Appl Intell
DOI 10.1007/s10489-014-0538-9

Escape planning in realistic fire scenarios with Ant Colony Optimisation

Morten Goodwin · Ole-Christoffer Granmo · Jaziar Radianti

© Springer Science+Business Media New York 2014

Abstract An emergency requiring evacuation is a chaotic event, filled with uncertainties both for the people affected and rescuers. The evacuees are often left to themselves for navigation to the escape area. The chaotic situation increases when predefined escape routes are blocked by a hazard, and there is a need to re-think which escape route is safest. This paper addresses automatically finding the safest escape routes in emergency situations in large buildings or ships with imperfect knowledge of the hazards. The proposed solution, based on Ant Colony Optimisation, suggests a near optimal escape plan for every affected person — considering dynamic spread of fires, movability impairments caused by the hazards and faulty unreliable data. Special focus in this paper is on empirical tests for the proposed algorithms. This paper brings together the Ant Colony approach with a realistic fire dynamics simulator, and shows that the proposed solution is not only able to outperform comparable alternatives in static and dynamic environments, but also in environments with realistic spreading of fire and smoke causing fatalities. The aim of the solutions is usage by both individuals, such as from a personal smartphone of one of the evacuees, or for emergency personnel trying to assist large groups from remote locations.

Keywords Escape planning · Optimisation · Ant colony optimisation · Swarm intelligence

1 Introduction

Evacuation planning is challenging due to the typically chaotic and un-organised situation occurring in a crisis situation. Unfortunately, decision makers often have an incomplete picture of hazards and potential escape routes. The situation is further complicated by the fact that people affected are often left alone without any contact with rescue personnel, or let alone a reliable overview of the ongoing crisis situation. In addition, the chaotic and dynamic nature of crisis situations quickly changes which path is the best escape route as hazards, such as fires, rapidly develop.

There is no doubt that decision making in crisis situations needs to be timely to minimize the consequences of hazards at hand. However, it is often difficult for the people affected to determine what are the best decisions in an evacuation situation. In fact, in most situations the evacuees are not aware of which path to follow for an escape. This is because evacuees either received insufficient information from the rescuers or are unfamiliar with the layout of the affected compound and locations of the hazards. Similarly, the emergency personnel often lack an overview of where people are located and which rooms are affected by hazards. This makes evacuation planning particularly difficult [1, 2].

This paper is part of a larger project[1] working on using smartphone technologies in emergency situations. In this

Preliminary versions of some of the results of this paper appeared in the Proceedings for 26th International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2013.

M. Goodwin (✉) · O.-C. Granmo · J. Radianti
Faculty of Engineering and Sciences, Department of ICT, Univesity of Agder, Grimstad, Norway
e-mail: morten.goodwin@uia.no

O.-C. Granmo
e-mail: ole.granmo@uia.no

J. Radianti
e-mail: jaziar.radianti@uia.no

[1] SmartRescue: http://ciem.uia.no/project/smartrescue

project, the smartphones are used to communicate to and from people affected by crisis situations. This enables rescue personnel to get an overview of the people escaping such as information on how they move. Further, the rescuers can communication how to best escape to the evacuees. In addition, the sensor information available in smartphones will communicate presence of hazards using camera, GPS, gyroscope, etc. This provides a threat map which is support both the rescue personnel and as a method to automatically determine the best escape routes [3].

The project has three main steps outlined in the sections below:

1. Collect information.
2. Calculate escape plan.
3. Communicate the plan to the people affected.

1.1 Collect information

Initially in a crisis situation, it is essential to get an overview of the people affected and the hazards present.

The aim is that prior to a crisis situation, for example when people embark on a cruise ship, they will have the option to download a mobile application for their phone. This application will in a crisis situation utilize the available sensors and communicate it to a central location. This way, the system, including emergency personnel, will be aware of life saving information such as locations of people, whether people are moving, the brightness in each room (indicating hazards such as fire or smoke). The application aims at automatic activation rather than being actively started by evacuees. This allows the application to automatically start when hazards are detected or communicated from a central location without any activation by the people affected.

1.2 Determine plan

Calculating the escape plan is the main contributions of this paper. Mathematical models have shown to be valuable for escape planning in large complex building with many people [4–8], but the approaches in the literature are mainly assuming a static representation of hazards. In contrast, our system will, based on available information, calculate the best escape plan and guide each affected person away from any potential hazard, as well as distributing the people to the proper escape areas even when the hazards change.

1.3 Communicate plan

When an adaptive plan is available, it should be communicated to the affected people, which can be done in two main ways. The primary method aimed for in the SmartRescue project is for emergency personnel to actively communicate the plan to the evacuees through any available means, such as loud speakers and communication directly to each affected person via the smartphone applications. Failing this, the smartphone applications should automatically present the plan using simple visual and verbal steps such as “turn around”, “go left” and safely guide people to an escape area.

It is also worth noting that the collection, determination and communication is decentralized without a need for a central network such as WiFi or GPS — which may be unavailable in emergency situations. The phones will communicate through so-called ad-hoc networks, which enables communication directly between phones without any central network available [9]. Further, decisions can be made when only part of the network is available such as when two groups are physically separated on board a ship without any common network connection between them.

Unfortunately, even in situations where an optimal escape plan exists and every person affected is aware of the plan, the human mind works so that not everyone will follow the plan [10]. Most significantly, in a crisis situation, factors such as panic spread, people pushing, jamming up and overlooking alternative exits prevent a crowd from following an optimal plan [11, 12]. Therefore, it is important that information about both hazards and people are continuously updated to always provide the best overview for rescue personnel and an update plan.

1.4 Outline

The rest of the paper is organized as following. Section 2 defines the problem to be solved as an optimisation problem with hazard functions, including functions based on realistic spread of fire and smoke. Section 3 continues with introduction of the ant colony optimisation with focus on finding safe escape routes. The approach and empirical results from five distinct environments are presented in Section 4 Lastly, Section 5 concludes this paper and maps out further work.

2 Problem formulation

Escape planning from a complex building or a large ship can be regarded as a combinatorial optimisation problem. In line with common practice [6], we treat the building layout as a bidirectional planar graph $G(V, E)$. Each possible location i is connected with a vertex $v_i \in V$, and each potential flow from vertex v_i to v_j is represented by an edge $e_{i,j} \in E$.

In addition we define a function $h(v_i, t)$ representing the hazard for v_i at time t, so that the function $h(v_i, t)$ returns probability values representing the likelihood of hazards.

