Toward Robotic Triage: a Distributed Task and Motion Planning Framework for Efficient Human-Robot Emergency Response

Grace Weaver[†], Abby Manalang[†], Patrick Sherman[§], Henning S. Mortveit[†], and Nicola Bezzo[†§]
[†]Department of Systems and Information Engineering
[§]Department of Electrical and Computer Engineering
University of Virginia, Charlottesville, Virginia 22903
Email: {afe8jr, dds2tr, ukw4tc, hsm2v, nb6be}@virginia.edu

***Abstract*—This paper proposes a novel human-robot coordination framework to enhance medical assistance during urgent disaster relief operations. Specifically, we propose to leverage a heterogeneous robotic system in which multiple UGVs and a UAV seamlessly coordinate with a medic to find and assist victims. The UGVs are tasked with exploring the environment, finding victims, performing basic triage operations, and reporting to a UAV. The UAV operates as a relay to quickly inform the medic about the location and status of victims, while providing an optimized route to follow. A task and motion planning (TAMP) approach is proposed to coordinate UGVs and the UAV with the medic. The problem is then cast as a Traveling Salesman Problem (TSP) incorporating medical policies from domain experts to define the best route for the medic to follow considering victims' locations and their severity. Extensive simulations and experiments are performed to showcase the effectiveness of the proposed framework compared to different strategies in terms of task allocation and planning. The results of the proposed human-robot triage approach shows that by strategically coordinating heterogeneous robotic systems it is possible to reduce the amount of time it takes to locate and assist casualties by an expert medic.**

***Note*—Simulations and experiment videos can be found in https://www.bezzorobotics.com/sieds25**

***Index Terms*—human-robot interactions, task and motion planning, autonomous mobile robots**

I. Introduction

In instances of disaster, medics are limited in time and resources to locate victims and perform life-saving intervention. During such safety critical operations, decreasing time and manpower are key factors to improve assistance and survivability of victims. In current practice [1], decisions regarding continued exploration vs. attending to victims fall to the medics who are responsible for weighing the costs between continuing to search for victims with more critical injuries and stopping to triage and aid known victims. In addition, medics must account for the limited survival time of victims depending on the severity of injuries. To solve this problem, in this paper we propose to leverage mobile robotic systems to assist a human medic with relief operations. Our proposed framework is inspired by hospital emergency room common practices in which a patient is first triaged by nurses and then admitted and visited by expert medics at times that depend largely on the gravity of the condition and the availability of the medic. Similarly, in our approach, we adopt a hierarchical

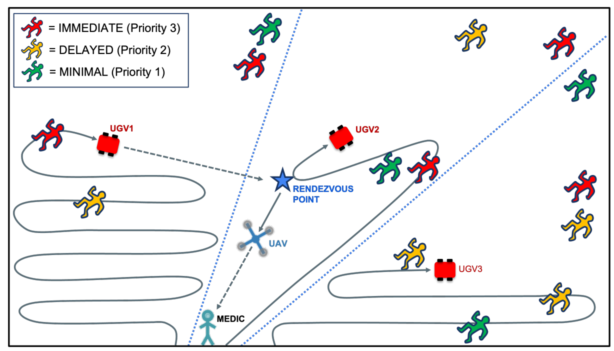


Fig. 1: Pictorial representation of the problem and proposed system and solution.

divide-and-conquer approach in which robots are in charge of exploring an environment in search of victims and performing basic triage. Victims are scored on the basis of severity before they are reported to a medic who will tackle more advanced medical aid, which would be hard to perform by a robot. By implementing such human-robot triage framework, medics can reduce the exploration and triage time of disaster areas with limited manpower.

Fig.1 depicts the proposed problem and solution. Our proposed solution leverages a heterogeneous robotics system with limited-range communication capabilities in which: i) a combination of UGVs are tasked with environment exploration, locating victims, performing basic triage, and reporting to ii) a UAV that acts as a relay to quickly deliver tasks to iii) a medic that will execute the received tasks.

