The mean of $\hat{\mu}_n$ can be defined as follows [4]:

$$\mathbb{E}(\hat{\mu}_n) = \frac{1}{n} \sum_{i=1}^n \mathbb{E}(Y_i) = \mu \tag{8}$$

where: n is the sample size. The expected value of $\hat{\mu}_n$ equals the population mean μ . This means that SMC is unbiased estimator of μ . Then the following formula can be used to calculate the variance of $\hat{\mu}_n$ [45]:

$$\mathbb{E}((\mu - \hat{\mu}_n)^2) = \frac{\sigma^2}{n} \tag{9}$$

Equation (9) states that the expected value of the squared difference between our estimator $\hat{\mu}_n$ and μ is a function of the variance of original distribution Y and the sample size n . Hence, the error of SMC approximation, is directly proportional to the variance of Y and inversely proportional to the sample size. The root mean squared error (RMSE) of $\hat{\mu}_n$ can be then calculated as follows:

$$\text{RMSE} = \sqrt{\mathbb{E}((\mu - \hat{\mu}_n)^2)} = \frac{\sigma}{\sqrt{n}} \tag{10}$$

We can use then, a standard deviation from the sample values $\hat{\sigma}_n$ as the estimator of σ [45]:

$$\hat{\sigma}_n^2 = \frac{1}{n-1} \sum_{i=1}^n (Y_i - \hat{\mu}_n)^2 \tag{11}$$

From the Central Limit Theorem, we know that $\hat{\mu}_n - \mu$ has approximately a normal distribution with mean 0 and variance σ^2/n [4]. Hence in order to calculate confidence intervals, we take the value of standard normal quantile and multiply them by factor $\hat{\sigma}_n/\sqrt{n}$. For the 95% confidence the formula will be as follows:

$$\hat{\mu}_n \pm 1.96 \times \hat{\sigma}_n/\sqrt{n} \tag{12}$$

For example, in $n = 1000$ simulations we obtain a probability of scenario with fatalities $\hat{p}_n = 0.01$. The probability of fatalities is the discrete Bernoulli probability distributions with variance $\text{Var}(Y) = p(1 - p)$. Then the uncertainty related to this estimation is calculated as follows:

$$\hat{p}_n \pm 1.96 \sqrt{\frac{\hat{p}_n(1 - \hat{p}_n)}{n}} = 0.01 \pm 1.96 \sqrt{\frac{0.01(1 - 0.01)}{1000}} = 0.01 \pm 0.0062; \tag{13}$$

The third type of error relates to the distributions. For instance, the open/close state of a window (which has an impact on fire development) is a subject to a distribution based on the outside temperature, time of the day and others related weather. Aamks uses dozens of such distributions, some of which are our rough estimates since there is no research on them. Another problem is the dependencies

Table 4
The Costs of a Single Loop Iteration per Agent

| Activity | Time (s) | Total share (%) |
|-----------------|-----------------------|-----------------|
| Position update | 8.12×10^{-6} | 3.97 |
| Velocity update | 1.07×10^{-5} | 5.26 |
| Speed update | 8.50×10^{-5} | 41.63 |
| State update | 7.75×10^{-8} | 0.03 |
| Goals update | 9.25×10^{-8} | 0.04 |
| FED update | 1.01×10^{-4} | 48.98 |
| Time update | 1.22×10^{-7} | 0.06 |

of the distributions: perhaps the evacuees with long pre-evacuation times should have higher velocities as at some point they should be witnessing the danger. Also, some parameters (e.g., evacuee choosing the exit route) may be a subject to distribution, and currently it is not.

These problems may be evaluated by comparison of data generated by Aamks with real statistics. So far we do not have an idea how to tackle this problem efficiently. We consider evaluating this error by launching simulations for the building stock and check whether we reconstruct historical data. This method is very laborious and not justified at that moment because our application still lacks some models, i.e. fire service intervention which has a significant impact on the fire.

4.2. Performance of the Model

The main loop of Aamks processes all agents in the time iteration. Table 4 summarizes how costly the specific calculations for a single agent within a single time iteration are. The tests were performed on the computer with Intel Core i5-2500K CPU at 3.30 GHz with 8 GB of RAM.

The total time of a single step of the simulation for one agent is 2×10^{-4} s and it grows linearly with the number of agents. The speed and FED calculations are most costly because they both make database queries against the fire conditions in the compartment. The time step for a-evac iteration is 0.05 s. There is no significant change in fire conditions within this time frame. Therefore for performance optimization, we update speed and FED every 20-th step of the simulation.