The escape area is a vertex $v_e \in V$ (sink), and the people are located in any vertex $v_s \in V$ (any vertex is a source).

Further, all routes from v_s to v_e in the search space are defined as **S**.

The aim of the application is to find a route $s* \in \mathbf{S}$ so that $f(s^*, t) \leq f(s, t) \forall_s \in \mathbf{S}$, where $f(s, t) = 1 - \Pi_{v_i \in s}(1 - h(v_i, t))$. I.e. minimizing the probability that a person encounters a hazard in at least one of the vertexes in the chosen route at time t.

2.1 Hazards

The hazard functions are based upon known observations from indications and estimations. If a hazard $h(v_i, t) = 1$, it means that there is a known hazard and vertex v_i is unsafe at time t, and all evacuees should be routed away from the corresponding room. Similarly, $h(v_i, t) = 0$ means that v_i is a known safe vertex. All other hazards are estimated based on known observations.

This paper presents three distinct variants of the hazard function. For clarity, these are noted as $h^1(v_i, t)$, $h^2(v_i, t)$ and $h^3(v_i, t)$. The algorithms do not have knowledge of how the functions behave, let alone which variant of the hazard is used on the model. This is because hazard propagations are complicated and require significant calculation to be correct. Hence, all algorithms assume an unknown stochastic function, $h(v_i, t)$, which returns hazard probabilities. In other words, from the view of the algorithm all functions are treated equally and only noted as $h(v_i, t)$.[2]

In its simplest form, $h^1(v_i, t)$ is a function defined as a uniform random function yielding a probability in [0, 1], which means that t is ignored. Section 4.1 presents results from this scenario.

The initial function is extended in two ways. Firstly, $h^1(v_i, t)$ is extended to $h^2(v_i, t)$ by making it time dependant. This means that the function is initially based on random uniform variables. Consequently, it updates according to the following rule with a global n representing time shifts:

$$\begin{aligned} h^2(v_i, t) &= \quad 1 - h^2(v_i, t-1) \forall_{v_i} \in V \, if \, t \bmod n = 0 \\ h^2(v_i, t) &= \qquad h^2(v_i, t-1) \forall_{v_i} \in V \; otherwise \end{aligned} \tag{1}$$

Thus, at a given time n the environment updates the function in line with (1) and makes all hazards exactly opposite. This produces a particularly difficult situation since an (near) optimal route changes to as far away from optimal as possible, and algorithms which have learnt an optimal route will need to completely relearn its learnt behaviour. Results from this hazard function are available in Section 4.2.

Secondly, the function $h^3(v_i, t)$ is defined to represent realistic probabilities of fatal hazards based on exposure levels of thermal radiation[3] and temperature emitted by the fire[13–15]. This extension relies upon timely measurements (or estimations) of temperature and radiation levels in emergency situations. Such measurements can be collected from fixed or mobile sensors [3].

Unfortunately, the literature only provides information on maximum possible exposure time for fixed temperatures and radiation level, which does not fit with the probabilistic $h^3(v_i, t)$ function. This is solved by converting the exposure time of temperature and radiation to uniformly distributed probabilities and combined as one overall hazard function.

It is worth noting that this is a simplification of reality. People's abilities to withstand hazards depend on many factors including age, health, gender and time spent in the hazard. Further, people have thresholds meaning that staying in a hazardous room a short time may not be fatal, but staying there longer is likely to cause fatalities. The functions in this paper are no way meant to be an overall representation of survivability. The functions are rather meant to show the probability of detecting fatal hazards which are likely to cause death or incapcitations. Whether or not people will actually die from the fatal hazards are out of scope of this paper.

The concrete mapping is done by creating two functions, $c(v_i, t)$, the probability of an evacuee having fatal impact due to high temperatures in room v_i at time t and $r(v_i, t)$, the probability of an evacuee having a fatal impact because of high radiation levels in room v_i at time t. The $c(v_i, t)$ and $r(v_i, t)$ functions are formally defined as following:

The probability of a fatal impact due to high temperatures in room v_i is based on the levels in [14], defines as:

$$c(v_i, t) = \begin{cases} \dfrac{1}{300\ s} & \textit{if for } v_i \, 140^\circ C > \textit{temperature at time } t \\ \dfrac{1}{3600\ s} & \textit{if for } v_i \, 140^\circ C \geq \textit{temperature} \geq 80^\circ C \textit{at time } t \\ 0 & \textit{otherwise} \end{cases} \tag{2}$$

which in layman's terms state that the probability of an evacuee encountering a fatal hazard in a room with temperature between 80 and 140 °C is for every second stayed there $\frac{1}{3600}$. If a room has temperatures above 140°C, the probability of a fatal hazard is $\frac{1}{300}$ for every second stayed there.

[2]This is not to be confused with that hazard propagations are ignored. In fact, paper shows that even without the complicated hazard propagations, the learning algorithms learn and predict the hazards.

[3]Note that radiation in this paper refers to thermal radiation emitted by fire, and should not be confused with any other form of radiation such as nuclear radiation.

Similarly, the probability a fatal impact because of of radiation in room v_i is defined as:

$$r(v_i,t)=\begin{cases}\frac{1}{90\ s}\ if\ for\ v_i 6.0kW/m^2 > radiation\ at\ time\ t\\ \frac{1}{180\ s}\ if\ for\ v_i 6.0kW/m^2 \geq radiation \geq 4.0kW/m^2 at\ time\ t\\ \frac{1}{600\ s}\ if\ for\ v_i 4.0kW/m^2 \geq radiation \geq 1.6kW/m^2 at\ time\ t\\ 0\ \ otherwise\end{cases} \tag{3}$$

which in layman's terms mean that with radiation exposure between 1.6 and 4.0 kW/m^2 evacuees encounter a fatal hazard with a probability $\frac{1}{600}$ per second, between 4.0 and 6.0 kW/m^2 they encounter a fatal hazard with probability $\frac{1}{180}$ and exposures above 6.0 kW/m^2 have a probability of impact of $\frac{1}{90}$.

The functions are visually shown in Fig. 1.

Finally, the hazard function, $h^3(v_i, t)$ is simply the probability of either $c(v_i, t)$ or $r(v_i, t)$ occurring:

$$h^3(v_i, t) = r(v_i, t) + c(v_i, t) - r(v_i, t) * c(v_i, t) \tag{4}$$

In other words, the hazard for a room i is defined as the probability of a person either encountering fatal heat radiation or temperature.

Sections 4.3, 4.4 and 4.5 presents results from this hazard function.