We propose a task and motion planning (TAMP) approach for robust, fast, and seamless coordination between UGVs, UAV, and medic by designing a state machine-based algorithm that coordinate motion and communication between UGV and UAV and between UAV and medic, with the UAV acting as a mobile hub between the UGV and the medic. Our solution is scalable and decentralized, making it computationally efficient: in fact, each UGV acts on a portion of the environment independently from the other UGVs and does not need to know the position of the medic, only the rendezvous position of the UAV, which in this work is assumed fixed.

When victims states are delivered to the medic, the medic, which is equipped with a computer or smart device to receive and elaborate data from the UAV, considers victim assignment as a Traveling Salesman Problem (TSP) to determine the best route between victims based on severity of injuries and distance from the medic. The solution of such TSP determines a time- and energy-efficient route that targets high-priority victims first, followed by lower-priority ones. For example, in Fig.1, UGV2 finds a high-priority victim (red), and is triggered to stop exploring and move to the rendezvous point (star in the figure) where a UAV is waiting. Upon receiving the task, the UAV flies to the medic to report the task while the UGV2 returns to explore the remaining section of its assigned environment. Similarly, UGV1 finds a high-priority task and switches to move toward the rendezvous location and wait until the UAV is available before returning to explore the rest of its area. The effectiveness of our approach is validated in simulations and experiments and compared with three different strategies in terms of task allocation, planning, and communication assumptions.

Our contribution is twofold: 1) we propose an innovative, general, and scalable robotic triage framework for heterogeneous robotic systems to guide a medic toward an optimal prioritized sequence of tasks, and 2) we provide simulations and experiment results comparing our method with different techniques to validate the proposed approach.

The rest of the paper is organized as follows: in Section II we provide an overview of the triage process and patient classification based on different priority levels. In Section III we provide an overview of the state of the art in human-robot triage systems. In Section IV we provide an overview of our methods. In Section V we describe the proposed solution and the behavior of different agents in the system. Section VI addresses the simulation study of our solution and results. Lastly, Section VII summarizes our research and discusses future work.

II. Background

Although there are multiple different methods of triage, we chose to focus on the SALT mass casualty triage, described in [2], due to the simple and efficient method for determining how urgent a victim needs life-saving intervention. The SALT mass casualty triage method is sorted into four groups: Sort, Assess, Life-saving interventions, and Treatment. During the sort phase, the patient is sorted into one of three categories: walk, wave/purposeful movement, and still/obvious life threat. For the purpose of our paper, when we refer to a robot performing triage, it is referring to the sort phase of the SALT method. These phases are sorted into numerical categories, with a *priority 1* representing a victim who is able to walk and labeled in the simulation with a green star, a *priority 2* representing a victim capable of purposeful movement but unable to walk and represented by a yellow star, and a *priority 3* representing an unconscious victim and represented by a red star.

III. Related Work

The use of robotic systems for disaster response traditionally focuses on search and rescue. [3] reviews robotic operations for disaster response used in practice, addressing the shortcomings with real-world examples. These real-world scenarios use teleoperated semi-autonomous robotic systems; current major causes of failure include human operator error, lack of robustness, and imperfect autonomy. We aim to address these sources of failure through our autonomous exploration and triage algorithms, limiting human interaction with the system and increasing robot decision-making capabilities. While [3] provides context to the real world application of robotic systems for disaster response, [4] introduces an AI-driven Robotic Triage labeling and Emergency Medical Information System (ARTEMIS) to test robotic triage and exploration. Our project utilizes similar triage classification, but expands on this method by incorporating multiple robots exploring a singular environment. Additionally, our method improves efficiency through the inclusion of a UAV acting as a communication relay with the medic. [5] uses a similar strategy with an agricultural robotic system that utilizes a UAV as a relay between a human controller and multiple-mobile robots. This work highlights how the dynamic nature of the UAV provides an advantage due to the ability for direct line of sight communication which it makes it useful in disaster-relief operations. Both [6] and [7] discuss methods used to aid the triage process. [6] focuses on TSP solutions for human-robot industrial systems inspection, which we also use to determine a prioritize path for the medic. Additionally, the task list is updated as more victims are located; because of this, the TSP updates with each victim added to the medic's task list. [7] provides an exploration pattern for the UGVs to divide and explore the environment. Our work uses such method to define exploration paths for the UGVs, but includes our own behaviors for task discovery. While many sources touch on aspects related to the proposed human-robot triage, none combine all the complex aspects into a generalizable multi-robot system solution.

