Andrew Sack
December 19, 2019
ME 149: Collaborative Robotics
Final Report

Modelling Firefighting Search and Rescue Operations

Introduction:

The purpose of this simulation was to create a model of multi-agent search and rescue that could be used to inform a real-life search and rescue exercise. The exercise was a firefighting scenario where both human and drone rescuers were used to rescue targets in a room that is being consumed by fire. This simulation was used to explore different distributions of team resources to see what strategy led to the highest number of targets rescued. The simulation uses multiple different modelling tools including agent-based modelling, path planning, and probabilistic detection of targets.

Methods:

The search and rescue scenario was simulated in MATLAB using agent-based modelling. Each searcher (one drone and two humans) was modelled as an agent. The search domain was also modelled as a separate agent. Agents were implemented as classes in MATLAB's object-oriented programming structure, thus each agent contained both properties and methods.

The room design was based heavily on the layout and dimensions of the lab that was used as the search domain for the real-life exercise. A photo and blueprint of the search domain are shown in Figure 1 and Figure 2, respectively. The control room is marked with a red circle in Figure 2. Human agents entered the room from the door in this room. Multiple different representations of the room were used within the class for different aspects of the simulation. A coarse graph-based representation was used to assist with path planning. The room was also modelled as a series of polygons for more aesthetically pleasing visuals as well as for collision detection.

Figure 1: Photo of Search Domain

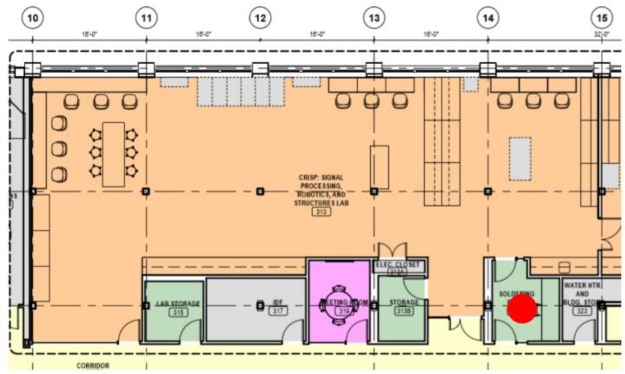

Figure 2: Blueprint of Search Domain

There are several key elements of the search domain included in the model. First of all, the outer boundary of the room is modelled by the white polygon in Figure 3. The obstacles in the search domain were simplified, such that only a few were included, and they were characterized as either tall obstacles or short obstacles. In Figure 3, tall obstacles are represented in blue and short obstacles are represented in orange. Tall obstacles are impassible by both human and drone agents, while short obstacles can by flown over by drone agents, but not by human agents.

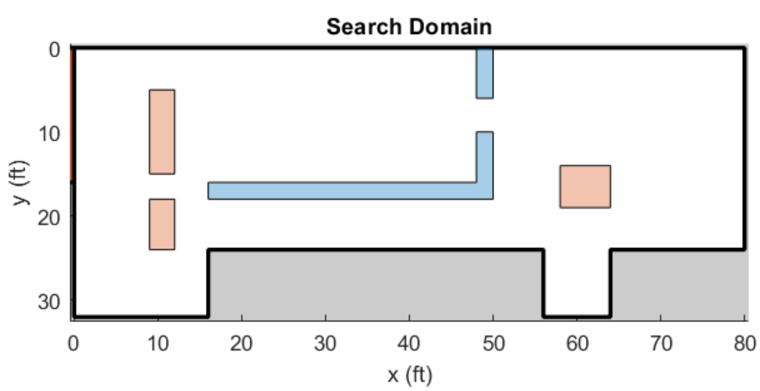

Figure 3: MATLAB Representation of Search Domain (Blue are tall obstacles; Orange are short obstacles)

There is also a piece of comprised floor running along the y-axis at x = 16 ft. One half of the beam is passible while the other half will incapacitate a rescuer if crossed. The safe side is determined randomly each trial and unknown to the human agents. The drone must reach a point on the other side of the line to inspect the floor and discover which side is safe. Before the drone checks the beam, both sides are assumed to be impassible by the human agents due to the deadly consequences of crossing the incorrect side. The beam is initially represented as a yellow line and, after the drone inspects it, the safe side turns green and the dangerous side turns red as shown in Figure 4.

