POLITECNICO MILANO 1863

Formal Methods For Concurrent and Real-Time Systems

Formal Analysis of Search-and-Rescue Scenarios

Homework Project
A. Y. 2023/2024

Authors:
Giacomo Brunetta (10740202),
Eleonora Cabai (10696943),
Edoardo Carlotto (10730627),
Francesco Santambrogio (10685653)

Instructors:
Prof. Pierluigi San Pietro
Dott.ssa Livia Lestingi

Contents

1 Introduction

1.1 Search-and-Rescue Scenarios

Search and rescue (SAR) is the act of searching and providing help to people in imminent danger through specialized personnel and specialized vehicles or devices. In our project, we examine a SAR scenario set in a room with raging fire. Within this room, there are various types of actors: civilians, who may either need assistance or be capable of helping others (referred to as zero responders); first responders, who are trained professionals providing aid; and drones, which patrol the area to support rescue efforts. The objective of the SAR operation is to ensure the safety of as many civilians as possible within a designated time frame.

1.2 Objective

Our objective is to implement a Network of Timed Automaton (NTA) that simulates the evolution of a search and rescue scenario to check how the decision policies of the actors affect the success of the operation. To do so we implemented both a deterministic and a stochastic model in UPPAAL (v 5.0.0) and tested the properties of the model on different scenarios.

2 UPPAAL Model

2.1 High-Level Model Description

The model consists of a room and the actors moving within it. The room is an N×M grid, where each tile can be an **exit**, a **fire**, or **empty space**. The actors include:

- **Victims**: civilians near the fire who need assistance.
- **Zero responders** or survivors: civilians capable of assisting others or reaching the exit unassisted.
- **First responders**: specialized personnel who rescue victims.
- **Drones**: devices that patrol the area to facilitate rescue operations.

Each entity is represented by a timed automaton, with its behavior dictated by the automaton's evolution. Detailed descriptions of these automata are provided in Section 2.6.

2.2 Scenario Generation

All the scenarios are generated automatically by a Python notebook that performs parametric model generation. The parameters are:

- the grid dimensions
- the number of exits
- the number of instances of each type of entity
- the visibilities of each drone
- the number of groups of contiguous fire tiles.

This approach ensures robust and scalable random scenario generation since it is enough to change the sliders of the chosen parameters in the script. The script produces C/UPPAAL code, which can be copy-pasted into UPAAL, thereby expediting the testing and evaluation process.

2.3 Global Declarations

In this section, we describe the parameters that characterize the scenario and the state variables that describe the status of the system.

2.3.1 System Parameters

Parameter	**Description**
`N`	grid height (number of rows)
`M`	grid width (number of columns)
`SURVIVORS_QTY`	amount of survivors
`VICTIMS_QTY`	amount of victims
`FIRST_RESPONDERS_QTY`	amount of first-responders
`DRONES_QTY`	amount of drones
`T_fr`	helping time of first-responders
`T_zr`	helping time of zero-responders
`T_v`	time after which a victim is a casualty
`drones_visibilities[]`	array of drones' visibilities
`bounds[]`	area bounds for the drones
`p_fail[]`	probability that the sensor of a drone fails
`p_listen[]`	probability that a survivor listens to drones' instructions

2.3.2 State Variables

Parameter	**Description**
`survivors[]`	positions of the survivors
`victims[]`	positions of the victims
`responders[]`	positions of the first responders
`drones[]`	positions of the drones
`survivors_busy[]`	busy status of the survivors
`victims_busy[]`	busy status of the victims
`responders_busy[]`	busy status of the first responders

2.4 Design Assumptions

In this section, we describe the system assumptions that we made in the modeling phase.

The first assumption regards the helping time and the time after which a victim becomes a casualty. In this regard, we assumed that $T_v > T_{zr}$ and $T_v > T_{fr}$, otherwise it is impossible for the victims to be rescued before becoming a casualty.

The second assumption is that **drones know at each time instance the location of all the first responders in the room**. This facilitates the evaluation of the best choice for a drone since it depends on the distance between the detected survivor and the closest first responder, who in general may not be within the drone's visibility range. See Section 2.5 for further details on the drone's policies.

