Modeling and simulation of complex systems

Project 4: Evacuation

Vu Trung Dung

January 2, 2025

Question

How to better manage the evacuation of a population on a beach in a tsunami context?

- flooding will not be modeled by itself
- just **the behavior of residents** in the face of the threat.

Modeling and Simulation

Situation

- People will only evacuate if they have been informed of the flooding.
  - We assume that only **10% of the population is informed at the beginning of the simulation**.
  - A person observing someone evacuating (at a distance of less than 10m) will have a probability of 0.1 of evacuating in turn.
- **Not all residents know where to evacuate** and only 10% will go directly to the shelter.
- People have multiple mobilities of evacuation: by car, by bike, or on foot.

Modeling and Simulation

Situation - My extension

- **The knowledge of the evacuation shelter can be transfered across people**.
  - The knowledge of the evacuation shelter can be shared with 2 person at 10% probability when they meet each other on the road while evacuating.
  - When the knowledge is shared, 2 person have 10% probability to change their evacuation to the closest shelter they're heading to.

Modeling and Simulation

Strategies to aware of flooding

Different strategies of aware of flooding to the 10% of the population:

- random.
- furthest from the shelter.
- closest to the shelter.

Find the most effective of these strategies in terms of:

- number of evacuees.
- evacuation time.
- time for the total evacuation/time spent on the roads.

GIS Data

In this project, we will use the GIS data of Hanoi and the Red River in the previous exercises.

Figure: GIS map for the simulation

Implementation (GAMA)

Extensions

- **Extensions 0**: GIS map, population, evacuation shelter, roads, flooding simulation, etc.
- **Extensions 1**: The evacuating behavior of the population.
- **Extensions 2**: Multimobility of population (car, bike, foot).
- **My Extensions**: The knowledge of the evacuation shelter can be transfered across people.
- **Extensions 3**: Experiment and analyze the effectiveness of different strategies of aware of flooding.

Implementation: Extensions 0

The Map

- Hanoi and Red River GIS map.
- The **Evacuation Shelter** is the N largest building in the map (red color). The sample code:

```
evacuations <- 8 first (
    building sort_by -each.shape.
        area
);
ask evacuations {
    self.is_evacuation <- true;
}
```

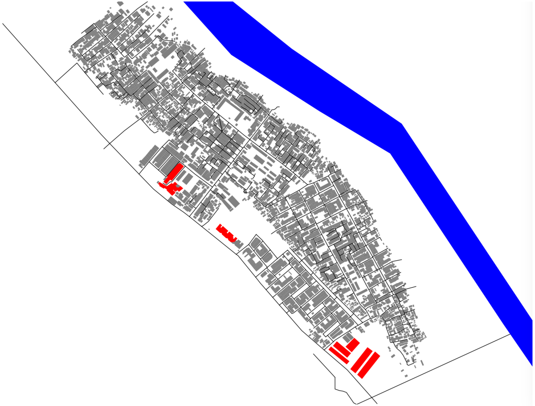


Figure: Map and Evacuation Representation

Implementation: Extensions 0

Species

- **People**: the inhabitants of the city.
- **Evacuation Shelter**: the shelter for evacuation.
- **Road**: the road for evacuation.
- **Building**: the building in the city.
- **Flooding**: the flooding area.

Implementation: Extensions 0

Species - Flooding

Implement the flooding simulation with the following parameters:

- **Flooding date**: the start date of flooding.
- **Grow rate and flooding speed**: how fast the flooding grows.

```
float grow_rate <- 0.5;
float flooding_speed <- 0.1;

reflex expand when: flooding_date <= current_date and every(1#m)
{
    grow_rate <- grow_rate + flooding_speed;
    shape <- shape + grow_rate;
}
```

Implementation: Extensions 1

Species - Initial Population

- No more 5 people in a building (customizable).
- People are located randomly in the city except the evacuation shelters.

```
create inhabitant number: 1000 {
    location <- any_location_in(one_of(building));

    ask any(building where (!each.is_evacuation
        and length(each.inhabitants) <
        max_n_inhabitants_in_building)) {
        self.inhabitants << myself;
        myself.location <- any_location_in(self);
    }
}
```

Figure: People in building

Implementation: Extensions 1

Species - Aware of Flooding

- Only 10% of the population know about the shelter at the beginning.

```
ask (int(percentage_of_people_known_shelter * length(inhabitant)) among
    inhabitant {
    target_shelter <- evacuations closest_to self;
}
```