IV. Problem Statement

The goal of this work is to *maximize the amount of high-priority patients that a medic treats while minimizing the amount of time it takes to treat all of the victims, and the energy exerted or equivalently the distance traveled.*

In our proposed research, we consider a heterogeneous robotic system with a combination of UGVs, UAVs, and a medic operating in a limited communication environment. We assume that throughout the environment there is an unknown number of victims at unknown locations, each with a given priority based on the extent of their injuries. For ease, we assume that the UGVs are able to perform triage and recognize the level of priority for each victim when in their field of view. We also assume that the UAV is able to predict the medic's location at all times due to a priority knowledge about their dynamics (e.g., speed) and their tasks, which the medic shares with the UAV when they meet.

V. Approach

Our proposed approach for finding the most efficient and effective triage strategy is divided into three parts as depicted in Fig.2. Each agent class – UGV, UAV, or medic – has associated a different role and behavior. We assume that each agent has limited communication range hence allowing communication only when in close proximity. A UGV is responsible for locating victims and performing basic triage, i.e., labeling each victim based on the priority levels outlined in the background section, and adding the detected victims to a task list. The UGV then reports this task list to a UAV strategically positioned in the middle of the environment. Lastly, the UAV flies to the medic to report the task list; after receiving the list, the medic computes the optimal order to visit the tasks.

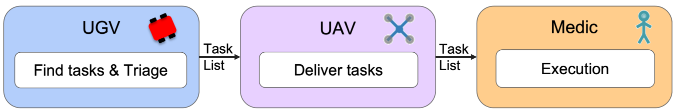


Fig. 2: Proposed heterogeneous system architecture diagram

A. Ground Vehicle-Explorer

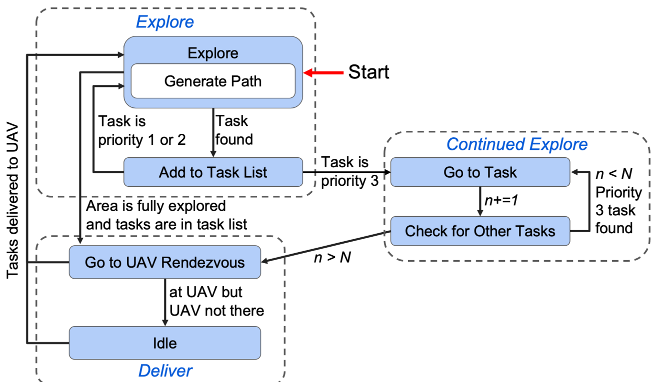


Fig. 3: UGV State Diagram

In our proposed solution, UGVs are tasked with *exploring* the environment to search for tasks (i.e., victims) following the state diagram in Fig. 3. During exploration, each robot's path is set in such a way to ensure coverage of the entire environment by leveraging the *Divide Areas Algorithm for Optimal Multi-Robot Coverage Path Planning* (DARP) [7]. The algorithm splits the environment evenly among the ground robots and assigns each robot a path to follow in order to survey the entire environment effectively. In our implementation, the ground robots follow their paths until a task appears in their sensing range. If the task is a lower priority (level 1 or 2), the robot adds the tasks to its task list and continues to follow its exploration path. If the task is of high-priority (level 3), then the agent moves into the *continued explore* state. While in this state, the UGV continues to explore the area in which the high-priority task was discovered by driving to the middle of the task and surveying the surrounding area for more tasks. The robot do this up to N times, where N is decided by the user, and then report all found tasks to the UAV. The UGV is programmed with such behavior in order to discover clusters of tasks; this will avoid additional trips to the UAV if multiple tasks are discovered in the same area. The UGV leaves the continued exploration state if any of the following conditions are met: 1) there are no additional high-priority tasks found, 2) there are fewer than N tasks found, and upon reaching the center of the last detected high-priority task, there are no additional tasks in the surrounding area, or 3) N tasks have been recovered. When one of these scenarios occurs, the UGV moves into the *deliver* state. In such state, the UGV moves to the UAV rendezvous point. If the UGV does not find the UAV at the rendezvous point, it waits until the UAV returns. If the UGV finds any lower-priority tasks while exploring, it waits to report them and continue exploring until a high-priority task is discovered or the assigned section is fully covered. In some instances, the UGV does not find any priority 3 tasks. In this case, the UGV reports all tasks to the UAV after finishing exploration.