2.2 Smoke

In fire situations, smoke causes significant challenges for the people involved both due to direct fatalities, e.g. carbon monoxide poisoning of the people affected, and because soot from the smoke reduces the distance of evacuees' vision. In fact, it is so that for a successful evacuation, there should be a minimum of 3 meters of vision in primary compartments and 10 meters in predefined escape routes [13]. Hence, when exposed to a fire, people are able to exit their rooms even with visions down towards 3 meters, but require 10 meters of vision to follow an escape plan in corridors leading directly to an escape area.

To compensate for this in our model, we define a time dependant function $m(v_i, t)$ which indicates whether or not it is possible for the evacuees to move due to smoke obscuration:

$$m(v_i,t)=\begin{cases} true if\ corridor\ v_i leads\ directly\ to\ an\ escape\\ \quad area\ and\ vision\ \geq 10m\ at\ time\ t\\ true if\ primary\ compartment\ v_i has\ vision\ \geq 3m\\ \quad at\ time\ t\\ false otherwise\end{cases} \tag{5}$$

This can be read as people get "stuck" in rooms where the vision is so low that it impairs their sight and movability.

3 Ant Colony Optimisation (ACO)

Problem solving approaches inspired by nature and animals, so called swarm intelligence, have received a lot of attention due to their simplicity and adaptability. Ant Colony Optimisation (ACO) is one of the most popular swarm intelligence algorithms due to its general purpose optimisation technique. ACO consists of artificial ants operating in a constructed graph. The ants release pheromones in favorable paths which subsequent ant members follow. This

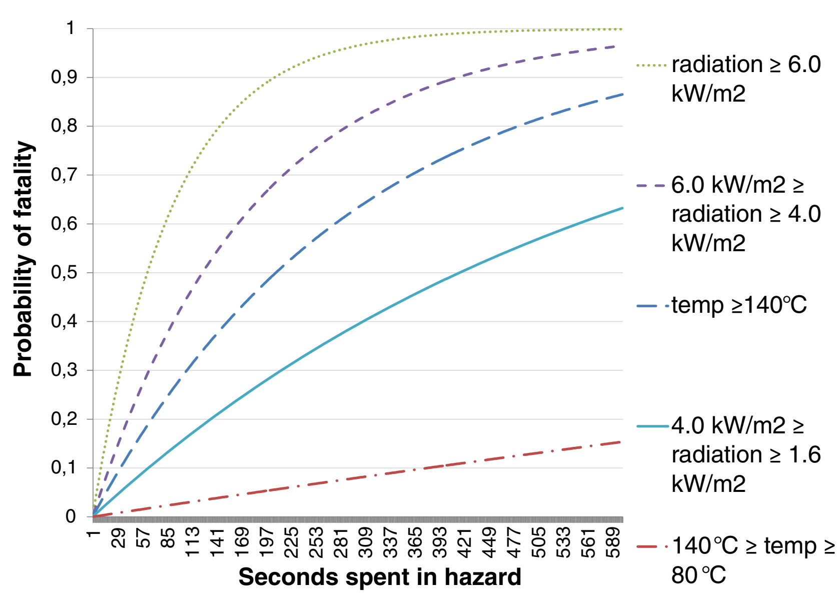

Fig. 1 Temperature and thermal radiation levels as input to $c(v_i, t)$ and $r(v_i, t)$

way, the colony of ants will walk more towards favorable paths and in consequence iteratively build the most favorable solution [16].

ACO was first used to find shortest path from a source to a sink in a bidirectional graph. It has later increased in popularity due to its low complexity and its ability to work in dynamic environments. The flexibility of ACO is apparent as it has successfully been applied in a wide variety of problems such as finding solutions for NP hard problems [17], rule based classification [18], and is shown to be particularly useful for routing in real time industrial and telecommunication applications [19].

Finding the shortest path in a graph $G(V, E)$ using ACO in its simplest form works as follows. Artificial ants move from vertex to vertex. When an ant finds a route s from the source v_s to the sink v_e, the ant releases pheromones $\tau_{i,j}$ corresponding all edges $e_{i,j} \in s$. The pheromones for all ants m is defined as

$$\tau_{i,j} \leftarrow (1-p)\tau_{i,j} + \sum_{k=1}^{m} \Delta\tau_{i,j}^{k} \tag{6}$$

The function is for ant k defined as

$$\Delta\tau_{i,j}^{k} = \begin{cases} Q/|s| if\, e_{i,j} \in s \\ \quad 0 otherwise \end{cases} \tag{7}$$

where Q is a constant.

The aim of each ant is to walk from v_s to v_e forming the path s. This is achieved by the following rule. When ant k is in vertex i it chooses to go to vertex j with the probability $p_{i,j}^{k}$ defined as

$$p_{i,j}^{k} = \begin{cases} \dfrac{\tau_{i,j}^{\alpha}\eta_{i,j}^{\beta}}{\sum_{e_{i,j}} \tau_{i,k}^{\alpha}\eta_{i,k}^{\beta}} & if\, e_{i,j} \in N(s^p) \\ \quad 0 & otherwise \end{cases} \tag{8}$$

where s^p is the partial solution of s, and $N(s^p)$ are the possible vertexes to visit given s^p. $\eta_{i,j}$ is the inverse heuristic estimate of the distance between node i and j, and α and β are numbers between 0 and 1 to give the relevant importance between pheromones and the heuristics function.

In its simplest form, the $\beta = 1$ and $\alpha = 1$ so that the ants only consider the pheromones — and the heuristic function is ignored, giving:

$$p_{i,j}^{k} = \begin{cases} \dfrac{\tau_{i,j}}{\sum_{e_{i,j}} \tau_{i,k}} & if\, e_{i,j} \in N(s^p) \\ \quad 0 & otherwise \end{cases} \tag{9}$$

In layman terms, the amount of pheromone released represent quality of the solution. This is achieved by each ant releasing a constant amount of pheromones. Consequently, the shorter the path found, the more pheromone per edge is released. Further, each ant is guided by a stochastic mechanism biased by the released pheromones. Thus, the ants walk randomly with a preference towards pheromones. In this way, the ants incrementally build up promising search space with means that a route s converges towards the shortest route from v_s to v_e, $s*$.

ACO has also successfully been applied for many network applications [19–22]. It has been empirically shown to have favourable results compared to other routing protocols with respect to short path routing and reduced load balancing. Therefore it seems particularly promising for finding the escape routes.

4 Solution

This section presents the ACO algorithms for finding escape routes in four distinct realistic environments.