- Only 10% of the population is informed about the flooding.

```
reflex flooding_announce when: flooding_inform_date <= current_date and !
    flooding_is_informed {
    flooding_is_informed <- true;

    ask (int(percentage_of_people_are_informed * length(inhabitant))
        among inhabitant {
        is_evacuating <- true;
    }
}
```

Implementation: Extensions 1

Species - People behavior

```
reflex observe_evaculating when: !is_evacuating and flip(
    percentage_of_following_evaculating) every(5#s) {
    if !empty((inhabitant where each.is_evacuating) at_distance 20#m)  {
        is_evacuating <- true;
    }
}

reflex find_shelter when: target = nil and is_evacuating every(5#s) {
    building target_building <- target_shelter;
    if (target_building = nil) {
        target_building <- one_of((building - visited_buildings) where each.
            is_safe);
    }
    visited_buildings << target_building;
    target <- any_location_in(target_building);
}
```

Implementation: Extensions 1

Species - Evacuation behaviour

```
reflex evacuating_people when: is_evacuation every(5#s) {
    ask (inhabitant at_distance 20#m) {
        target_shelter <- myself;
        target <- any_location_in(target_shelter);
    }
    ask (inhabitant at_distance 0.5#m) {
        number_evacuted_people <- number_evacuted_people + 1;
        do die;
    }
}
```

Implementation: Extensions 2

Multimobility of population

Initialize inhabitants with different evacuation modalities:

```
int count <- 0;
create inhabitant number: nb_of_people {
    count <- count + 1;
    if count < nb_of_people * percentage_of_car {
        traffic_weight <- traffic_weight_factor * 5;
        speed <- 10 * pedestrians_speed;
    } else if count < nb_of_people * (percentage_of_car + percentage_of_bike)
        {
        traffic_weight <- traffic_weight_factor * 2.5;
        speed <- 8.5 * pedestrians_speed;
    } else {
        traffic_weight <- traffic_weight_factor;
        speed <- pedestrians_speed;
    }
}
```

Implementation: Extensions 2

Road and Traffic Weight

- Calculate the traffic weight of each road.

```
float capacity <- 1 + shape.perimeter/10;
float total_traffic_weight <- 0.0
    update: sum((inhabitant at_distance 1) collect each.traffic_weight);
float speed_rate <- 1.0
    update: exp(-total_traffic_weight/capacity) min: 0.1;
```

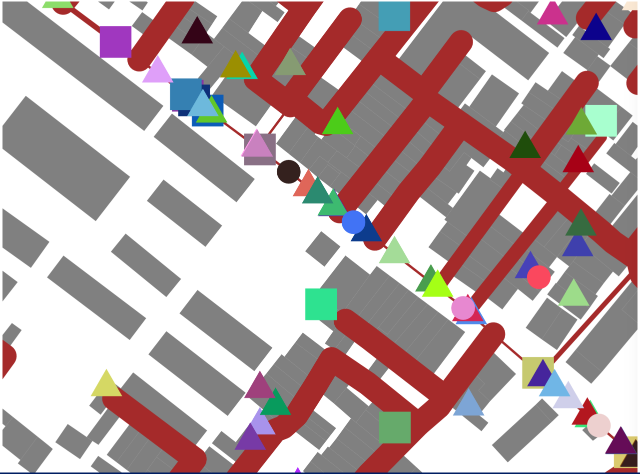
- Update the weight of the road network.

```
reflex update_speed {
    road_weights <- road as_map (each::each.shape.perimeter / each.
        speed_rate);
}
```

Implementation: Extensions 2

Representation of different evacuation mobilities

- **Car**: big squares.
- **Bike**: triangle.
- **Foot**: small circle.



Implementation: My Extensions

Transfer knowledge of evacuation shelter

```
reflex share_shelter when: is_evacuating and every(5#s)
        and flip(percentage_of_share_shelter) {

    ask inhabitant at_distance 10#m {
        building shelter <- [myself.target_shelter, self.target_shelter]
            closest_to myself;
        if (shelter != nil) {
            is_evacuating <- true;
            target_shelter <- shelter;
            target <- any_location_in(shelter);
            myself.target_shelter <- shelter;
            myself.target <- any_location_in(shelter);
        }
    }
}
```

