**TITLE: Multi-Drone Coordination for Search and Rescue Missions**

1. **Project Overview**

In disaster-stricken areas, search and rescue missions are critical to saving lives. However, these missions can be dangerous and time-sensitive, and they require efficient coordination and resource allocation. The aim of this project is to design a multiagent system using drones that can detect the survivors and deliver food and medication to them in a timely and efficient manner. The drones will have to take food and medications from the distribution center and then deliver them to the survivor as soon as possible. The drones will interact with each other to avoid collision. The distribution center can be in different places, or it can be many in numbers. The drone’s performance for one distribution center in different places will be checked. Also, the drone’s performance for multiple distribution centers in different centers will be checked.

* 1. **Report Modification**

24/02/2023 Initial version

03/03/2023 Completed section 1,2.

20/03/2023 Modified section 1,2 and Completed section 3.

27/03/2023 Modified section 1,2,3 and completed section 4.

* 1. **Context**

The cyber-physical system for this problem will involve integrating the drone with a control system that allows for real-time monitoring and control of the drone's operations. The drones will take decisions about detecting the survivors, providing the food and medication to the survivors, taking the food and medication from the distribution center, checking the other drones so that they won’t collide with each other and reaching the survivor as soon as possible by searching for the area with most survivors.

The solution for this problem must consider the constraints of drone capabilities for one distribution center for food and medications in different random places or for multiple distribution center in different random places, as well as the hypothesis that the drone can detect survivors, detecting distribution center, receiving food and medication from distribution center and delivering food and medication to the survivors, detecting drones not to collide. Additionally, the solution must consider the potential risks and challenges associated with operating drones in a search and rescue mission, such as detection issues.

A multiagent solution is adequate to solve this problem because it involves multiple drones that need to coordinate and collaborate with each other to efficiently search and rescue survivors, deliver food and medication, and cover a large area. Since there are various tasks and actions involved, the decentralization of the decision-making process allows for more efficient and effective decision-making by the drones in their respective areas.

The quality of service for this problem is timely delivery of food and medication to the survivors and the quality of business for this problem is cost-effectiveness in terms of the resources used by the drones to carry out the search and rescue mission.

* 1. **Multiagent solution overview**

**Agents**

**Drones:** The agent for this problem is the drone, which is responsible for detecting and locating survivors, picking up food and medication from the distribution center and delivering the food and medication to the survivors' location as soon as possible to save more lives. To avoid collisions the drone will interact with each other. The drone will check for the location with less or no drones in a grid location with survivors.

**Environment**

For this drone problem the environment will contain the distribution center from where the drones will get food and medication supplies for the survivors, will contain multiple drones which will pick up the food and medication for the survivors from distribution center, fly to detect the survivors and supply them the food and medications, will contain multiple survivors who will wait for the drones to supply food and medication. Also, there will be survivors to whom the drones supplied the food and medications. We will consider them as rescued survivors. There may be many constraints, but we will only consider the drone, distribution center and the survivors to success the mission.

**Interaction**

The interaction in this problem involves the drone as the agent interacting with the environment. The drone will perform actions such as picking up food and medication from distribution center, flying, detecting survivors, delivering food and medication to the survivors, detecting drones and going to the location with less drones and most survivors to avoid collision. It will detect the location of survivors to supply food and medication or distribution center to pick up food and medication for survivors and their surroundings and use that information to make decisions about where to fly and what actions to take. Like if the drone is full of food and medication it will search for survivors to supply food and medication. After supplying the food and medication the drone will be empty and it will need to be full again to save other survivors. Also, if a drone locates survivor and there is an existing drone in the same location, the new drone will consider whether the located drone is full or not. If the located drone is empty, then the drone needs to go to that location. Also, one situation can come that there are several survivors and the located drone is not full enough to supply food and medication to the survivor. That time the drone needed to go to that location. That’s how the decision process for drones will work. When a drone becomes empty, it will go to the distribution center and then receive food and medication for the survivors and becomes full again and again go for the mission to save the survivors.

**Organizations**

* + The drone will go to the distribution center right after becoming empty and after becoming full of food and medication the drone will go to search for survivors.
  + The drone should detect the survivors and their location accurately.
  + The drone should be able to deliver food and medication to the survivors.
  + The drone should optimize the search and rescue mission. As the drones will search for the location with most survivors, there is more possibility of having survivor in that location. The sooner the drone will be able to detect survivors, the faster it will be able to supply the food and medication to the survivor.
  + To avoid collision the drones will be able to detect each other.