B. UAV Relay

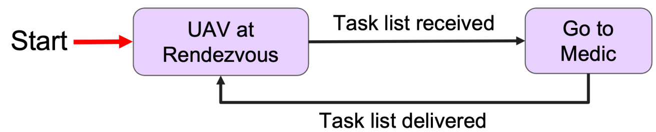


Fig. 4: UAV State Diagram

The UAV in this work acts as a relay and is responsible for delivering tasks received from the UGVs to the medic as depicetd in Fig. 4; it is the only agent that interacts directly with the medic. The UAV knows the medic's dynamics (e.g., average and max walking speed) and the order list of tasks that the medic would be working on which is shared once the UAV and medic meet. The UAV's behavior can be divided into two states as depicted in Fig.4: *rendezvous* and *go to the medic* to deliver the tasks. When the UAV is at the rendezvous state, it positions itself in the middle of the environment. This provides a shorter distance for all UGVs to report victims to the UAV, allowing the UGVs to quickly return to exploring the environment. We chose this approach for ease of discussion and implementation, however other methods can be considered to further minimize the distance traveled by the UGVs such that: 1) making the UAV's rendezvous point dynamic with the middle point estimated at every point in time based on current predicted locations of UGVs, or 2) making the UAV constantly or periodically visit each UGV with a round-robin-like approach. In our approach, to prevent the UAV from excessive energy use, the rendezvous point remains in the middle of the environment throughout the exploration process. During the deliver state, the UAV reports all tasks to the medic. Although the UGV only reports to the UAV after discovering a high-priority task, the UAV reports all tasks that the UGVs have discovered organized in a task list, regardless of the priority.

C. Medic

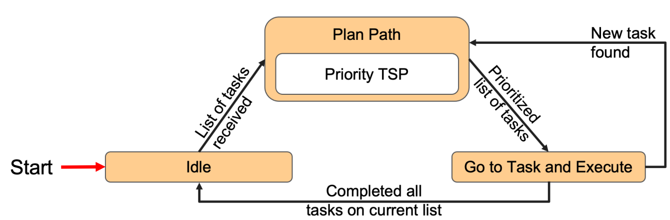


Fig. 5: Medic State Diagram

The medic is responsible for treating the victims reported by the UAV. The medic may decide to go to tasks based on the order they were received, closest distance, or any other metric. Since we are concerned with treating higher-priority patients first, we cast this problem as a Traveling Salesman Problem (TSP). The complete medic behavior is described by the state machine shown in Fig.5. While the medic waits for tasks, they remain in an idle state. When the medic receives the task list from the UAV, it moves into the *plan path* state. In this state, the medic determines the order to tend to victims through a priority TSP solution. The priority TSP first splits the task list $\mathcal{T}$ into three separate lists based on priority. This algorithm utilizes a brute-force TSP, so that each route is considered, and the route with the shortest path is selected by the medic. Since the TSP is run on each task list separately, the optimized path is generated for each priority task list, producing the shortest path among the known higher-priority tasks, followed by the shortest path for the lower-priority tasks. These lists are merged into a singular optimized task queue for the medic, stored in $\mathcal{T}_{ordered}$. The prioritization of the queue through the TSP is shown in Alg. 1. If there are no tasks discovered of a