First, the algorithms interact with a **static environment** where the hazard functions remain unchanged, yet unknown. This resembles classical optimisation problems where the aim is to find a route s^* so that $f(s^*, t) \leq f(s, t) \forall_s \in \mathbf{S}$. Hence, the environment is populated with the $h^1(v_i, t)$ function. (See Section 4.1 for results.) Subsequently, the problem is extended to interact with **dynamic environments** so that the probability of a hazard in v_i, $h^2(v_i, t)$, is no longer fixed but changes regularly according to some unknown stochastic functions. This shows how well ACO works when environments change, such as fire spreading. (See Section 4.2 for results.)

Further, ACO handles **realistically simulated environments** where the hazard function, $h^3(v_i, t)$ is based on measurements of thermal radiation and air temperature. The simulation is carried out using a Fire Dynamic Simulation (see Section 4.3 for results).

Subsequently, the environment is extended with **smoke**, represented with the function $m(v_i, t)$ so that the vision range and in turn movability evacuees is significantly reduced when smoke is present (See Section 4.4 for results).

Lastly, the environment is updated with imperfect knowledge so that only hazard functions close to evacuees are available. Hence, evacuees will not have any knowledge of fires far away even if it may obstruct evacuation routes. This simulates an **ad-hoc network** setup with only part of the environment's sensors are available to evacuees (see Section 4.5 for results).

This section presents empirical evidence for ACO working in all the above mentioned environments. In these experiments, all graphs are bidirectional, planar and connected — in line with common practice [6]. This is done in two ways: Without loss of generality, the first two experiments (Sections 4.1 and 4.2) are carried out on randomly generated graphs of 1,000 vertexes, and 5,000 randomly distributed edges, of which 1,000 edges are used to make sure the graph is connected. The latter experiments (Sections 4.3–4.5) are carried out on a graph modelled after an actual ship, and hazards are updated based on realistic simulation results.

All experiments are an average of 1,000 runs.[4]

4.1 Static environments

ACO has been used for static routing in many situations before. In these experiments ACO is used in its simplest form, as described in Section 3, with a slight adjustment.[5]

The constant Q is replaced with a function of s and t:

$$Q(s,t) = \Pi_{v_i \in s}(1 - h(v_i, t)) \tag{10}$$

I.e. $Q(s,t)$ represent the inverse hazard probability at time t. The consequence of this is that safe paths are given large amounts of pheromones, and unsafe paths are given low amount. The pheromone updates are therefore as

$$\Delta\tau_{i,j}^{k} = \begin{cases} Q(s,t)/|s| & if\, e_{i,j} \in s \\ 0 & otherwise \end{cases} \tag{11}$$

Figures 2 and 3 show the behaviour of the ACO the static environment. Figure 3 shows the behaviour were there in addition to the normal setup the graph is manipulated so that there exists an s so that $f(s,t) = 0$ — meaning that there always exists a safe path. The optimal solution is calculated using Djikstra's algorithm [24] by considering $h(v_i, t)$ as basis the cost function for edged $e_{*,i}$.

Figure 3 shows the same experiment but with adjustments of hazard probabilities so that there is an s so that $f(s,t) = 0$ — meaning that there always exists a safe path.

Both experiments show that ACO is able to find the near optimal solution with very few iterations.

4.2 Dynamic environments

ACO has been used for dynamic environments in many situations [25–28]. This is achieved by letting, for each time step, the pheromones evaporate with a defined probability, typically between 0.01 and 0.20 [16]. The evaporation probability is a balance between convergence accuracy and adaptability. I.e. you choose to what extent the ants should work towards a more optimal solution or should be able to adapt to other potential solutions. Dressler et al. [27] showed that ACO based routing works well in situations with significant dynamics and continuously broken and newly established connections, which resembles finding an escape route when hazards change.

[4]Many additional experiments have been carried out. For reasons of brevity, only those more relevant are presented in this paper. Some of the additional experiment results are available in [23].

[5]Note that for the static environment the variants of $h(v_i, t)$ used is $h^1(v_i, t)$.

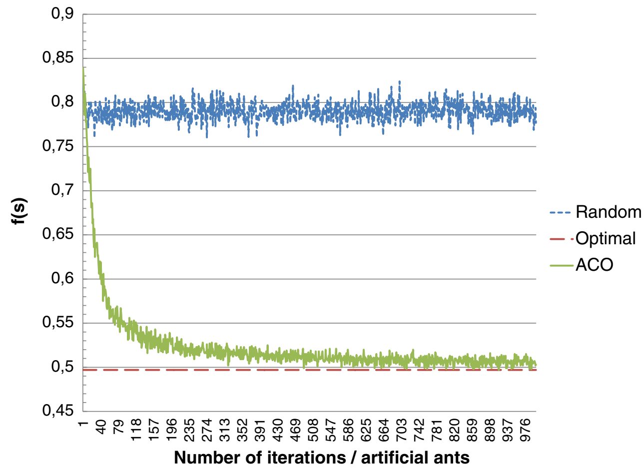

Fig. 2 Experiment results in static environments of randomly generated large graph comparing ACO to random and optimal. $h(v_i, t)$ random to 1 or 0

Figures 4 and 5 show ACO in dynamic environments where the variant of $h(v_i, t)$ used is $h^2(v_i, t)$ and the update function (see (1)) is called every 200th.

Thus, for every 200th iteration the environment is exactly opposite which in turn means that, if an algorithm has learnt an optimal route, the route changes to as far away from optimal as possible.

The results show in Fig. 4 that when the evaporation rate is set to 0, the ACO learns a near optimal solution which becomes outdated when $h(v_i, t)$ changes, and it is not able to adapt to the new optimal solution. Further, every time the hazard probabilities change the algorithm is further away from the solution. On the other hand, Fig. 5 shows that when the evaporation rate is set to 0.2, the algorithm is able to quickly adapt to new environments — and is thus able to interact well with dynamic environments.