**Goals**

The main goal of this drone delivery system is to deliver the food and medication as soon as possible. The system should be able to effectively locate and provide aid to survivors and ensure efficient allocation of resources among the drones. We’ll check the system works good for the one distribution center of food and medication in which random space. Also, we’ll check the performance for multiple distribution center of food and medication.

**Motivation**

The motivation of this problem is to design a multiagent system using drones for search and rescue missions, where the drones can avoid collision with each other, can detect survivors and can deliver food and medication in emergency situations. The system aims to enhance the effectiveness and efficiency of search and rescue operations, which can ultimately lead to saving more lives.

1. **Conceptual Model**
   1. **Agent**

**Perception:** The drone perceives the environment through its sensors. It can detect the survivors grid location, the distribution centers grid location, can move to the survivor or distribution center, can supply food and medication to the survivor and can pick up food and medication from the distribution center. Drones are able to perceive other Drones, otherwise they may collide.

**Knowledge:** The drone has knowledge of the map of the environment and the location of the distribution center, location of the survivors it has detected, has knowledge whether the drone is full of food and medications or not. If a survivor already receives food and medication, it will be considered as rescued survivor so that the drone will not give the food and medication to the same survivor again and again. As the survivor needs food and medications and the survivor will run towards drone so it will be easier for the drone to detect the survivor. The drone will not collide with other drones as the drone is aware of other drones.

**Decision Process:** The objective function of the drone is to minimize the flying time and give the food and medication as soon as possible to save more lives. The drone's decision-making process aims to maximize the number of survivors it can deliver food and medication to while minimizing fuel consumption. The drone will detect the survivor and will deliver the food and medications. The drone decides to fly to the survivor if it is full of food and medication. If the drone is full it will fly to the survivor and supply the food and medication and if the drone is empty the drone will fly to the distribution center. If the drone detects a drone near it, it will try not collide with other drone.

**Action Model:** Based on its decision, the drone acts and navigates to the location of the survivor, drops the food and medication, and goes to distribution center to refill the food and medication.

Diagram

Description automatically generated

Fig 1: Action model flow diagram

* 1. **Environment**

**Survivors:** Represented as entities in need of food and medication, located in different areas of the environment.

**Rescued Survivors:** Represented as entities of survivors who has got food and medication supply from drones, located in different areas of the environment.

**Distribution Center:** Represented as entities from where the drone will take the food and medications.

The environment is dynamic, and the locations of survivors and resources may change over time. The drone must be able to perceive its environment, including the locations of survivors and resources, and make decisions based on this information to accomplish its goal of delivering aid and assistance to survivors.

* 1. **Interaction**

The interaction is decentralized and involves multiple drones, survivors, centralized or decentralized distribution center. Each drone is responsible for detecting survivors within its designated area and supplying them with food and medications.

When a survivor is detected, the drone will provide assistance by delivering necessary supplies such as food and medication. The drone will fly to the location of the survivor and drop off the supplies. After dropping off the supplies the drone will be empty, so it will go to the distribution center to get the food and medication and again the drone will search for another survivor to help with food and medication. In this way the drone will be able to save the survivors as much as possible.

Overall, the interaction in this simple drone problem involves a collaborative effort among multiple drones to efficiently locate survivors and provide necessary assistance.

* 1. **Organization**

The drone agent will be responsible for detecting survivors and delivering food and medication. The drone will detect the survivor and interact with them by delivering food and medication. The survivor who got the food and medication supply will be rescued survivor and after delivering the food and medication the drone will be empty. So, the drone will take the decision to fly to the distribution center and refill the food. If a drone detects another drone, it will try not to collide. Overall, the organization of this problem will be a decentralized system where each drone agent will operate independently and make decisions based on the activities.

1. **Repast Implementation**
   1. **From the conceptual model to the simulation model**

To build the simulation model from the conceptual model we need to be more precise and logical to design the model. As we don’t know how many survivors or drones we have, we will have a random generator which will generate some survivors and drones and will assign them to different grid points. As the drone needs to save as many survivors as it can so it will have a method to detect most of the survivors and give them food and medications. If a drone detects another drone it will try not to collide with each other. We’ll check the drone performance to save the survivors when the distribution center is centralized and again check the drones performance when the distribution center is decentralized. For the initial stage we will check with one centralized distribution center, and it can be anywhere in the grid. Next, we’ll check with decentralized distribution center which can be of any number and anywhere in the grid. About the survivors, whenever a drone saves a survivor, the survivor will turn into rescued survivors so that the drone will not supply food and medications to the same survivor again and again. The survivor will want to get the food and medications as soon as possible so the survivor will run towards the location with drones. When the survivor runs, they will get tired and take rest to increase their energy and then they will run again towards drone. If the survivor becomes rescued survivor means the survivor got food and medications from the drone it will not move and also the drone will be concern only for survivors and the distribution center.