The last component of the search domain model is the fire. The fire is modelled as a polygon spanning from the left wall of the room to a flame front. The flame front is determined by a series of equations that move control points further into the room every second, while also including some Gaussian noise. The fire is impassible by drone agents and human agents use twice as much endurance while inside the fire.

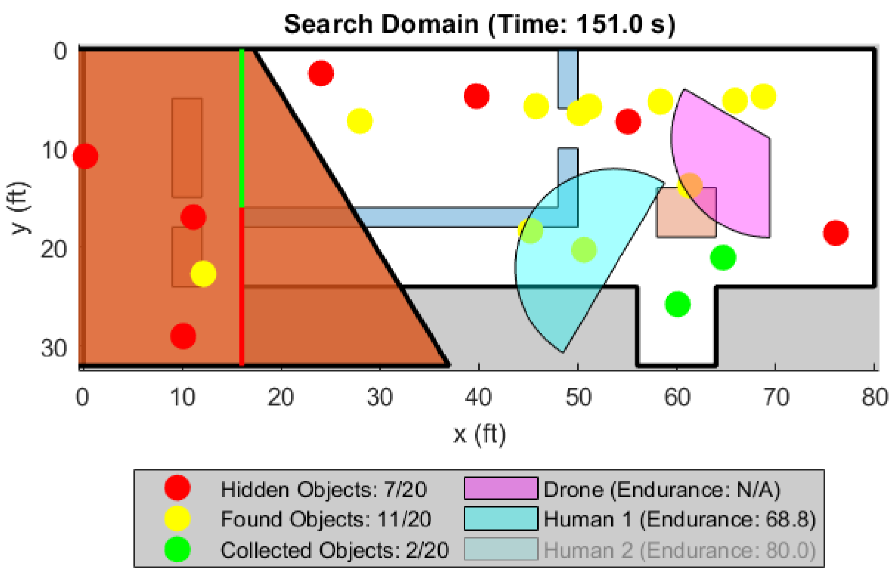

Figure 4: MATLAB Visualization of Simulation (Green line is safe floor; Red line is compromised floor; Red-Orange polygon is the fire)

Inside the search domain are 20 objects to be rescued by the search agents. Objects are represented as points within the search domain. The points were randomly determined and then held constant for the duration of the testing. Two different sets of points were tested and their locations are shown in Figure 5 and Figure 6. Additionally each point had a random probability uniformly distributed between 0.2 and 1 associated with it. This probability is to represent how well hidden that object is. Objects also had a state associated with them. Initially all objects were hidden. They then transitioned to found and collected.