4.3 Realistic environments

This section presents empirical results in a realistic environment. This is achieved through a sequence of steps. First we model a graph which the algorithms interact with based on an actual ship. Subsequently, the variant of the hazard function $h(v_i, t)$ is $h^3(v_i, t)$ (see Section 2.1) which is updated based on known fatalities. Finally and most importantly, this environment uses a well established fire simulator, a simulation for populations of the hazards [29]. I.e. the functions $r(v_i, t)$ and $c(v_i, t)$ are populated with results from the fire simulator. This yields realistic scenarios, including a realistic hazard function $h^3(v_i, t)$, as a step towards relevant empirical evidence of the usefulness of the

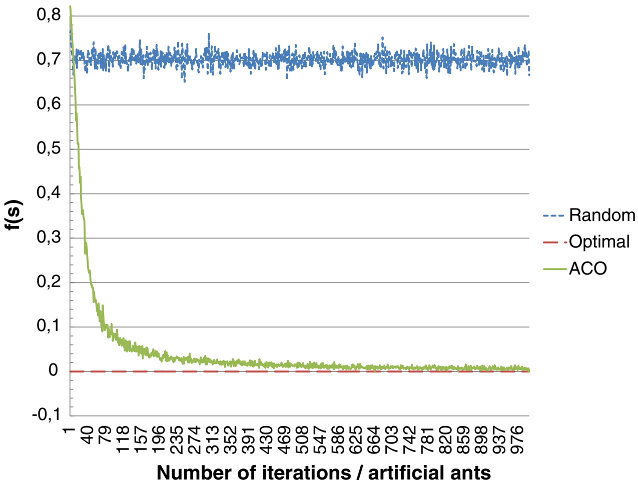

Fig. 3 Experiment results in static environments of randomly generated large graph comparing ACO to random and optimal. $h(v_i, t)$ random to 1 or 0 forcing that there is an s so that $f(s, t) = 0$

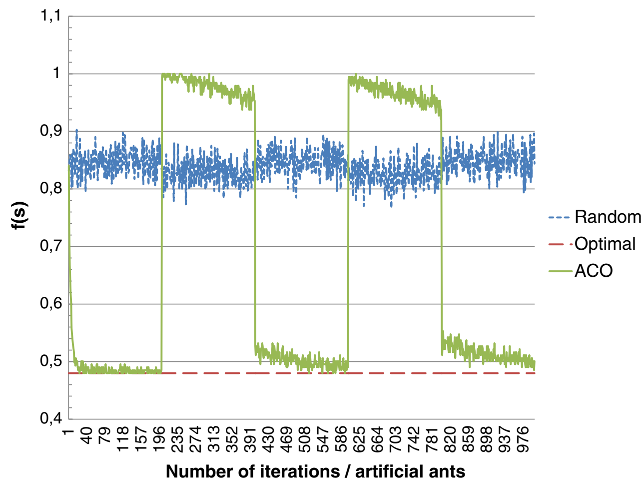

Fig. 4 Experiment results in dynamic environments. Hazard probabilities, $h(v_i, t)$, updates when $t \mod 200 = 0$ — every 200th iteration. Evaporation rate set to 0, ACO cannot adapt

algorithms. The details of each of the steps are described in the following sub sections.

4.3.1 Setup

Ship model The ship model is based on deck sections of the Thompson Spirit[6] with some minor adjustments.

The ship consists of 290 rooms and 4 corridors spread over two floors, including four exists to escape areas. The rooms vary slightly in size but are in average 15.84 m^2.

In line with the actual ship, the model of the ship uses wooden interior and "British style" carpets on all floor surfaces. Further, the floors and walls are modelled with combustion properties to enable a realistic fire spread of the material. These properties include emissivity, heat, conductivity in line with properties of the actual material used. Further emphasis is put on thickness and density of the interior to yield realistic fire spread.

The authors are aware that these parameters vary depending on the spread of the fire, open and closed areas, efforts on putting the fire out etc. However, our aim is to provide a realistic model in order to empirically show that the proposed solution is able to cope in such situations. Whenever there is room for interpretation, the model is always implemented with the worst case scenario in mind. In practice, this means that all rooms have carpets, all doors are left open and no efforts on extinguishing the fire is carried out during the simulation. If the proposed solution is able to work well in this worst case scenario, it is expected that it will also yield good results in better situations.

[6]An outline of the Thompson Spirit is available at http://www.iglucruise.com/thomson-spirit/deck-plans?deck=83.

Note that the outdoor and recreational areas, such as swimming pools and bars, are deliberately left out of the model. These are areas that have little influence on an evacuation.

The model is placed with 3 measurement devices in each room: temperature, smoke and heat, which is used for population of the hazard function $h^3(v_i, t)$. The heat is measured from the room center, radiation at ground level and smoke in the average human sight at 1.6 m. This means that the simulation yields one measurement for temperature, smoke and heat per room per second.

Fire To generate and simulate the actual fires, the well established tool Fire Dynamics Simulator (FDS) developed by the National Institute for Standards and Technology is

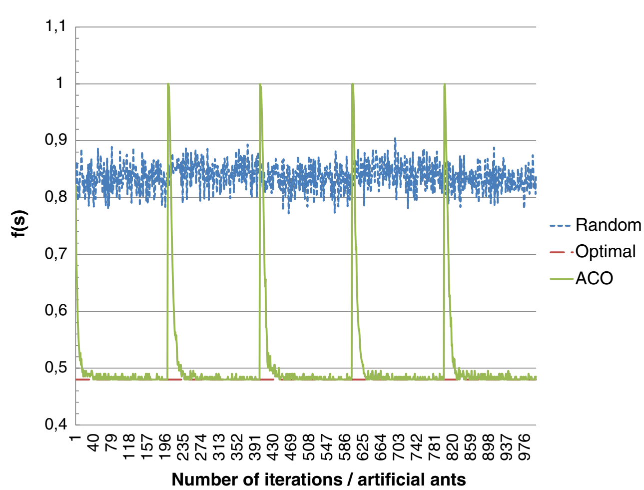

Fig. 5 Experiment results in dynamic environments. Hazard probabilities, $h(v_i, t)$, updates when $t \mod 200 = 0$ — every 200th iteration. Evaporation rate set to 0.2

used [29]. FDS used computational fluid dynamics of a fire-driven fluid flow to highlight the spread of heat and smoke from fires. This is done by a set of partial differential equations that shed light on the conservation, mass, momentum and energy within the fire and the surrounding space [30]. The tool has been used to realistically model various fire dynamics phenomena such as transport of heat and combustion products from fire, capture the heat transfer, flame spread and fire, and for designing smoke systems, sprinkler activation or fire reconstructions.

The fire was modelled so that it spreads from a 1 m^2 fire burner, and the fire was modelled to spread fast with growth coefficient was defined as 0.1876 kW/s, reaching a total of 67,536 kW during time 600 s.

This setup enables assessing effects of fire hazards both numerically and visually, which in turn enables easy integration with the proposed solution and hazard functions.

Results The simulations were carried out with fire starting in the middle of the first floor, and kept for 600 simulated seconds.[7] A visual representation of the temperature is shown in Fig. 6. Note that the fire reaches one corridor at ∼150 s, and both corridors at close to ∼500 s.