* 1. **Indicators’ implementation**

To implement the agent, environmental we’ll have four classes drone, distribution center, survivor and rescued survivor. There will be a class DroneBuilder for the context. In the repast environment we’ll have our package “drones” and in src directory we’ll have all the classes which are to be implemented.

**Drone**: The drone class will contain the main agent drone. In the constructor it will have the space and grid location. The step() is the method which helps the drone to take decision whether it need to move towards survivor or towards distribution center. There is a Boolean value full, if this value is “true” that means the drone contains food and medications and it need to go towards survivor on the other hand if the full value is “false” that means the drone is empty and it needs to go towards distribution center. Also, the drone tries to save as many survivor as it can, so it tries to choose the location with most survivors. The moveTowards(), moveTowardsDistributionCenter() method moves the drone until it reaches its destination survivor or distribution center. When the drone reaches the distribution center the “full” variable becomes “true” and when the drone reaches the survivor rescue() is called. There the drone becomes empty after supplying the food and medication. So, the “full” variable becomes “false” and as the survivor becomes rescues survivor, one survivor is reduced from survivor and a rescued survivor is added instead of the survivor. Every tik is counted above the step() method.

**Survivor**: The survivor has space and grid same as drone. It has energy which is used to check whether the survivor will run towards drone or will take rest. The run() method is implemented to take the drones location and moveTowards() is called to run towards drone if the survivors energy is greater than zero. The watcher class above the run() method checks “moved” which indicates the drone moved and that’s how the survivor gets to know about drones existence. When the survivor runs, the survivors energy looses which in implemented in moveTowards().

**Rescued Survivor**: When the survivor is allocated with food and medication by the drone the survivor is turned into rescued survivor. The RescuedSurvivor class contains gird and space constructor only. It doesn’t contain any other method.

**Distribution Center**: When the drone becomes empty it goes to the distribution center to get full. The distribution center has grid and location. So, the constructor contains grid and locations. The distribution center can be one or many. One for centralized system and many for decentralized system and can be in any point of the grid location.

**DroneBuilder**: The drone builder is the context builder. The continuous space, grid location is implemented here. All the four classes object are implemented here using loops. Using the params the initial value will be given for all the classes. Object moving is implemented in this class. This class controls all the other class which contains agent as well as environment.

In the context.xml two projection for continuous space and grid are implemented and by the run button the implemented codes are run and checked.

1. **Evaluation**
   1. **Scenario**

Here,

 means Survivors,

 means rescued survivors,

 means drones,

 means distribution center.

In the scenario of Fig 2 the distribution center is in the right below side of the grid. There are 200 survivors, 2 rescued survivors and 5 drones. In the scenario of Fig 3 the distribution center is in the right above side of the grid. There are 200 survivors, 2 rescued survivors and 5 drones. In Fig 4, the distribution center in middle grid (27, 29). There are 200 survivors, 2 rescued survivors and 5 drones. In Fig 5, the distribution center is in left below grid(2,1). In the scenario Fig 6, there are multiple distribution center (5), there are 200 survivors, 2 rescued survivors and 5 drones. The drones are in purple color square box, survivor are seen as human model, green square means rescued survivor and the distribution center is in blur dots.

Chart, scatter chart

Description automatically generated

Fig 2: DistributionCenter Right below (blue dot)

Chart, scatter chart

Description automatically generated

Fig 3: DistributionCenter Right above (blue dot)

Chart, scatter chart

Description automatically generated

Fig 4: distribution center in middle grid (27, 29)

Chart, scatter chart

Description automatically generated

Fig 5: left below grid(2,1)

Chart, scatter chart

Description automatically generated

Fig 6: Multiple distribution center (5)

* 1. **Results**

For the first scenario (Fig 1) we ran our simulation for 100 ticks and got result where the drone could rescue total 32 survivors and the updated survivor number was 170 (Fig 7) graph in (Fig 9). As the activities only affect the survivor number and rescued survivor number, so these two values were changed rest were same as initial.