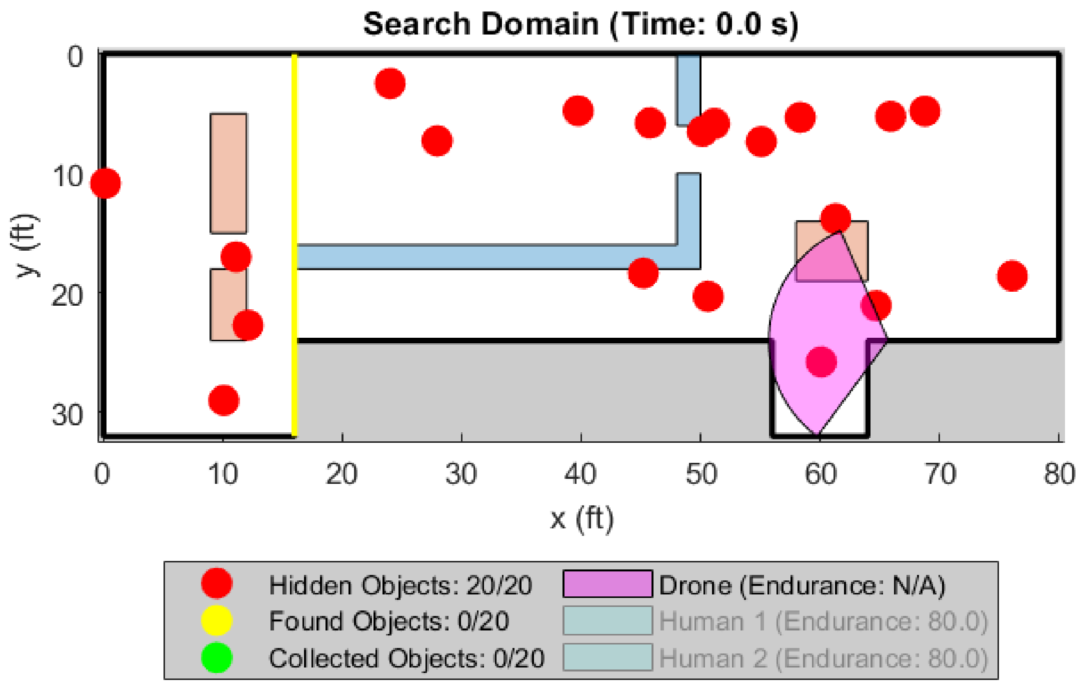

Figure 5: Set 1 of Object Locations

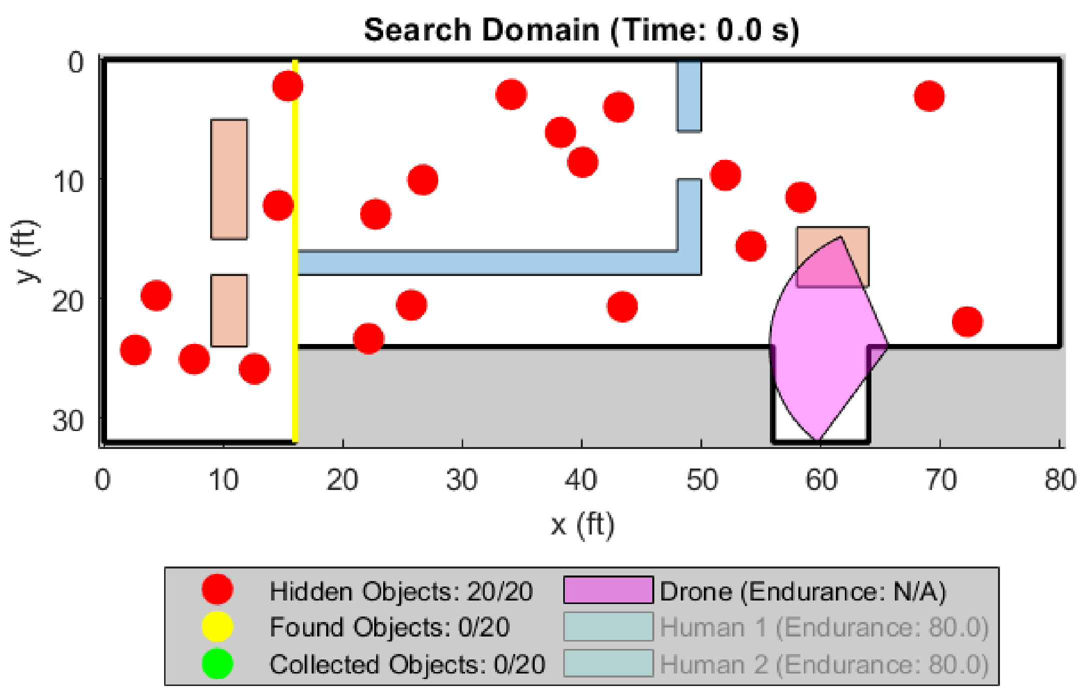

Figure 6: Set 2 of Object Locations

There are two types of searcher agents: human and drone. Both are generally similar with different parameter values, as listed in *Table 1*, but similar behavior. Agents navigate from point to point using a modified A* algorithm. The search domain represented as a graph with 1 ft wide nodes. A* is performed on this graph to produce a valid path. However, this path often includes many turns, which impacts the view region of the agents, so the path is then simplified by drawing the longest possible straight lines between nodes and removing extraneous nodes.