Algorithm 1 Priority TSP

Input: Task list $\mathcal{T}$ with priorities $\{1, 2, 3\}$
Output: Ordered task queue $\mathcal{T}_{ordered}$
Initialize lists $\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3 \leftarrow \emptyset$
for each task τ in $\mathcal{T}$ **do**
  if priority$(\tau) = 1$ **then**
    $\mathcal{T}_1 \leftarrow \mathcal{T}_1 \cup \{\tau\}$
  else if priority$(\tau) = 2$ **then**
    $\mathcal{T}_2 \leftarrow \mathcal{T}_2 \cup \{\tau\}$
  else if priority$(\tau) = 3$ **then**
    $\mathcal{T}_3 \leftarrow \mathcal{T}_3 \cup \{\tau\}$
  end if
end for
$P_3 \leftarrow$ TSP$(\mathcal{T}_3)$ ▷ Optimized path for priority 3
$P_2 \leftarrow$ TSP$(\mathcal{T}_2)$ ▷ Optimized path for priority 2
$P_1 \leftarrow$ TSP$(\mathcal{T}_1)$ ▷ Optimized path for priority 1
$\mathcal{T}_{ordered} = \{P_3, P_2, P_1\}$
return $\mathcal{T}_{ordered}$

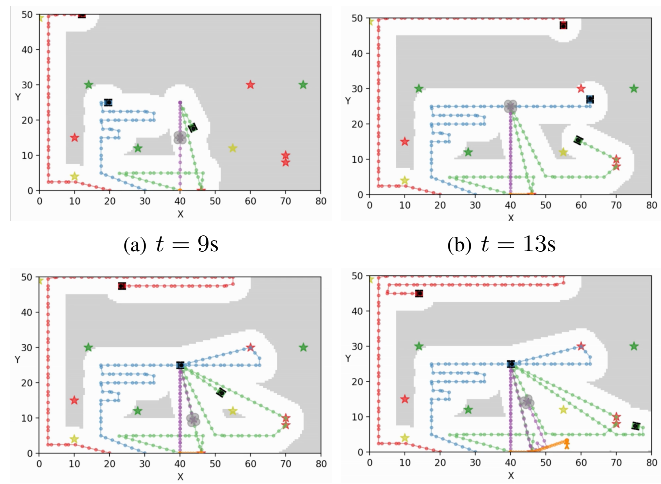
certain priority, for example, if there are no priority 2 tasks, the finalized task list would consist of the priority 3 tasks sorted using the TSP followed by the priority 1 tasks sorted by the TSP. As long as there are tasks in the queue, the medic will continue treating victims. Once a task is completed, it is removed from the queue and added to a completed task list to prevent the medic from going to the same victim twice. When all tasks are completed from the task queue, the medic remains idle at the last task completed. Whenever a new task is added, either from the relay delivering a new task or the medic discovering a task, the TSP is run and a new path is generated. The medic can also add more victims to the task queue if they are discovered while traveling. If a medic finds a new victim, however, the priority TSP must be run again; just because a medic discovers a victim does not mean they will be treated right away.

VI. Simulation and Experiment Results

A. Simulations

Several simulations were performed to validate our proposed robotic triage approach. For each simulation, we used a 80m×50m rectangular environment with the rendezvous point located at $(40, 25)$. Each simulation consisted of a random number and location of tasks with 3 UGVs, 1 UAV, and 1 medic.

Fig. 6 shows a series of snapshots for our proposed triage approach for a case study with 5 *priority 3* (red), 3 *priority 2* (yellow), and 3 *priority 1* (green) tasks scattered in the environment.



(a) $t = 9$s (b) $t = 13$s (c) $t = 18$s (d) $t = 21$s

Fig. 6: Sequence of snapshots demonstrating our proposed approach in action with 3 UGVs, 1 UAV, and 1 medic.

The different behaviors discussed in the approach section can be seen in these snapshots. At $t = 9$s (Fig. 6(a)), the ground robots are exploring the environment on their respective paths. The green UGV is making its way back to resume the exploration from where it left off after finding and reporting a high-priority task to the UAV at the rendezvous location. In the same snapshot, the UAV is flying toward the medic to deliver the task. Note also that the red UGV doesn't report yet to the UAV the two lower-priority tasks (yellow) found during its exploration.

At $t = 13$s (Fig. 6(b)), the green UGV has found two other high-priority tasks and is navigating to the rendezvous location to report them. Here, we highlight the *continue explore* behavior presented in Fig. 3 and in the UGV section: the UGV navigates to the first high-priority task to check if other tasks are available; it finds another high-priority task. It navigate to this task but doesn't find other tasks; thus it proceeds to report these two tasks to the UAV. In our implementations we used $N = 5$.