5. Discussion

Aamks has been actively developed since 2016. At this stage, we want to come up with a complete platform that can be launched by the users, produce reasonable results and then we will be progressively improving the details, like adding vertical evacuation, which is years of work.

We are aware that detailed technical documentation is needed before Aamks can actually be deployed and used reliably.

Vertical evacuation is troublesome and not implemented. There is RVO2 3D, but it is for aviation where agents can pass above each other—apparently not for our needs. Besides, we think that 2D is more readable, by clear presentation of a selected floor. We like the idea that vertical evacuation can be still considered 2D, just rotated, and we plan to move in this direction.

Currently, Aamks uses the shortest path metric to direct evacuees to the exits. We are planning to add the mechanism for the evacuees to choose other routes, e.g., evacuees may not know the full topology. Aamks does not model the social or group behaviors. However, it is difficult to evaluate, how much the lack of such functionalities impacts the resulting probability distributions.

Aamks has read-only access to CFAST, which means we cannot modify the CFAST state once the simulation has started. We randomly choose which doors are open/close at the beginning of each simulation, and we leave these doors in this position no matter whether people pass through this door. We are not aware of scientific researches for how often evacuees close the door after they pass through them. Surely, in reality, the door needs to be open for at least the time an evacuee passes through them and door may be closed again, but currently, Aamks does not alter doors positions.

There seems to be lot's of space for improvement in Aamks. We work with the practitioners and know the reality of fire engineering. We know the limitations of our current implementations and most of them can be addressed—there are models and approaches that we can implement, and the major obstacle is the limits of our team resources. Therefore we invite everyone interested in joining our project at <http://github.com/aamks>.

We are aware and take it seriously that things need to be tested. Before Aamks is indeed used in production, it needs to be tested, verified and documented. We have put Aamks through the NIST 1822 test suite [55] and Zhang and Seyfried [70] test. We pretty much failed at Zhang and Seyfried's test, since this is how RVO2 works. At some point we may replace RVO2 with another module—there are quite a few open source implementations of crowd simulators, it is not hard, but just not high on our priority list, since it is just one of many aspects of Aamks framework. When it comes to the NIST 1822 test suite, there are some assumptions that the simulations are deterministic (e.g., agents initial placement is strictly defined, the exit door is defined strictly for each agent) which caused us trouble with performing the tests. In the end, it is difficult to judge on some aspects. We already decided that Aamks should not be only probabilistic and allow for defining deterministic parameters too. Once we introduce the determinism, we will update the tests results. For now, we publish the current test results and videos.⁸ Also, on the same website we compare Aamks against Pathfinder, but it just a rough overview, because of the difficulty of setting the very same initial parameters for the two.

⁸ <https://aamks.szach.in/?node=tests>.

6. Conclusion

The problem we are solving is to provide an easy to use, working solution for the risk assessment available as an open source software. Our work makes two contributions: (a) velocity-based model for evacuation and (b) evacuation module interacting with the fire conditions.

So far the project has been well received by practitioners and academics. The stochastic-based workflow of Aamks is not a new concept, but since no wide-used implementation has been created so far, this is an additional motivation that drives our project. There are opinions in the community that stochastic-based approach is how fire engineering should be done.

We develop Aamks framework in order to support the performance-based design approach with probabilistic tools. In this article, we discussed only the performance criteria associated with the prevention of incapacitation by fire products. Currently, we are working on performance criteria that are based on thermal damage. A methodology similar to that used with fire products can be applied [51].

We are aware that currently, providing sufficient visibility is the main factor affecting design acceptability. We address this issue by the probability that $ASET < RSET$ [1]. This probability is calculated as a ratio of fire scenarios where the evacuees have contact with smoke, to the total number of fire scenarios. The smoke is, however, characterized by pre-defined (by users) visibility, temperature and layer height parameters. The method is quite straightforward, however, we claim that this is an only rough estimation of safety. The more detailed performance criteria can be further evaluated by the discussed in the article FED-based method.

Acknowledgements

Aamks is hosted at github at <http://github.com/aamks>. The authors would like to thank Rector of The Main School of Fire Service for his support in the grid infrastructure for software development and simulations environment. Additionally, the authors thank Mateusz Fliszkiewicz for development of GUI, Przemysław Kubica and Mariusz Pecio for support in the development of the multisimulation method.