The setup is slightly different from previous experiments. In this case we have a fire that spreads out over a time frame of 600 s. These seconds are simulated, and for each second a given number of consecutive ants are released. This means that every second is in itself a distinct scenario similar to the complete experiments in Figs. 2 and 3. This facilitates convergence of the algorithm so that the route $s*$ represents the "best" possible result.[8] Further, in the follow empirical results three variants of ACO are used: with 10, 100 and 1,000 ants. This is done to show the convergence of the algorithms — how many ants are needed to find near optimal solutions. Thus, in contrast to Figs. 2 and 3 the following figures show snapshots of the algorithm after 10,100 and 1,000 ants for 600 different scenarios.

Keep in mind that all simulations run 1,000 times, and the objective is still the same, finding a route s to minimize $f(s)$. I.e. find the safest possible escape route.

Figure 7 shows the behaviour where the evacuees starting position is manipulated so that they all start from a fixed location on board the ship, the room with id 229. The room chosen to be particularly difficult to escape from is on the opposite side of the ship's escape area on the second floor — so that the fire always occurs between the evacuees and the escape area. Hence, the first room in the route s, the source v_s, is always "room 229". Note that there still exists an escape route, it is just more difficult to find.

Conversely, Fig. 8 shows the behaviour where the evacuees start from random locations on board the ship. Hence, the first room in the route s is uniformly randomly selected among all $v_i \in V$. It is noteworthy the difficult situation in Fig. 7 is remarkably similar to the over all situation in Fig. 8, only with an over all higher value for $f(s)$ for all algorithms. A conclusion to be drawn from this is that even in the difficult situations ACO is able to find safe escape routes, but requires more ants to reach convergence.

4.4 Realistic environment with smoke

This section takes into account that smoke significantly reduces vision and movability of people affected by fire hazards. In short, the model is updated with the function $m(v_i, t)$ (see (5)) which represents to what extent people are able to move considering the heavy smoke available. The $m(v_i, t)$ is populated by the FDS tool [29] presented in Section 4.3.

Consequently, the function $f(s, t)$ is adjusted with the above consideration to $f'(s, t)$:

$$f'(s,t) = \begin{cases} 1 - \Pi_{v_i \in s}(1 - h(v_i, t)) & \textit{if } m(v_i, t) = \textit{true} \ \forall v_i \in s \textit{ holds} \\ 1 & \textit{otherwise} \end{cases} \tag{12}$$

Hence, people get "stuck" in rooms with heavy smoke, and any rooms in the escape route with people "stuck" yields fatality for the evacuees.

Figure 9 shows the empirical results from this situation. This, execpt for considering smoke, is the exact same experiment as shown in Fig. 8, but provides notably different results. Hence, $f(s, t)$ and $f'(s, t)$ are very different and the results show that the fatalities are much higher when smoke is considered. This indicates that smoke has a much greater impact on evacuations than fire. This is in line with the literature which clearly states that smoke is real killer, not heat or temperature [13]. Thus, a hypothetical fire without smoke would be relatively easy to escape from compared to an actual fire the produces smoke.

The figure also shows that smoke significantly affects the performance of the ACO algorithm, but can easily be overcome by running enough ants so that it converges. Most significantly, ACO with 1,000 ants still provide close to optimal results.

[7] Two other simulations were carried out with similar setup but with fires starting in the corner of the first floor and the middle of the second floor. The results were similar to those shown in this paper — and are therefore left out.

[8] The ants do not have any knowledge of the spread of the fire other than what they observe through the function $h(v_i, t)$, so "best" means most viable seen from the small windows available from the ants. In practice, data from fixed and mobile fire sensors will populate $h(v_i, t)$.

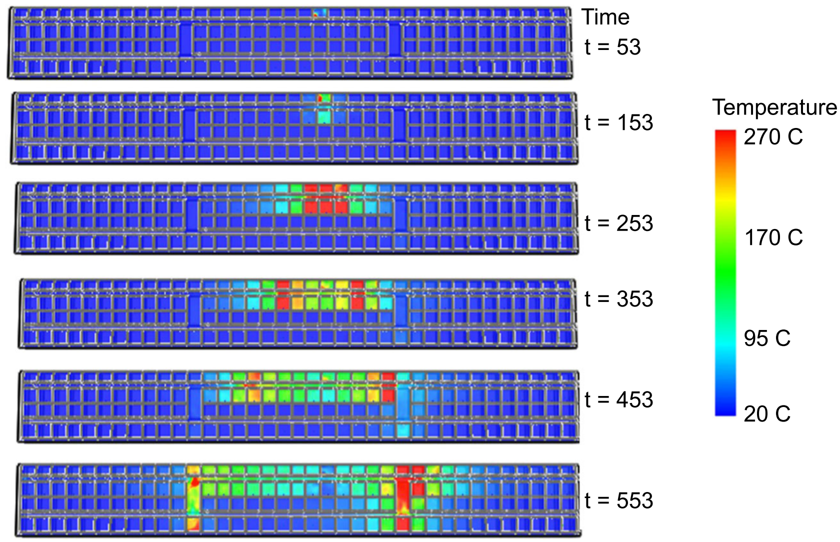

Fig. 6 Heat map visualization illustrating the room temperature as the fire spreads over time. The color-code scale on the right side shows the temperature

4.5 Realistic environment with ad-hoc networks

In emergency situations, information from smoke and fire sensors, both stationary and from smartphones, is seldom available to all. This is because networks, such as WiFi and GPS, become unavailable in fire situations. As described in Section 1.2 a mitigation for this is introduction of so called ad-hoc networks, which means that temporary networks appear when people are within close range of each other. The impact is that smartphones can only communicate with other smartphones in close range, which in turn means that only sensor data close by is available.

This simulation is done so that any function related to v_i is only available if it is n rooms away from the starting position of the evacuees. Consequently, pheromones can only update n rooms out. Say, if n is set to 5, it is a simulation of an ad-hoc network with a range of 5 rooms in each direction, and the only information available is within these 5 rooms. In essence, only the $h(v_i, t)$ functions and amount of pheromones from these 5 rooms can be polled. All other rooms are assumed safe even though they have fatal fire. Note that as pheromones in a room v_i aim at foretelling the safety of both room v_i and following rooms in any s containing v_i.