The third consideration regards the drone's decision-making process. There is no need to check the residual surviving time of victims, because **if a zero responder is tasked with assisting a victim, it will always prioritize the nearest one**. If there's insufficient time to aid the closest victim, no other victim can be saved. There's no need to evaluate the chances of saving other victims because the nearest one has the highest chance of being saved.

The fourth assumption is that at each time instant the **zero responders know the position of the closest door**. This is reasonable if we consider that emergency exits are purposely visible in case of emergency. Additionally, the **first responders know the position of the closest victim**, reflecting their expertise in emergencies.

2.5 Moving and Communication Policies

In this model, three types of entities are capable of moving: drones, first responders, and zero responders.

Since they can fly over any entity and any position, drones have a policy (Fig. 1) that depends uniquely on their visibility and their bounds. Every **drone moves back and forth inside its bounds** to ensure that every cell of the map is periodically checked. We generate the **bounds to minimize the overlap** with the patrolling areas of the drones. If overlap does occur, we ensure that a survivor receives instructions from only one drone at a time. The communicating state of the drone is committed while the moving state is not, meaning that transitioning to complete the communication will take precedence over transitioning to a deciding state.

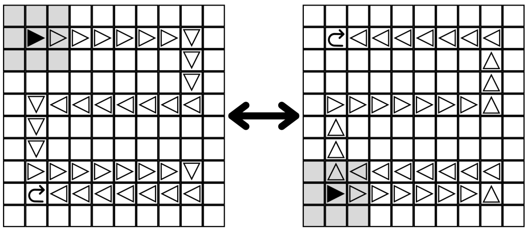

Figure 1: Moving policy for a drone with $N_v = 1$

Additionally, a survivor is marked as busy as soon as a drone begins communicating with them, so **a drone cannot communicate with a survivor who is already committed to helping a victim**. This avoids redundant rescuing requests to the same survivor, or even conflictual ones if made by different drones in case of overlapping visibility ranges of the drones. The drone that begins communication is chosen non-deterministically among those that could enter the communication status.

First responders and zero responders both move with a greedy policy that tries to minimize the Manhattan distance between their position and their goal. In the case of first responders, the goal is to reach the closest victim, while for zero responders the goal is the closest exit, reflecting their respective instinct during emergencies. Zero responders cannot go inside or near a fire, so they never become victims on purpose. First responders can move next to a fire, as we suppose they have fireproof equipment, but not inside.

The requirements do not specify how responders reach a victim. We modeled this behavior as follows: **when a zero responder reaches a first responder to go together to a victim, they instantly occupy the same tile**, simulating the rescuing process where the zero and first responders go into the actual tile of the victim to pick them up. This is a mere detail of implementation that does not contradict the non-overlapping requirement for people.

2.6 Templates

The model pairs each template to an entity and, to simplify the system modeling, civilians are represented by two separate templates: survivors - or zero responders - and victims.

2.6.1 Drones

The timed automaton of drones, at each time unit, patrols its area according to its moving policy shown in Section 2.5 and checks whether there is a pair of an available survivor and a victim within its visibility range. Then, the drone behaves as follows:

- if the distance between the detected survivor and the victim is smaller than the one between the survivor and a first responder plus the one between the first responder and the victim, the drone sends the `rescue` signal to the survivor to directly help the victim. In the stochastic model, there is a probability `p_fail` that the sensors of the drone do not see the survivor.
- if the opposite condition holds true, the drone orders the survivor to go to the first responder using the `callFirstResponder` channel, and orders the first responder to wait for the survivor using the `waitForSurvivor` channel.

2.6.2 First responders

The timed automaton of first responders is modeled to:

- if there is a victim within 1-cell range, help them and send them the `save` signal to notify that they are safe;
- if they receive the `waitForSurvivor` signal, wait for a zero responder to arrive and the signal on the `goToVictim` channel, and reach with them the victim to help;
- if none of the previous, they move following their moving policy.