Implementation: Extensions 3

Trategies Implementation (1)

```
reflex flooding_announce when: flooding_inform_date <= current_date and !
    flooding_is_informed {
    flooding_is_informed <- true;

    int nb_informing_people <- int(percentage_of_people_are_informed * length(
        inhabitant));

    if (initial_inform_strategy = "random") {
        ask nb_informing_people among inhabitant {
            is_evacuating <- true;
        }
    } else if (initial_inform_strategy = "furthest") {
        ...
}
```

Implementation: Extensions 3

Trategies Implementation (2)

```
reflex flooding_announce when: flooding_inform_date <= current_date and !
    flooding_is_informed {
    ...
    } else if (initial_inform_strategy = "furthest") {
        ask inhabitant {
            distance_to_shelter <- max(evacuations collect distance_to(self,
                each));
        }
        ask nb_informing_people first (inhabitant sort_by -each.
            distance_to_shelter) {
            is_evacuating <- true;
        }
    } else {
    ...
}
```

Implementation: Extensions 3

Trategies Implementation (3)

```
reflex flooding_announce when: flooding_inform_date <= current_date and !
    flooding_is_informed {
    ...
    } else {
        ask inhabitant {
            distance_to_shelter <- min(evacuations collect distance_to(self,
                each));
        }
        ask nb_informing_people first (inhabitant sort_by each.
            distance_to_shelter) {
            is_evacuating <- true;
        }
    }
}
```

Experiment

Comparison

- Run 3-simulations in parallel.
- Draw a chart to compare number of evacuees in realtime.

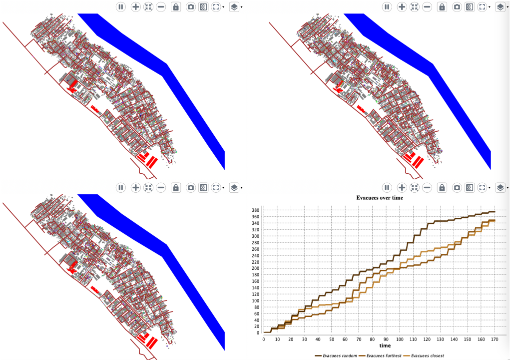


Figure: Multiple simulation

Experiment
Comparison - Result (1)

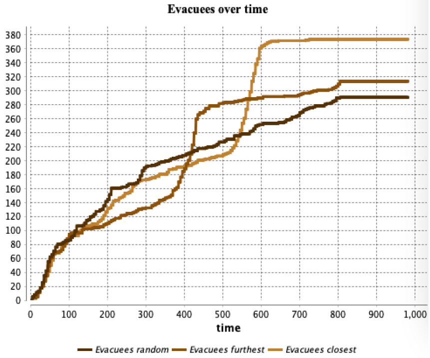


Figure: number of people = 500

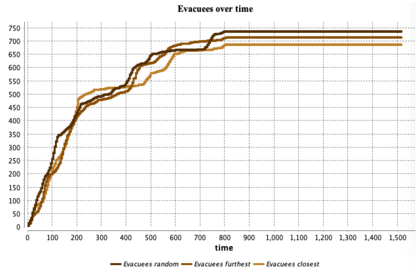


Figure: number of people = 1000

Experiment

Comparison - Result (2)

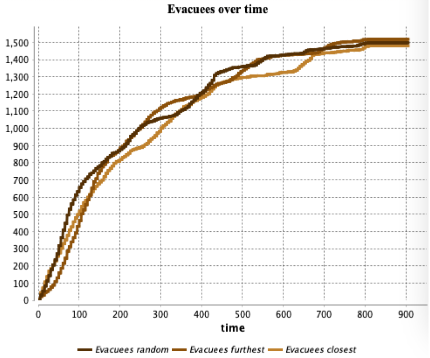


Figure: number of people = 2000

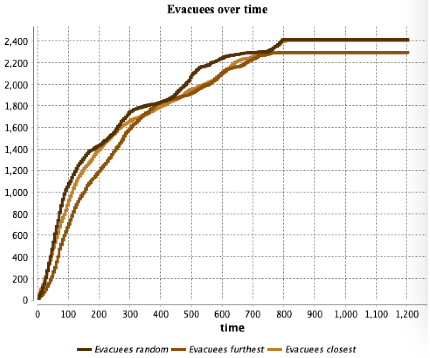


Figure: number of people = 3000

Experiment

Comparison - Conclusion

- The **random strategy** is the most effective on term of the number of evacuees.
- The **closest strategy** is the most effective on term of the evacuation time.
- If we have enough number of people and enough time, there is no difference between the strategies.

Experiment

Batch exploration

TODO

Q & A