At $t = 18$s (Fig. 6(c)), the UAV is traveling to the medic to alert them about the two new red tasks found by the green UGV. The green UGV is going back to continue its exploration while the blue UGV is at the rendezvous location waiting to report a red task, that it found during its exploration, to the UAV.

At $t = 21$s (Fig. 6(d)), the UAV is navigating back to the rendezvous point where it will collect the blue robot's tasks, while the medic is making their way to the two red tasks it just received.

The final path of the medic is shown in orange in Fig. 7. As can be noted, the medic visits high-priority tasks before taking care of lower-priority ones. The medic receives the location of the fifth high-priority task – located at $(10, 15)$ – only after visiting the priority 2 task (yellow) at $(55, 12)$.

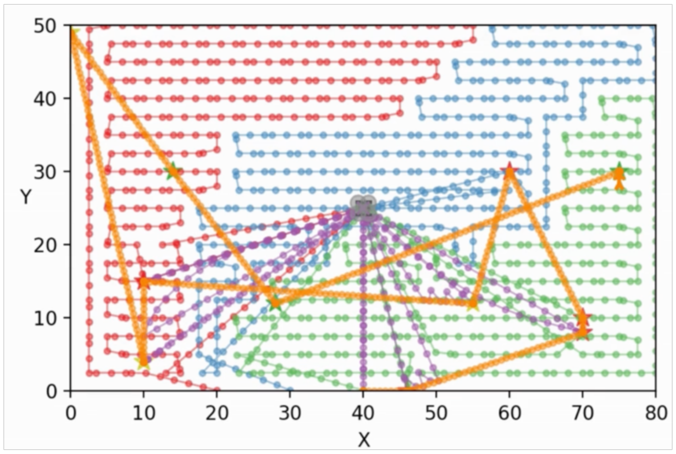


Fig. 7: The final paths followed by the robots and medic.

Comparisons with other methods: Our proposed prioritized robotic triage approach was compared with the following methods.

No priority method: In this case the priority of the victim's injuries is not taken into account when the medic decides which tasks to visit first or when the UGV decides when to report tasks. The medic's decision of what task to visit is based only on the distance to the task as if all tasks have the same high priority. The UGV reports any task they find, regardless of that tasks priority. The ground vehicle continues to search for any additional tasks in the area after finding a victim, regardless of priority. The final path of the medic for this method is shown in orange in Fig. 8(a).

No UAV relay method: The second method tested is when there is no UAV present in the environment and the UGVs follow the same behavior of our approach, i.e., they deliver to the medic only when they find a high-priority task. This leads to the UGVs having to travel to and communicate their tasks to the human agent themselves. The final path of the medic for this method is shown in purple in Fig. 8(b).

Constant Communication method: The third method tested considers constant communication everywhere in the environment, that is, the UGVs are able to communicate tasks to the medic without traveling to the medic's location. There is no UAV in this method. The medic here executes tasks as they are received following the prioritized TSP solution presented in our approach. This method would require the UGVs to have the functionality to share information over a network, which is often difficult in disaster triage situations where a communication infrastructure may not exist or be damaged. The final path of the medic for this method is shown in red in Fig. 8(c).

We ran each method on 10 different cases studies and performed a statistical analysis with the following metrics:
i) a score s to evaluate how well the medic served tasks based on priorities, mathematically:

$$V = ||V^* - V_m|| \tag{1}$$

$$s = \sum_{i=1}^{N_T} V(i) \tag{2}$$

where V^* is a vector containing an ordered list of the tasks based on priorities, V_m is a vector containing the actual final order of task priorities served by the medic, and N_T is the total number of tasks in the environment. To create a numerical score, each priority type inside V^* and V_m has a numerical value assigned: for example, in our simulations we chose *priority 3*=100, *priority 2*=50, and *priority 1*=20;
ii) the total time to complete the mission;
iii) the UGV average distance traveled which is also proportional and an indicator of the average energy consumed; and
iv) the total distance traveled by the medic.