2.6.3 Civilians

Zero Responders

The timed automaton of zero responders is such that, at each time unit

- if there is an exit within 1-cell range, they go to the `Safe` state;
- if a drone sends a `rescue` signal, they reach the nearest victim, help them, and become `Safe`.
- if a drone sends a `callFirstResponder` signal, they reach the first responder and then go to and help the nearest victim;
- if none of the previous, they move to a nearby cell, avoiding the ones near to a fire (see 2.5)

In the stochastic model, there is a probability `p_listen` that the entity follows the instruction of the drone.

Victims

Victims are civilians with a fire in a 1-cell range, they cannot move and their timed automaton has only three states: `In_need`, `Being_helped` and `Safe`.

- The victim stays in the `In_need` state, until a zero responder is instructed on how to help them or a first responder, is nearby.
- When a zero responder and a first responder starts to reach the victim, or a first responder is in the 1-cell range, the victim transitions to the `Being_helped` state. Being in this state lasts for T_{zr} or T_{fr} time units, depending on the fact that a zero responder is helping alone or a first responder is involved.
- After the assistance has been concluded, the victim transitions to the `Safe` state

If more than T_v time units elapsed since the start of the simulation, the victim becomes a casualty. This happens even after the helping of a responder has started, if the rescue is not complete before the global time reaches T_v. The behavior was modeled this way because T_v time is intended to limit a person's resistance when exposed to high temperatures and toxic smoke near a fire, which are perilous even during rescuing.

3 Analysis and Results

3.1 Properties

We checked different types of properties of the model using the UPPAAL verifier. In particular, the goal is twofold: verify that Search-and-Rescue Scenarios operate as intended and meet the requirements, as well as assess different types of scenarios to see which ones have better performance.

Safety Properties

The safety properties helped us throughout the development of the model to verify it operated as intended and that the implementations met the requirements

1. It is always guaranteed that any **person is never in the fire**:
$\forall\Box((\forall s \in \text{Surv } \neg\text{s.Safe} \implies \text{map[s.x][s.y]} \neq \text{FIRE}) \land$
$(\forall v \in \text{Vict } \neg\text{v.Safe} \implies \text{map[v.x][v.y]} \neq \text{FIRE}) \land$
$(\forall r \in \text{First-Resp map[r.x][r.y]} \neq \text{FIRE}))$
2. It is always guaranteed that **two people are not in the same cell**:
$\forall\Box((\forall \text{s1, s2} \in \text{Surv } (\text{s1} \neq \text{s2} \land \neg \text{ s1.Safe} \land \neg \text{ s2.Safe}) \implies \text{s1.pos} \neq \text{s2.pos}) \land$
$(\forall \text{r1, r2} \in \text{First-resp r1} \neq \text{r2} \implies \text{r1.pos } \neq \text{r2.pos}) \land$
$(\forall s \in \text{Surv}, r \in \text{First-resp } (\neg\text{s.Safe } \land \neg\text{s.Reach\_vict } \land \neg \text{ s.Helping\_with\_FR}) \implies \text{s.pos} \neq \text{r.pos}))$
3. It is always guaranteed that **survivors and victims are outside the map if and only if they are safe** (in Uppaal the iff is not supported, so double implication was used:
$\forall\Box((\forall s \in \text{Surv } \neg\text{s.Safe } \iff \text{s.x} < \text{N} \land \text{s.y} < \text{M}) \land$
$(\forall v \in \text{Vict } \neg\text{v.Safe } \iff \text{v.x} < \text{N} \land \text{v.y} < \text{M}))$
4. It always guaranteed that for **each drone bounds are valid and it never goes out of its bounds**:
$\forall\Box(\forall d \in \text{Drone(d.bound\_min\_x} \geq 0 \land \text{d.bound\_max\_x} < \text{N}) \land$
$(\text{d.bound\_min\_y} \geq 0 \land \text{d.bound\_max\_y} < \text{M}) \land$
$(\text{d.x} \geq \text{d.bound\_min\_x} \land \text{d.x} \leq \text{d.bound\_max\_x}) \land$

(d.y $\geq$ d.bound_min_y $\wedge$ d.y $\leq$ d.bound_max_y))

5. It is always guaranteed that a **survivor never moves near the fire**:
$\forall\Box(\forall s \in \text{Survivors.Moving} \implies (\neg\text{tileIsNearSomething(s.pos, FIRE)}))$

Deadlock Property

This property was used to demonstrate the deterministic model's resilience against deadlocks. In the stochastic model it is not possibile to check this property due to UPPAAL's implementation.
$\forall\Box(\neg$ deadlock).