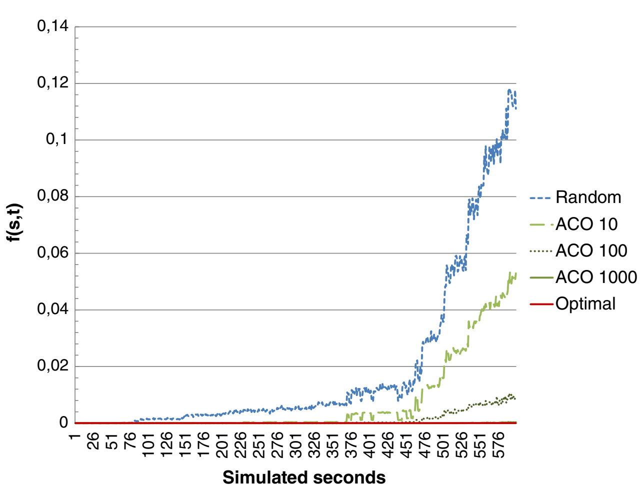

Fig. 7 Experiment results in realistic environments from a fire starting in the first floor. Evacuees starting from a fixed position in "room 229"

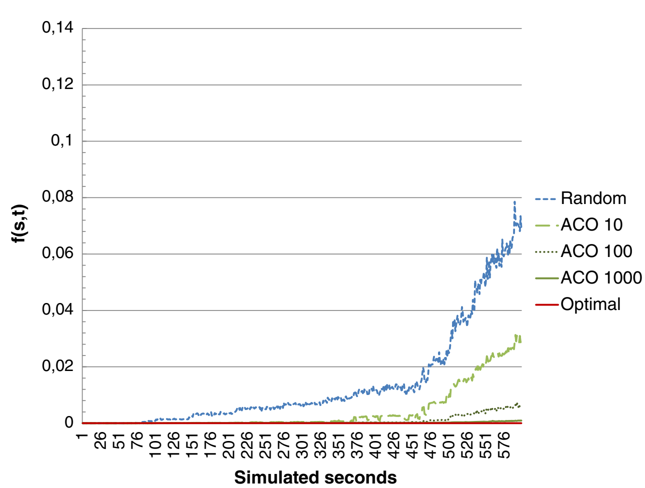

Fig. 8 Experiment results in realistic environments from a fire starting in the first floor. Evacuees starting from random positions

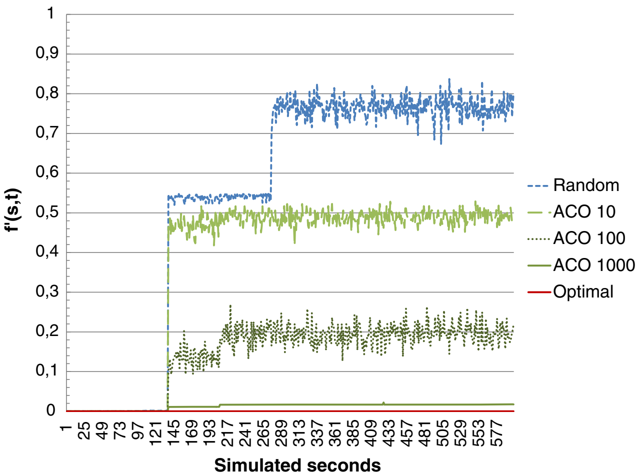

Fig. 9 Experiment results in realistic environments starting from random locations with smoke affecting the movability of the evacuees

Figure 10 shows the results in simulated ad-hoc networks. Note that this is the same experiment as presented in Fig. 8 where the complete network is available. For reasons of clarity, Fig. 10 only shows from 350 to 600 s, and include propagation setups 5 to 50. For comparative purposes, the results from Fig. 8 are included where there were not restrictions on propagations. All results are for ACO with 100 ants.

The figure clearly shows, in line with expectations, when the number of rooms the algorithm can read sensor data from increases, ACO is able to find a smaller value of $f(s,t)$. However, the difference is remarkably small. Even with a network of only 5 rooms, it is able to find a very good solution. Further, networks with 50 rooms yield and $f(s,t)$ very close to situations with sensor data from all rooms available.

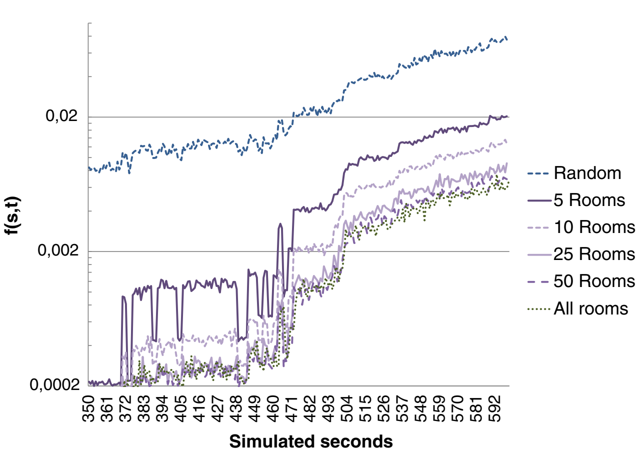

Fig. 10 Experiment results in realistic environments starting from random locations with ad-hoc networks where only part of the sensor data is available

5 Conclusion

This paper presents an application using Ant Colony Optimisation (ACO) for finding safe escape routes in emergency situations. ACO is used in five distinct environments. Firstly, ACO operates in a stationary environment where it quickly reaches a near optimal solution. Secondly, ACO is run in dynamic situations where hazards rapidly change. Further, ACO is evaluated in three realistic environments: fire on board a ship without smoke, fire on board a ship with smoke and fire on board a ship in an ad-hoc network scenario. The realistic scenarios are achieved through set ups from the Fire Dynamics Simulator tool.

In all tested scenarios ACO is empirically able to reach a near optimal solution. In some setups, ACO with 1,000 ants find the optimal route in almost all situations. This leads to the unproven assumption that given enough ants, ACO will always find the optimal escape route. In our further work, we plan to formally verify that whether assumption is correct.

In real evacuation situations, most people escape together in groups with their family and friends. Further, panic tends to spread in emergency situations which in practice means that many will not follow any provided escape routes. These situations are also planned for further testing.

Additionally, we plan to examine methods for faster convergence of ACO including testing the min-max variants of ACO, and examining the possibilities of a multi layered approach.

Most importantly, we are currently implementing a smartphone application for safe escape planning that will use the ACO planning.