Open Access

This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Babrauskas V, Fleming JM, Don Russell B (2010) RSET/ASET, a flawed concept for fire safety assessment. *Fire Mater* 34(7):341–355
2. Barrois T (1835) Essai sur l'application du calcul des probabilités aux assurances contre les incendies [A proposition on the application of probability theory on fire insurance]. *Mmoires de la société royale des sciences, de l'agriculture et des arts de Lille* 11:85–282
3. Van den Berg J, Lin M, Manocha D (2008) Reciprocal velocity obstacles for real-time multi-agent navigation. In: IEEE international conference on robotics and automation, 2008. ICRA 2008, IEEE, pp 1928–1935
4. Bertsekas DP, Tsitsiklis JN (2002) Introduction to probability, Athena Scientific Belmont, MA
5. Brandes U, Erlebach T (2005) Network analysis: methodological foundations, 3418Springer, Berlin
6. British Standards Institute (2004) Pd 7974-6:2004, the application of fire safety engineering principles to fire safety design of buildings
7. Bryner R, Gann N (2008) Combustion products and their effects on life safety. *Fire Protection Handbook* 1:6–11
8. Bukowski RW, Clarke F, Hall JR, Stiefel S (1990) Fire risk assessment method: description of methodology. *Cent Fire Res*
9. Chazelle B (1982) A theorem on polygon cutting with applications. In: 23rd annual symposium on foundations of computer science, 1982. SFCS'08, IEEE, pp 339–349
10. Christian RP, Casella G (2007) Monte Carlo statistical methods. Springer, New York
11. Collaborative work (2014) FRAME fire risk assessment for engineering. FRAME—validation. Tech. rep
12. Dutra TB, Marques R, Cavalcante-Neto JB, Vidal CA, Pettré J (2017) Gradient-based steering for vision-based crowd simulation algorithms. In: Computer graphics forum, vol 36(2). Wiley Online Library, pp 337–348. <https://doi.org/10.1111/cgf.13130>
13. Ericson CA et al (2015) Hazard analysis techniques for system safety. John Wiley & Sons, Hoboken
14. Fiorini P, Shiller Z (1998) Motion planning in dynamic environments using velocity obstacles. *The International Journal of Robotics Research* 17(7):760–772
15. Frantzich H, Nilsson D (2003) Utrymning genom tät rök: beteende och förflyttning. Department of Fire Safety Engineering, Lund University, Sweden
16. Fraser-Mitchell J (1994) An object-oriented simulation (crisp 11) for fire risk assessment. *Fire Saf Sci* 4:793–804
17. Fruin JJ (1971) Pedestrian planning and design. Metropolitan association of urban designers and environment planners
18. Gann RG, Averill JD, Butler K, Jones W, Mulholland G, Neviasser J, Ohlemiller T, Peacock R, Reneke P, Hall Jr J (2001) Sublethal effects of smoke on survival and health. In: Proceedings of the 2nd International Symposium on Human behaviour in fire, MIT, Cambridge, USA, pp 285–296
19. Hadjisophocleous GV, Mehaffey JR (2016) Fire scenarios. In: SFPE handbook of fire protection engineering, Fifth Edition. Springer, Berlin, pp 1262–1288
20. Haouari-Harrak S, Fromy P, Boulet P, Mehaddi R, Blanchard E (2016) Fire risk assessment with a stochastic approach—model development and application to an auditorium. In: Proceeding of 14th Interflam Conference
21. Hartzell GE, Switzer WG, Priest DN (1985) Modeling of toxicological effects of fire gases: V. Mathematical modeling of intoxication of rats by combined carbon monoxide