Table 1: Parameter Values of Human and Drone Agents

	Human Agent	**Drone Agent**
View Radius	10 ft	10 ft
Field of View	180 degrees	120 degrees[1]
Velocity	2 ft/s	3 ft/s
Detection Odds	0.075 per 0.1 sec	0.05 per 0.1 sec

There are some key differences between human agents and drone agents. Drone agents have different restrictions to their motion since they can fly over short obstacles and cannot enter the fire. Drones also cannot collect objects and do not have an endurance limit. Human agents can collect objects by remaining within two feet of a found object for one second. They can pick up 5 objects before needing to return to the entrance to drop off the objects. They also have an endurance limit of 80 units. 1 endurance unit is used per second that the agent is in the search domain, with 2 units used per second inside the fire. Human agents will move to the closest found, but not collected object, looking for objects along the way. If there are no found objects, the agent will then move to a random point. Drone agents will first move to the compromised floor to determine which side is safe. Then, they will move to random points, looking for objects along the way. Object detection is modelling probabilistically, with each agent having a probability of detecting an object within its view region each time step. This probability is the product of the agent's detection odds and the object's hide quality probability, as described earlier.

Two different resource distribution strategies were tested. The humans-only strategy uses two human agents and no drone. The humans are both released immediately. The simulation of this strategy was only run for 80 seconds, as that was the maximum amount of time the agents could perform with the endurance limit. The drone-assisted strategy first releases the drone which searches the room by traversing from random point to random point. Then, at 140 seconds (160 seconds remaining), the first human is released and at 220 seconds (80 seconds remaining), the second human is released. This strategy runs for 300 seconds.

A Monte-Carlo simulation of 100 trials was performed on the model. The humans-only strategy and drone-assisted strategy were tested with two different sets of object locations. A T-Test was then performed to analyze the difference between the two strategies.

Results:

Objects Found:

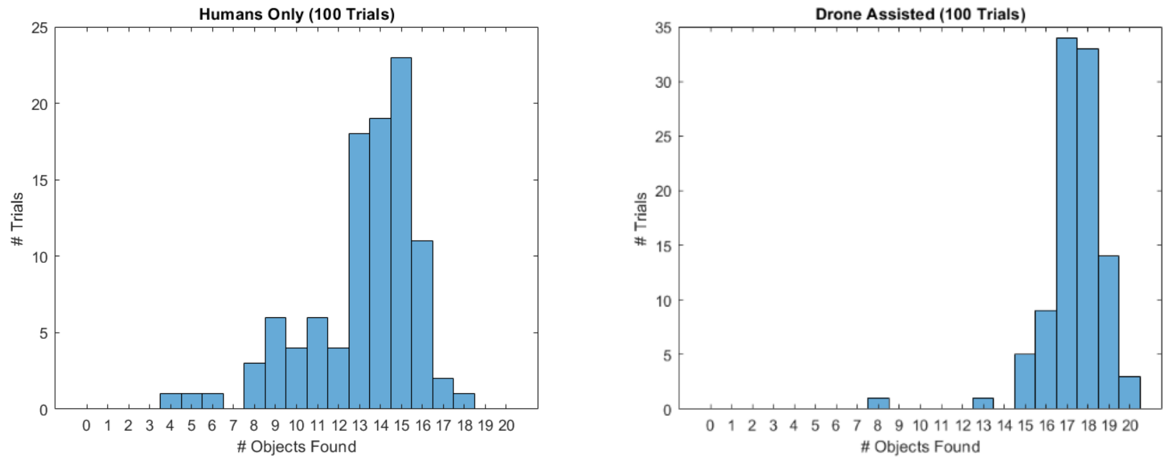

Figure 7: Histogram of Objects Found with Data Set 1

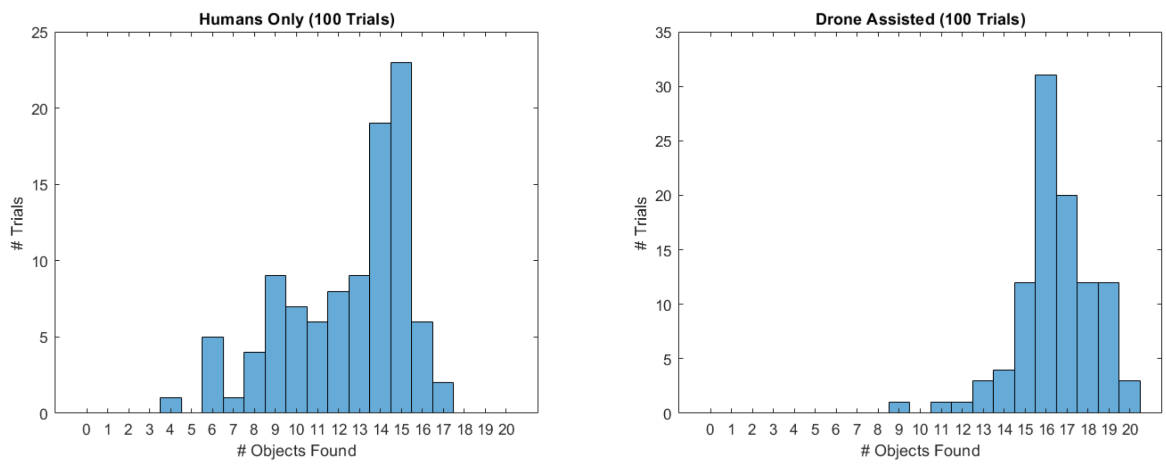

Figure 8: Histogram of Objects Found with Data Set 2

Objects Collected:

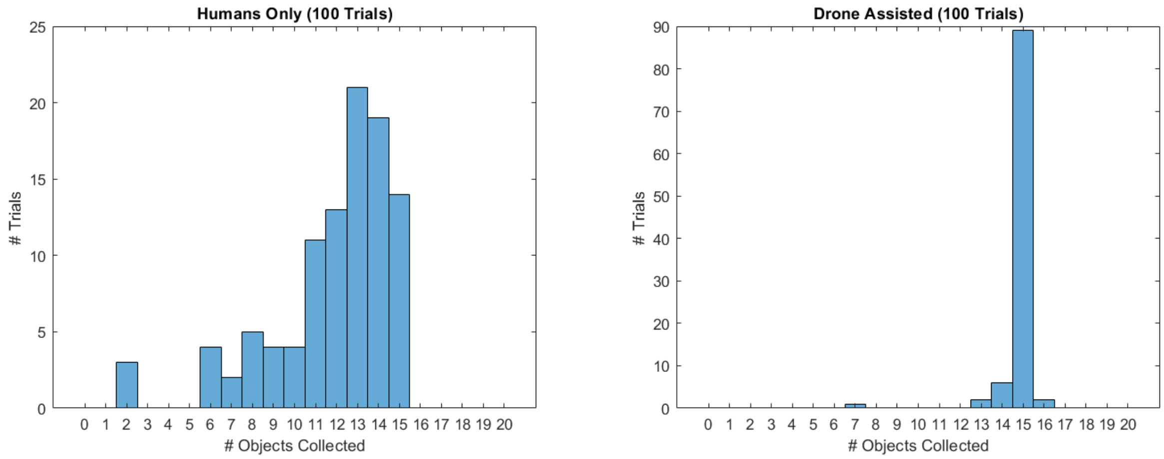

Figure 9: Histogram of Objects Collected with Data Set 1

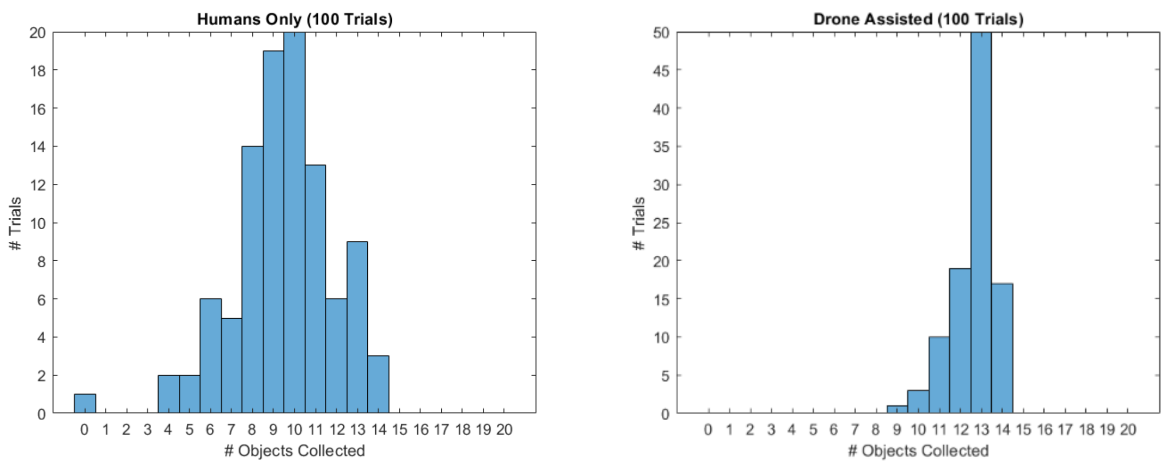

Figure 10: Histogram of Objects Collected with Data Set 2

Table 2: Means of All Data

	Humans Only	**Drone Assisted**
Mean Objects Found – Set 1	13.2	17.4
Mean Objects Found – Set 2	12.4	16.5
Mean Objects Collected – Set 1	11.9	14.8
Mean Objects Collected – Set 2	9.5	12.7

Table 3: P-Score of Humans-Only vs Drone-Assisted

	P-Score – Objects Found	P-Score – Objects Collected
Set 1	4.28e-24	6.66e-16
Set 2	1.49e-19	1.26e-21
Combined Sets	6.99e-42	1.75e-35

Histograms of the results of all trials of the Monte-Carlo simulation are included in Figures 7-10. The mean of each of these data sets was calculated and is listed in Table 2. T-Test's were performed comparing the results of the humans-only strategy to the drone-assisted strategy for each data set and the resulting p-scores are listed in Table 3. All p-scores are significantly less than 0.05 meaning the null hypothesis is rejected and there is a statistical difference in performance between the two strategies.

Discussion:

Based on the p-scores in Table 3, there is a clear advantage to the drone-assisted strategy over the basic humans-only strategy. This is unsurprising as, not only is there an additional agent being utilized in this strategy, the agents are also more spread out in both space and time in this strategy. One of the clear takeaways of the real-life exercise was to spread agents out. If agents are looking at the same thing, one of them is wasting time and, since endurance is very limited in this exercise, wasted time will hurt the team's performance.

Overall, the results of the simulation seem reasonable. In the real-life exercise, between 8 and 15 objects were collected each trial in what was ultimately a hybrid of the two strategies tested. The mean number of objects collected for all configurations simulated ranged from 9.5 to 14.8 objects. This is within the range of real-life results, so it appears that the simulation is a reasonably accurate model of the real scenario.

One interesting result is that for all metrics, both strategies performed worse on data set 2 than they did on data set 1. If you look at the location of objects in Figure 5 and Figure 6, this makes sense. Set 2 has more objects on the left of the search domain. Not only is this area further from the entrance, it also is the first area consumed by fire. Additionally, the far left of the search domain is inaccessible in the humans-only strategy because humans cannot cross the compromised floor without a drone inspecting it. This likely explains the discrepancy in performance between the two data sets, however, testing more data sets is required to get a more accurate measure of the generalized performance of each strategy.

Summary:

This simulation modelled a real-life search and rescue exercise and was able to test different resource distribution strategies to be used in the exercise. As demonstrated by very low p-scores, the developed drone-assisted strategy outperformed the baseline humans-only strategy. This was reflected in the real-life exercise where the use of the drone, as well as the separation of agents, proved to be important factors in increasing the number of objects rescued. Overall, the simulation appeared to be a relatively accurate model of the real exercise. Being able to accurately model real objects and behaviors is crucial to engineering because it enables systems to be designed without the need for large amounts of expensive, time-consuming, or impossible testing. While this simulation was only a model of a simple and harmless exercise, much of the same tools and techniques can be applied to more realistic search and rescue scenarios that are impossible to perfectly predict and test. Thus, with further refinement, simulations like this could have a real, life-saving impact in emergency situations.

References:

1. https://thenextweb.com/plugged/2017/12/12/review-my-whole-family-had-fun-playing-with-this-little-drone/

Appendix:

See attached video for fully animated simulation.