Performance Properties

Performance properties, including the two mandatory ones, were used to assess the performance of different scenarios, each evaluating a specific aspect of the model.

1. It is **possible for a percentage `Civilians_p_sometimes_safe%` of all civilians to reach a safe state** within time T_{scs}:

$$\exists\Diamond((\sum_{v\in\text{Vict}}\text{v.Safe} + \sum_{s\in\text{Surv}}\text{s.Safe} \geq (\text{Vict}_{tot} + \text{Surv}_{tot}) * \text{Civil}_{sometimes\_safe}) \wedge (T_{global} \leq T_{scs}))$$

Also, we decided to define two properties to **separately verify the percentages of victims and zero responders** that reach a safe state.

- $\exists\Diamond(\sum_{s\in\text{Surv}}\text{s.Safe} \geq (\text{Surv}_{tot} * \text{Surv}_{sometimes\_safe}) \wedge (T_{global} \leq T_{scs}))$
- $\exists\Diamond(\sum_{v\in\text{Vict}}\text{v.Safe} \geq (\text{Vict}_{tot} * \text{Vict}_{sometimes\_safe}) \wedge (T_{global} \leq T_{scs}))$

2. A percentage `Civilians_p_always_safe%` **of all civilians is always guaranteed to reach a safe state** within time T_{scs}:

$$\forall\Diamond((\sum_{v\in\text{Vict}}\text{v.Safe} + \sum_{s\in\text{Surv}}\text{s.Safe} \geq (\text{Vict}_{tot} + \text{Surv}_{tot}) * \text{Civil}_{always\_safe}) \wedge (T_{global} \leq T_{scs}))$$

Also, we decided to define two properties to **separately verify the percentages of victims and zero responders** that reach a safe state.

- $\forall\Diamond(\sum_{s\in\text{Surv}}\text{s.Safe} \geq (\text{Surv}_{tot} * \text{Surv}_{always\_safe}) \wedge (T_{global} \leq T_{scs}))$
- $\forall\Diamond(\sum_{v\in\text{Vict}}\text{v.Safe} \geq (\text{Vict}_{tot} * \text{Vict}_{always\_safe}) \wedge (T_{global} \leq T_{scs}))$

3. A percentage `Civilians_p_helping%` **of all civilians is always guaranteed to help a victim** before exiting the map:

$$\forall\Diamond(n_{helping\_civilians} \geq (\text{Surv}_{tot} * \text{Civil}_{p\_helping}))$$

The same properties were evaluated in the stochastic model as well, since they already exhaustively verify the model, by simply adding the *Pr* keyword. In particular, the temporal upper bound for the safety properties is 1000, while for the performance properties is T_{scs}.

These properties were used to yield some performance metrics used to evaluate different configurations and scenarios that will be described in the next section.

Parameter	Description
`Civilian Always Safe`	% of civilians that always reaches a safe state within T_{scs}
`Survivor Always Safe`	% of survivors that always reaches a safe state within T_{scs}
`Victim Always Safe`	% of victims that always reaches a safe state within T_{scs}
`Civilian Sometimes Safe`	% of civilians that eventually reaches a safe state within T_{scs}
`Survivor Sometimes Safe`	% of survivors that eventually reaches a safe state within T_{scs}
`Victim Sometimes Safe`	% of victims that eventually reaches a safe state within T_{scs}
`Helping Rate`	% of survivors that helps a victim

3.2 Scenarios

For evaluating the properties, we have considered the following parameter values: $T_{fr} = 5$, $T_{zr} = 7$, and $T_v = 12$. The value of T_v is not so high due to the low resistance of non-equipped civilians to very high temperatures. T_{fr} is smaller than T_{zr} since first responders have more experience in rescuing. Moreover, T_{scs} is always set with a positive offset added to T_v, since otherwise no victims are checked as safe within T_{scs}. We then tuned T_v and T_{scs} accordingly to have reasonable values. For the stochastic model, we set `p_listen` $= 0.85$ for all survivors and `p_fail` $= 0.1$ for all drones to equally assess the performance of specific characteristics in different scenarios. Note that these parameters can arbitrarily be configured for all entities separately.