- and hydrogen cyanide atmospheres. *J Fire Sci.* <https://doi.org/10.1177/073490418500300504>
22. Helbing D, Farkas I, Vicsek T (2000) Simulating dynamical features of escape panic. *Nature* 407(6803):487
 23. Helbing D, Farkas IJ, Molnar P, Vicsek T (2002) Simulation of pedestrian crowds in normal and evacuation situations. *Pedestrian and evacuation dynamics* 21(2):21–58
 24. Helbing D, Isobe M, Nagatani T, Takimoto K (2003) Lattice gas simulation of experimentally studied evacuation dynamics. *Phys Rev E—Stat Phys Plasmas Fluids Relat Interdiscip Top.* <https://doi.org/10.1103/PhysRevE.67.067101>
 25. Hostikka S, Keski-Rahkonen O (2003) Probabilistic simulation of fire scenarios. *Nuclear engineering and design* 224(3):301–311
 26. Hughes R, Ondřej J, Dingliana J (2014) Holonomic collision avoidance for virtual crowds. In: *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, pp 103–111. Eurographics Association
 27. Hughes R, Ondřej J, Dingliana J (2015) Davis: density-adaptive synthetic-vision based steering for virtual crowds. In: *Proceedings of the 8th ACM SIGGRAPH conference on motion in games*, pp 79–84. ACM
 28. Hull TR, Stec AA, Paul K (2008) Hydrogen chloride in fires. *Fire Safety Science* 9:665–676
 29. Hurley MJ, Rosenbaum ER (2016) Performance-based design. In: *SFPE handbook of fire protection engineering*, Fifth Edition. Springer, Berlin, pp 1233–1261
 30. Hurley MJ, Gottuk DT, Hall Jr JR, Harada K, Kuligowski ED, Puchovsky M, Watts Jr JM, Wieczorek CJ, et al (2015) In: *SFPE handbook of fire protection engineering*. Springer, Berlin
 31. Isobe M, Helbing D, Nagatani T (2004) Experiment, theory, and simulation of the evacuation of a room without visibility. *Phys Rev E—Stat Phys Plasmas Fluids Relat Interdiscip Topics.* <https://doi.org/10.1103/PhysRevE.69.066132>
 32. Jin T (1978) Visibility through fire smoke. *J Fire Flammability* 9:135–155
 33. Jin T (2016) Visibility and human behavior in fire smoke. In: DiNenno PJ, Drysdale D, Beyler CL, Walton WD, Custer RLP, Hall Jr JR, Watts Jr JM (eds) *The SFPE handbook of fire protection engineering*, 3rd edition. Society of Fire Protection Engineers/National Fire Protection Association, Quincy, Massachusetts
 34. Jones WW, Forney GP, Peacock RD, Reneke PA (2017) A technical reference for cfast: an engineering tool for estimating fire and smoke transport. NIST TN 1431. <https://github.com/firemodels/cfast/releases>
 35. Kapadia M, Pelechano N, Allbeck J, Badler N (2015) Virtual crowds: steps toward behavioral realism. In: *Synthesis lectures on visual computing: computer graphics, animation, computational photography, and imaging* 7(4):1–270
 36. Karamouzas I, Heil P, Van Beek P, Overmars MH (2009) A predictive collision avoidance model for pedestrian simulation. In: *International workshop on motion in games*, pp 41–52. Springer, Berlin
 37. Karamouzas I, Skinner B, Guy SJ (2014) Universal power law governing pedestrian interactions. *Physical review letters* 113(23):238,701
 38. Karlsson B, Tomasson B (2005) Repeatability tests of a fire risk index method for multi-storey apartment buildings. *Fire Safety Science* 8:901–912
 39. Korhonen T (2016) *FDS + Evac—Technical reference + user guide*. VTT Technical Research Centre of Finland. Evac 2.5.2
 40. Lee WS, Grosh DL, Tillman FA, Lie CH (1985) Fault tree analysis, methods, and applications. a review. *IEEE transactions on reliability* 34(3):194–203

