Multi-Agent Pathfinding for Online Emergency Evacuation

Supervisor: Álvaro Torralba, alto@cs.aau.dk

August 31, 2020

Emergency evacuation plans establish a pre-determined route to enable everyone to exit some space in case of risk. Evacuation plans often specify the paths that agents should take in case of an emergency, establishing predetermined routes from every point in the map to an exit.

In this project, we aim to design personalized evacuation plans, where once the emergency has occurred, the path for each agent is computed in a way that minimizes the risk for all agents. Thus, the plan is computed online, taking into consideration the current location of every agent, as well as specifics of the emergency such as which zones involve a larger risk. The plan can be notified to each agent (e.g., in their mobile device). This allows for much better plans than the offline approach, since bottlenecks and high-risk zones can be better avoided. The objective is to develop a tool that can be applied to different scenarios like evacuation from a single building during a fire emergency, or evacuation of an entire city in case of Tsunami.

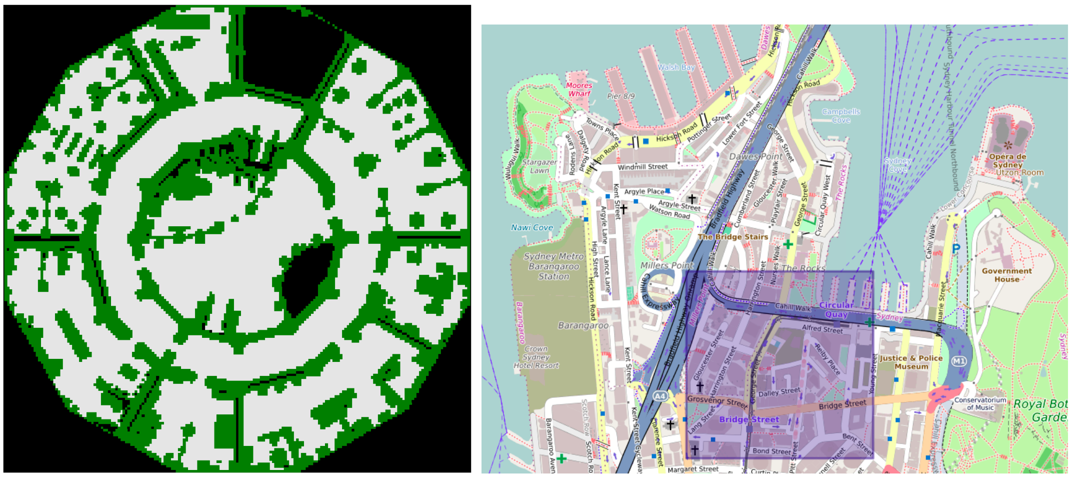

Figure 1: Maps to evaluate the online evacuation plans: from a videogame (left) or a city (right) — see https://movingai.com/benchmarks/grids.html

This is related to the problem of multi-agent pathfinding (MAP), where given a set of agents at a given location, one must find the path the agents should take to reach their target location. Usually, most approaches attempt to minimize the sum of time steps by each agent or the maximum makespan. The problem in this project is related to MAP but not identical. Agents start in a “danger zone” and their target is to reach a “safe zone”. The objective is to minimize the overall risk, i.e. the sum of the risk taken by each involved agent.

Your goal is to develop a program that anyone can use for their own scenarios, either designing specific heuristic search algorithms, or by using some existing tool like planning or ASP. Assuming as input a map like the ones in the figure above, and a list of agents and their current locations, your program must output the path that each agent must take. You will also create a benchmark set in order to test how your program scales on scenarios of increasing size.