It is worth pointing out that to check the existence properties, the traces have been explored in a Depth First Search Order to speed up the verifier since for the existence it is enough to verify one valuable trace by definition. Instead, for the for-all properties Breadth-First Search, UPPAAL's default, is fine since all traces need to be explored in parallel.

3.3 First Testcase

In the first testcase (Figure 2), we examine a 10x10 grid containing 10 civilians, 6 of whom need assistance, along with 2 first responders. We compared the satisfaction percentages of each property under three different scenarios: one drone with a visibility range $N_v = 4$, 2 drones with $N_v = 2$, and 4 drones with $N_v = 1$. The goal was to analyze how performance varies with different numbers of drones and their visibility ranges within the same testcase.

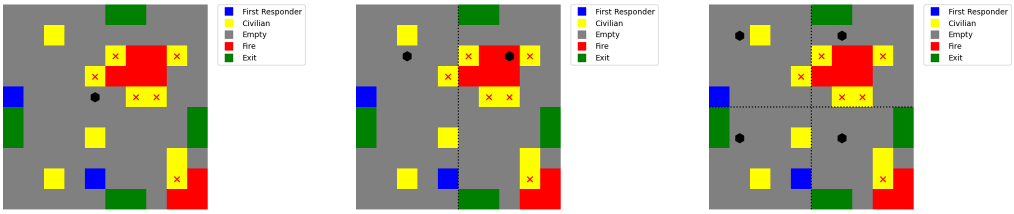

Figure 2: From the left: 1 drone with $N_v = 4$, 2 drones with $N_v = 2$, 4 drones with with $N_v = 1$

Performance Evaluation of the Deterministic Model

Figure 3 shows that as the number of drones on the map increases, the percentage of always-safe civilians, as well as victims alone, rises, while the percentage of civilians receiving help decreases.

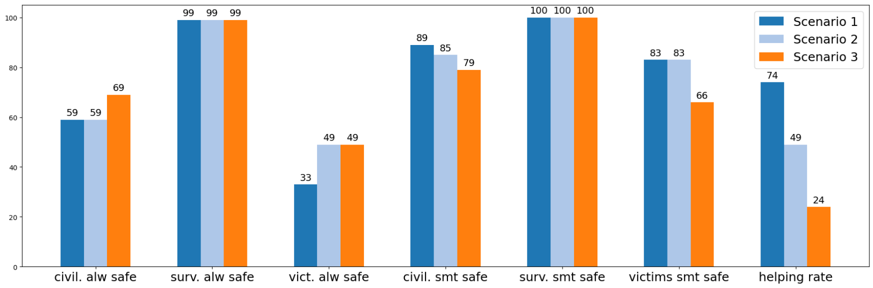

Figure 3: Results of the deterministic model for scenarios 1, 2 and 3

Performance Evaluation of the Stochastical Model

All the properties are satisfied with 95% of confidence and a probability ≥ 0.97505. Moreover, from Figure 4 it can be noticed that the percentage of always-safe civilians increases and the helping percentage of civilians decreases with the increase of the number of drones in the map.