When comparing the three alternate methods to our approach in Table I, we can see that our method performs best in the time and score metrics. This implies that our method of robotic triage ensures a time-efficient process and that high-priority victims are treated first.

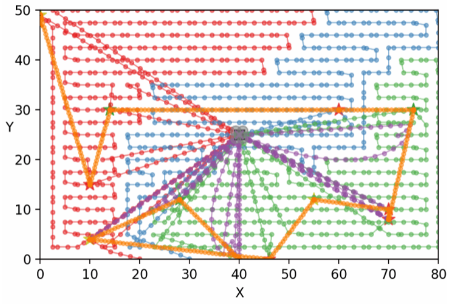
TABLE I: Comparison between different methods

Method	Time	Score	UGV Avg. Distance	Medic Distance
Our Approach	**271.2779**	**10**	645.0055996	218.5377444
No Relay	353.4441	20	671.2213625	215.1282713
No Priority	504.378	312	724.6349834	**158.9780604**
Constant Communication	278.6356	84	**549.970609**	222.4307906

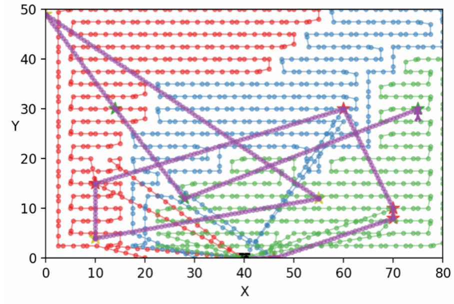
Additionally, our method performs well in the distance traveled by the UGVs, second only to the constant communication method where the UGVs do not have to travel to the UAV or medic to report tasks. The method with no priority may reduce the amount of distance the medic travels since the medic is not required to triage higher-priority tasks first. The medic decides which tasks to triage based on their distance from the task, leading to a shorter distance traveled overall. Similarly, the constant communication method naturally has the lowest UGV average distance traveled since in this case the robot doesn't need to travel anywhere to report its found tasks and can simply continue following its path.

B. Hardware Experiment

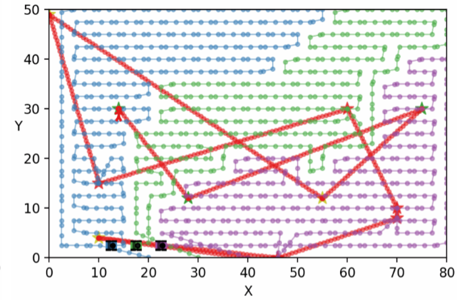
Fig. 9 shows results for a real-world experiment in which two Husarion Rosbot2 UGVs, a medic implemented with a Rosbot2 wearing a yellow helmet, and a Bitcraze Crazyflie 2.0 quadrotor UAV are tasked to find, triage, and serve tasks in unknown locations in the environment. At $t = 28$s, (Fig. 9(a)), UGV1 finds a high-priority task (red) and travels to the rendezvous point to report it to the UAV. At $t = 30$s (Fig. 9(b)), UGV1 is returning to its designated path, and the UAV is traveling to the medic to report the tasks found by UGV1. At $t = 49$s, (Fig. 9(c)), the medic makes its way to the task reported by UGV1. The UAV returns to the



(a) No priority method

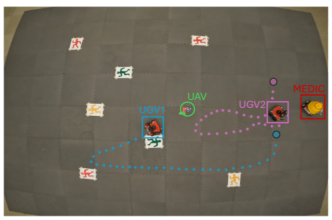


(b) No UAV relay method

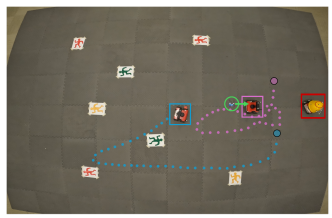


(c) Constant communication method

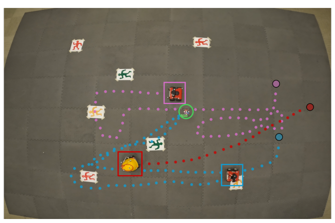
Fig. 8: Comparison of medic final paths for three different methods.