Chart, scatter chart

Description automatically generated

Fig 7: Distribution Center Right below result after 100 ticks survivor 170 and rescued survivor 32

For the second scenario (Fig 2) we ran our simulation for 100 ticks and got result where the drone could rescue total 26 survivors and the updated survivor number was 176 (Fig 8) graph in (Fig 10). As the activities only affect the survivor number and rescued survivor number, so these two values were changed rest were same as initial drone number 5 and 1 distribution center.

Chart, scatter chart

Description automatically generated

Fig 8: Distribution center right above result after 100 ticks survivor 176 and rescued survivor 26

Chart, line chart

Description automatically generated

Fig 9: Performed on 100 ticks for right below result after 100 ticks survivor 170 and rescued survivor 32

Chart, line chart

Description automatically generated

Fig 10: Performed on 100 ticks for right above result after 100 ticks survivor 176 and rescued survivor 26

Chart, scatter chart

Description automatically generated

Fig 11: Performed on 100 ticks for distribution center grid (27,29). The survivor count 156 and rescued survivor 46 result scenario

Chart, line chart

Description automatically generated

Fig 12: Performed on 100 ticks for distribution center grid (27,29). The survivor count 156 and rescued survivor 46 graph representation

For the third scenario (Fig 4) distribution center in middle grid (27, 29) we ran our simulation for 100 ticks and got result where the drone could rescue total 46 survivors and the updated survivor number was 156 (Fig 11) graph in (Fig 12). As the activities only affect the survivor number and rescued survivor number, so these two values were changed rest were same as initial drone number 5 and 1 distribution center.

For the fourth scenario (Fig 5) distribution center left below grid(2,1) we ran our simulation for 100 ticks and got result where the drone could rescue total 27 survivors and the updated survivor number was 175 (Fig 13) graph in (Fig 14). As the activities only affect the survivor number and rescued survivor number, so these two values were changed rest were same as initial drone number 5 and 1 distribution center.

Chart, scatter chart

Description automatically generated

Fig 13: left below grid(2,1) survivor count 175 and rescued survivor count 27

Chart, line chart

Description automatically generated

Fig 14: left below grid(2,1) survivor count 175 and rescued survivor count 27

For the fifth scenario (Fig 6) we ran our simulation for 100 ticks and got result where the drone could rescue total 37 survivors and the updated survivor number was 165. The updated scenario is in (Fig 16) and graph is in (Fig 15). As the activities only affect the survivor number and rescued survivor number, so these two values were changed rest were same as initial.

Chart, line chart

Description automatically generated

Fig 15: Initial 200 survivor, 2 rescued survivors, 5 drones, 5 distribution centers After 100 ticks survivor count 165 and rescued 37

Chart, scatter chart

Description automatically generated

Fig 16: Initial 200 survivor, 2 rescued survivors, 5 drones, 5 distribution centers After 100 ticks survivor count 165 and rescued 37

* 1. **Discussion**

From the result section we can get the idea of distribution center in different location. The drone performance for distribution center in right corner and left corner average survivor remains 173 which means in average the rescued survivor for distribution center in corners is 27 for 200 survivors. If we check the case for the distribution center in center of the grid. The survivor count is 156. So within 200 survivors (200-156)=44 survivors are rescued within 100 ticks.So, for one distribution center in different location, when the distribution is near to center the decentralized multiagent solution for drones gives better performance.

TABLE 1. Result View

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Drone | Distribution Center | | Survivor | Rescued Survivor | Drone after run | Distribution Center  After  run | | Survivor after run | Rescued Survivor after run |
| Number | Position | Number | Position |
| 5 | 1 | Right below | 200 | 2 | 5 | 1 | Right below | 170 | 32 |
| 5 | 1 | Right above | 200 | 2 | 5 | 1 | Right above | 176 | 26 |
| 5 | 1 | center | 200 | 2 | 5 | 1 | center | 156 | 46 |
| 5 | 1 | Left | 200 | 2 | 5 | 1 | left | 175 | 27 |
| 5 | 5 | random | 200 | 2 | 5 | 5 | random | 165 | 37 |

We can notice an amazing fact that when we checked for 100 ticks for 5 distribution centers for 200 survivors the drone could supply food and medication to165 survivors which is greater than the survivor count for one distribution center in center (156). When the distribution center is in the center of grid location the drone can supply food and medications to more survivors than multiple distribution centers.

The run was multiple times for different values. Some average scenarios are shown in the scenario and result section. We can get the output data in a csv file or in txt file. So, finally we can conclude that, for the decentralized drone problem to save survivors, the drone performs well in the distribution center in the center of grid location than the distribution center in corner of the grid location or multiple distribution center in random places.