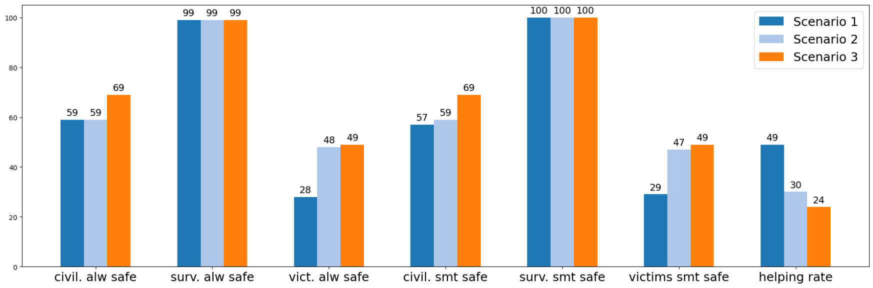

Figure 4: Results of the stochastic model for scenarios 1, 2 and 3

3.4 Second Testcase

In the second testcase (Figure 5), we consider a 10x10 grid with 8 civilians in need and 2 drones. We compared the satisfaction percentages of each property under two different scenarios: one with 6 first responders and 2 zero responders, and the other with 2 first responders and 6 zero responders. The objective was to analyze how performance varies with different proportions of zero and first responders.

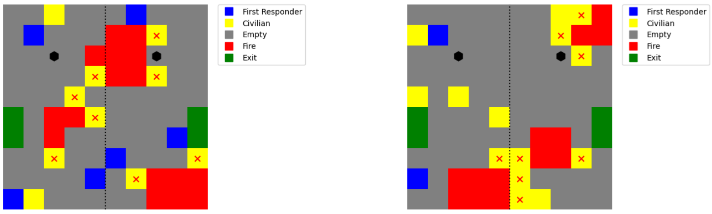

Figure 5: on the left: 6 first-responders and 2 survivors; on the right: 2 first-responders and 6 survivors

Performance Evaluation of the Deterministic Model

From the graph below it is obvious that in Scenario 4, where there are more first-responders, the percentage of saved victims is substantially higher, whereas in Scenario 5 the values are lower due to the higher number of survivors.

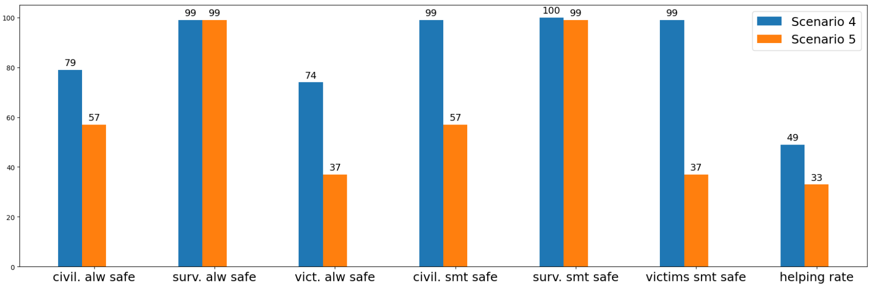

Figure 6: Results of the deterministic model for scenarios 4 and 5

Performance Evaluation of the Stochastical Model

All the properties are satisfied with 95% of confidence and a probability ≥ 0.97505.

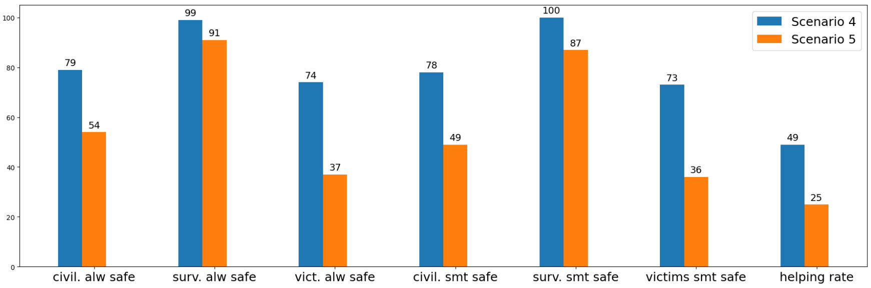

Figure 7: Results of the stochastic model for scenarios 4 and 5

3.5 Third Testcase

The third testcase (Figure 8) takes into consideration two different scenarios with different map sizes (10x10 vs 20x20) but an equal number of entities, namely 8 victims, 4 first-responders, 4 survivors, and 2 drones with $N_v = 2$. The objective here was to see how the different map configurations, and in particular their dimensions, affect the performance of the model.