References

1. Li Q, Rus D (2005) Navigation protocols in sensor networks. ACM Trans Sens Netw (TOSN) 1(1):3–35
2. Li Q, De Rosa M, Rus D (2003) Distributed algorithms for guiding navigation across a sensor network. In: Proceedings of the 9th annual international conference on Mobile computing and networking. ACM, pp 313–325
3. Radianti J, Granmo OC, Bouhmala N, Sarshar P, Yazidi A, Gonzalez J (2013) Crowd models for emergency evacuation: a review targeting human-centered sensing. In: Proceedings of the 2013 46th Hawaii International Conference on System Sciences (HICSS). IEEE, pp 156–165
4. Thompson PA, Marchant EW (1995) A computer model for the evacuation of large building populations. Fire Saf J 24(2):131–148
5. Thompson PA, Marchant EW (1995) Testing and application of the computer model simulex. Fire Saf J 24(2):149–166
6. Hamacher H, Tjandra S (2002) Mathematical modelling of evacuation problems — a state of the art. Pedestrian and Evacuation Dynamics 2002:227–266. http://www.springer.com/mathematics/applications/book/978-3-540-42690-5
7. Kim D, Shin S (2006) Local path planning using a new artificial potential function composition and its analytical design guidelines. Adv Robot 20(1):115–135

8. Braun A, Bodmann B, Musse S (2005) Simulating virtual crowds in emergency situations. ACM, pp 244–252
9. Li F. Y., Kure Ø. (2005) Optimal physical carrier sensing range in multirate wireless ad hoc networks: analytical versus realistic In: Wireless conference 2005-next generation wireless and mobile communications and services (European Wireless), 11th European, VDE, pp 1–7
10. Liu S, Yang L, Fang T, Li J (2009) Evacuation from a classroom considering the occupant density around exits. Phys A: Statistical Mechanics and its Applications 388(9):1921–1928
11. Wang J, Lo S, Sun J, Wang Q, Mu H (2012) Qualitative simulation of the panic spread in large-scale evacuation. Simulation
12. Helbing D, Farkas I, Vicsek T (2000) Simulating dynamical features of escape panic. Nature 407(6803):487–490
13. Jin T (2002) Visibility and human behavior in fire smoke. In: DiNenno PJ, Drysdale D, Beyler CL, Walton WD, Custer RLP, Hall JRJr, Watts JMJr (eds) The SFPE handbook of fire protection engineering, 3rd edn. Society of Fire Protection Engineers/National Fire Protection Association, Quincy, Massachusetts
14. Purser D (2008) Assessment of hazards to occupants from smoke, toxic gases, and heat. In: SFPE handbook of fire protection engineering, vol 4, pp 2–96
15. HSE Indicative human vulnerability to the hazardous agents present offshore for application in risk assessment of major accidents. Online: Accessed 10 Dec 2013
16. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–39
17. Gutjahr W. (2000) A graph-based ant system and its convergence. Futur Gener Comput Syst 16(8):873–888
18. Liu B, Abbas H, McKay B (2003) Classification rule discovery with ant colony optimization, vol 2003. IEEE, pp 83–88
19. Abolhasan M, Wysocki T, Dutkiewicz E (2004) A review of routing protocols for mobile ad hoc networks. Ad Hoc Netw 2(1):1–22
20. Dorigo M, Stützle T (2010) Ant colony optimization: overview and recent advances. Handbook of metaheuristics, pp 227–263
21. Ducatelle F, Di Caro G, Gambardella L (2006) An analysis of the different components of the anthocnet routing algorithm. Ant Colony Optimization and swarm intelligence, pp 37–48
22. Murtala Zungeru A, Ang L (2012) Performance evaluation of ant-based routing protocols for wireless sensor networks
23. Goodwin M, Granmo OC, Radianti J, Sarshar P, Glimsdal S (2013) Ant colony optimisation for planning safe escape routes. In: Recent trends in applied artificial intelligence. Springer, pp 53–62
24. Dijkstra EW (1959) A note on two problems in connexion with graphs. Numer Math 1(1):269–271
25. Di Caro G, Dorigo M (2011) Antnet: distributed stigmergetic control for communications networks. arXiv preprint arXiv:1105.5449
26. Di Caro G, Ducatelle F, Gambardella L (2005) Anthocnet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks. Eur Trans Telecommun 16(5):443–455
27. Dressler F, Koch R, Gerla M (2012) Path heuristics using aco for inter-domain routing in mobile ad hoc and sensor networks. Bio-inspired models of network, information, and computing systems, pp 128–142
28. Selvan G, Pothumani S, Manivannan R, Senthilnayaki R, Balasubramanian K (2012) Weakness recognition in network using aco and mobile agents. In: 2012 International Conference on Advances in Engineering, Science and Management (ICAESM). IEEE, pp 459–462
29. McGrattan K, Forney GP (2008) Fire dynamics simulator (version 5), user's guide, vol 1019. NIST special publication, pp 1–186
30. Chow WK (1995) Use of computational fluid dynamics for simulating enclosure fires. J Fire Sci 13(4):300–334

Morten Goodwin was born in Nøtterøy, Norway. He acquired his M.Sc. and B.Sc. at Agder University College, Norway 2005 and 2003 respectfully. He received his Ph.D. From Aalborg University in 2011 with the thesis "Towards Automated eGovernment Monitoring". The Ph.D. was on applying learning algorithms and web-mining techniques on eGovernment indicators which are difficult to measure automatically. He is currently an Associate Professor at the Department of ICT, University of Agder, Norway. His research interests include datamining, optimisation, machine learning and software development. Additionally, he has several high ranked scientific publications.

Ole-Christoffer Granmo is Director of the Centre for Integrated Emergency Management (CIEM) and heads the Artificial Intelligence and Its Industrial Applications Group at University of Agder, Norway. He obtained his M.Sc. in 1999 and the Ph.D. degree in 2004, both from the University of Oslo, Norway. His research interests include Intelligent Systems, Stochastic Modelling and Inference, Machine Learning, Pattern Recognition, Reinforcement Learning, Distributed Computing, Computational Linguistics, and Surveillance and Monitoring. Within these areas of research, Dr. Granmo has written more than 80 refereed journal and conference publications. He also serves on the Editorial Board of Crisis Communications, specializing within artificial intelligence support for crisis preparedness and management.

Jaziar Radianti received her Ph.D in System Dynamics from University of Bergen, Norway. Dr. Radianti is a researcher for CIEM (Centre for Integrated Emergency Management) at Department of ICT, University of Agder, Norway. She has served as a reviewer for numerous international conferences. Her research interests include the application of simulation approaches, especially system dynamics, fire dynamics, and Bayesian network modeling for disaster and crisis management. Currently, she is also developing an interest toward research in mobile sensing, and its intersections with human centric computing, machine learning techniques and the application for decision support in a crisis.