41. Li X, Chen T, Pan L, Shen S, Yuan H (2008) Lattice gas simulation and experiment study of evacuation dynamics. *Physica A: Statistical Mechanics and its Applications* . <https://doi.org/10.1016/j.physa.2008.05.024>
42. Metropolis N, Ulam S (1949) The Monte Carlo method. *Journal of the American statistical association* 44(247):335–341
43. Mononen M (2009) Navigation mesh generation via voxelization and watershed partitioning. AiGameDev. comSlides available online (as of 30th July 2013) at https://sites.google.com/site/recastnavigation/MikkoMononen_RecastSlides.pdf
44. Moussaïd M, Helbing D, Theraulaz G (2011) How simple rules determine pedestrian behavior and crowd disasters. *Proceedings of the National Academy of Sciences* 108(17):6884–6888
45. Owen AB (2013) Monte Carlo theory, methods and examples. <http://statweb.stanford.edu/~owen/mc/>
46. Peacock RD, Jones W, Reneke P, Forney G (2017) CFAST-consolidated model of fire growth and smoke transport (version 7) user's guide. Citeseer. <https://github.com/firemodels/cfast/releases>
47. Pelechano N, Allbeck JM, Badler NI (2007) Controlling individual agents in high-density crowd simulation. In: *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on computer animation*, pp. 99–108. Eurographics Association
48. Pettré J, Ondřej J, Olivier AH, Cretual A, Donikian S (2009) Experiment-based modeling, simulation and validation of interactions between virtual walkers. In: *Proceedings of the 2009 ACM SIGGRAPH/Eurographics symposium on computer animation*, pp 189–198. ACM
49. Predtechenskii M, Milinski AI (1978) *Planning for foot traffic flow in buildings*. Amerind Publishing Co Pvt Ltd, New Delhi. Russian edition: *Proektirovanie Zhdanii s Uchetom Organizatsii Dvizheniya Lyudskikh Potokov*, Stroiizdat Publishers, Moscow (1969)
50. Purser DA (2002) Toxicity assessment of combustion products. *SFPE handbook of fire protection engineering* 3:2–6
51. Purser DA (2016) Combustion toxicity. In: *SFPE handbook of fire protection engineering, Fifth Edition*. https://doi.org/10.1007/978-1-4939-2565-0_62
52. Rahikainen J, Keski-Rahkonen O (1998) Determination of ignition frequency of fires in different premises in finland. *Fire Engineers Journal* 58:33–37
53. Reynolds CW (1987) Flocks, herds and schools: a distributed behavioral model. In: *ACM SIGGRAPH computer graphics*, vol 21, pp 25–34. ACM
54. Reynolds CW (1999) Steering behaviors for autonomous characters. *Game developers conference 1999*:763–782
55. Ronchi E, Kuligowski ED, Reneke PA, Peacock RD, Nilsson D (2013) The process of verification and validation of building fire evacuation models. *NIST Technical Note* 1822
56. Singh S, Kapadia M, Hewlett B, Reinman G, Faloutsos P (2011) A modular framework for adaptive agent-based steering. In: *Symposium on interactive 3D graphics and games*, p 9. ACM
57. Singh S, Kapadia M, Reinman G, Faloutsos P (2011) Footstep navigation for dynamic crowds. *Computer Animation and Virtual Worlds* 22(2–3):151–158
58. Speitel LC (1995) Toxicity assessment of combustion gases and development of a survival model. Tech. rep., DTIC Document
59. International Organization for Standardization IO (2012) *Fire safety engineering Fire riskassessment Part 1: general*. ISO 16732-1:2012(E)
60. Stec AA, Hull TR (2010) *Fire toxicity*. Elsevier, Amsterdam

61. The Ministry of Business, Innovation and Employment: C/VM2 Verification Method: Framework for Fire Safety Design (2013). New Zealand
62. Tillander K (2004) Utilisation of statistics to assess fire risks in buildings. VTT Technical Research Centre of Finland
63. Ulam S, von Neumann J, Richtmyer R (1947) Statistical methods in neutron diffusion. Tech. Rep. LAMS-551, Los Alamos National Laboratory. <http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-00551-MS>
64. van den Berg J, Guy SJ, Lin M, Manocha D (2011) Reciprocal n-body collision avoidance. In: Pradalier C, Siegwart R, Hirzinger G (eds) Robotics Research. Springer Tracts in Advanced Robotics, vol 70. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-19457-3_1
65. Wade C, Baker G, Frank K, Harrison R, Spearpoint M (2016) B-RISK 2016 user guide and technical manual. Study Report SR364. Tech. rep
66. Weiguo S, Libi F (2015) Verification and validation methods. In: Cuesta A, Abreu O, Alvear D (eds) Evacuation modeling trends, pp 81–102. Springer, Berlin
67. Wolinski D, Lin MC, Pettré J (2016) Warpdriver: context-aware probabilistic motion prediction for crowd simulation. ACM Transactions on Graphics (TOG) 35(6):164
68. Wu C, Zarrinmehr S, Asl MR, Clayton MJ (2015) Facilitating fire and smoke simulation using building information modeling. In: International conference on computer-aided architectural design futures, pp 366–382. Springer
69. Xing L, Amari SV (2008) Fault tree analysis. In: Misra KB (ed) Handbook of performability engineering, pp 595–620. Springer, London. <https://doi.org/10.1007/978-1-84800-131-2>
70. Zhang J, Klingsch W, Schadschneider A, Seyfried A (2012) Ordering in bidirectional pedestrian flows and its influence on the fundamental diagram. Journal of Statistical Mechanics: Theory and Experiment 2012(02):P02,002