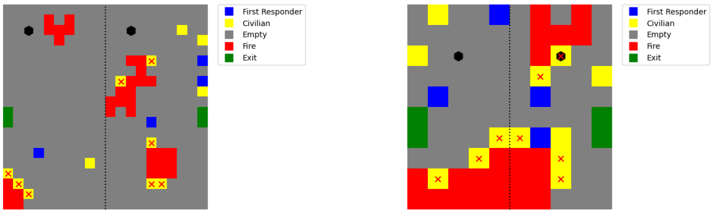

Figure 8: on the left: 20x20 room; on the right: 10x10 room

Performance Evaluation of the Deterministic Model

From the evaluation of the Deterministic Model, it can be deduced that a 20x20 grid has a better rate of saved victims and civilians, whereas the rate of saved survivors is similar.

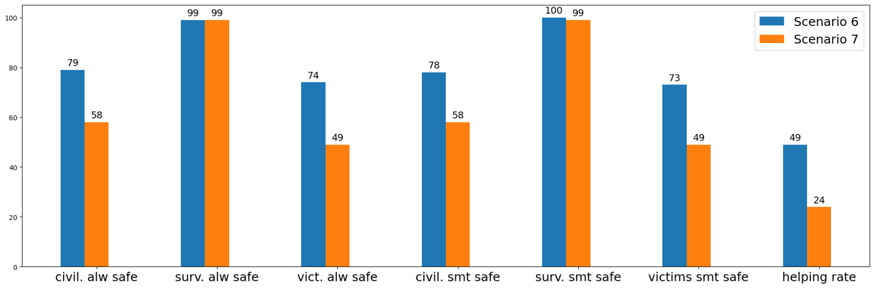

Figure 9: Results of the deterministic model for scenarios 6, 7

Performance Evaluation of the Stochastical Model

From Figure 10 it can be noticed that the rates of saved civilians are higher in the 20x20 grid, except the helping rate which are similar in the 10x10 grid.

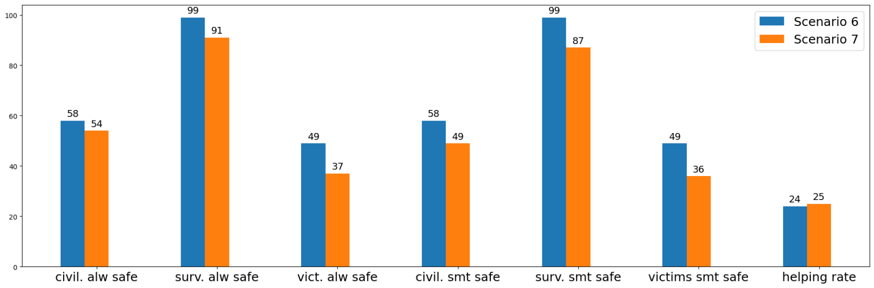

Figure 10: Results of the stochastic model for scenarios 6, 7

4 Conclusions

This Search-And-Rescue model is designed to be general and scalable, allowing all parameters to be adjusted as needed. We verified that it functions as intended and meets all key requirements by checking safety and deadlock properties. Additionally, we evaluated its performance by comparing different configurations through various performance properties.

Our analysis indicates that, for a given configuration, increasing the number of drones while decreasing their visibility range is the optimal choice for maximizing the number of safe civilians. However, this leads to a decrease in the helping rate, as civilians are instructed less frequently by drones to assist victims and instead focus more on finding the exit and saving themselves according to their policy. Furthermore, having more first responders results in a higher number of safe civilians compared to having more zero responders. This outcome is realistic, as first responders prioritize saving as many victims as possible, while zero responders tend to focus on escaping and saving themselves, reflecting their respective movement policies. Additionally, scenarios with a larger area perform better than those with half the size but the same number of entities. This might seem counterintuitive since a larger area requires responders to travel longer distances, thereby taking more time to save a victim. However, as the helping rate suggests, in larger settings, drones more frequently and easily intercept survivors on their way to the exit.

In summary, the model is most effective with a larger number of drones with shorter visibility ranges, a greater number of first responders, and a larger operational area, all contributing to maximizing the number of civilians saved which is the ultimate goal of the system.