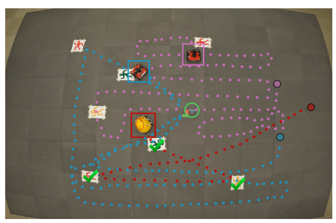
(a) $t = 28$s



(b) $t = 30$s



(c) $t = 49$s



(d) $t = 93$s

Fig. 9: Experiment results with 2 UGVs, 1 UAV, and 1 medic.

rendezvous point after reporting the task to the medic. At $t = 93$s, (Fig. 9(d)), the medic travels to a high-priority task in the upper left of the environment while UGV2 travels to the rendezvous point after finding a high-priority task in the upper right quadrant of the environment. The medic ends up visiting first priority 3 tasks before switching to priority 2 and priority 1 tasks if no other high-priority tasks are available, demonstrating the applicability of our approach with a real heterogeneous robotic system.

VII. Conclusions

In this work, we have presented an optimized and time-efficient framework for human-robot triage. Our approach leverages heterogeneous robotic systems in a coordinated exploration, delivery, and task assignment and execution framework. Results from comparisons with other methods show that our approach performs the fastest, produces the best priority score, and records a low average UGV distance, second only to an ideal method with constant communication. Although our method has a higher medic distance traveled than some of the other methods, it completes high-priority tasks faster than other methods, aiding victims with lower survivability rates in a shorter time frame.

Future extensions to this work could further aid in implementing a robotic system into the triage process: for example, a medic could recruit the lowest-priority victims after treating them to help treating other patients or to find victims similar to the UGV's behavior. In future work, we plan to include more complex environments with obstacles, changes in terrain conditions, and disturbances. A deeper study on how the number of medics, UGVs, or UAVs affects the metrics is also on our agenda to better assess the presented metrics and performance of the proposed solution. From an implementation point of view, future work could include considerations on the types of sensors to be used to enable robotic triage in real outdoor environments and conditions. Additionally, more realistic assumptions on the medic dynamics, such as fatigue and mental load, are needed to further improve the proposed method.

Acknowledgments

This work is sponsored by Northrop Grumman through the University Basic Research Program. The authors would like also to thank Lt. Col. Joel Herness with the US Air Force for useful discussion on triage in military operations.

References

[1] J. Bazyar, M. Farrokhi, A. Salari, and H. R. Khankeh, "The principles of triage in emergencies and disasters: a systematic review," *Prehospital and disaster medicine*, vol. 35, no. 3, pp. 305–313, 2020.

[2] M. C. Bhalla, J. Frey, C. Rider, M. Nord, and M. Hegerhorst, "Simple triage algorithm and rapid treatment and sort, assess, lifesaving, interventions, treatment, and transportation mass casualty triage methods for sensitivity, specificity, and predictive values," *The American journal of emergency medicine*, vol. 33, no. 11, pp. 1687–1691, 2015.

[3] S. Park, Y. Oh, and D. Hong, "Disaster response and recovery from the perspective of robotics," *International Journal of Precision Engineering and Manufacturing*, vol. 18, pp. 1475–1482, 2017.

[4] R. K. Senthilkumaran, M. Prashanth, H. Viswanath, S. Kotha, K. Tiwari, and A. Bera, "ARTEMIS: AI-driven Robotic Triage Labeling and Emergency Medical Information System," *arXiv preprint arXiv:2309.08865*, 2023.

[5] A. Ranjha, G. Kaddoum, and K. Dev, "Facilitating urllc in uav-assisted relay systems with multiple-mobile robots for 6g networks: A prospective of agriculture 4.0," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 7, pp. 4954–4965, 2021.

[6] V. Shah, M. Heeter, J. Vallarino, P. Sherman, L. Bramblett, and N. Bezzo, "Evolutionary-based coordination of multi-robot systems with dynamic constraints," in *2024 Systems and Information Engineering Design Symposium (SIEDS)*. IEEE, 2024, pp. 7–12.

[7] A. C. Kapoutsis, S. A. Chatzichristofis, and E. B. Kosmatopoulos, "Darp: Divide areas algorithm for optimal multi-robot coverage path planning," *Journal of Intelligent & Robotic Systems*, pp. 1–18, 2017.