2024 IEEE 20th International Conference on Automation
Science and Engineering (CASE)
August 28 - September 1, 2024. Bari, Italy

Practical Heuristics for Victim Tagging During a Mass Casualty Incident Emergency Medical Response

Maria Ana Cardei[1] and Afsaneh Doryab[1]

***Abstract*— Mass casualty incidents (MCIs) are a growing concern. Recent efforts to improve the MCI emergency medical response focus on resource allocation, responder coordination, or transportation efficiency, and often overlook the initial, yet critical, step of tagging victims on scene. Tagging victims in minimal time is vital in providing information that guides subsequent time-constrained response actions. In this paper we present a mathematical formulation of multi-agent victim tagging to minimize the time it takes for responders to tag all victims. Five distributed heuristics are formulated and evaluated with simulation experiments. The heuristics considered are on-the go, practical solutions that represent varying levels of situational uncertainty in the form of global or local communication capabilities, showcasing practical constraints. Extensive simulations demonstrate that local methods are more efficient for adaptive victim tagging, specifically choosing the nearest victim with the replanning option.**

I. Introduction

A mass casualty incident (MCI) is defined as a situation in which casualties greatly outnumber available local resources, which eventually overwhelms the local healthcare system within a time frame [1]. Some common types of MCIs involve natural disasters such as wildfires, hurricanes, and earthquakes or terrorist attacks such as mass shootings and bombings. When emergency medical responders arrive at the scene of an MCI, an initial task is to locate individuals and assess their injuries promptly, known as triage, followed by physically tagging them with a color-coded label corresponding to injury severity. All victims are tagged before assessing the next steps. Fast and efficient victim tagging is a critical initial stage in the MCI response process, providing vital information that guides subsequent decisions. It enables the timely assessment of the total number of victims, their distribution based on the severity of injuries, and their respective locations throughout the MCI environment.

Most of the current work done to improve the emergency response to MCIs has a focus on improving resource allocation [2], responder coordination [3], [4], or transportation to treatment facilities [5]. In [6], authors model the rescue, treatment, and transportation of MCI victims to hospital locations, testing communication modes between agents. Tagging victims during the rescue process is overlooked and assumed. Various works show there is a lack of research done in the on-scene or pre-hospitalization stage, specifically the task of tagging victims fast. To our knowledge, the problem of minimizing victim tagging time has not been formally addressed. Therefore we leverage insights from MCI response and from similar fields such as search and rescue, to compare and evaluate efficient, practical heuristic solutions. We aim to discover insights that can be valuable to emergency departments and inform victim tagging practices.

In this paper we study the victim tagging problem for a team of medical responders. We formalize victim tagging as an integer linear programming problem and evaluate and compare five distributed, on-the-go responder methods to minimize the time it takes to tag all victims under varying levels of uncertainty in communication. Our experimental results show that the Local Nearest Victim Policy outperforms others, and local policies (under greater uncertainty) are more efficient than global policies for adaptively choosing the next victim to tag. Individual policy analyses give us additional insights into the number of responders needed for various victims and the efficiency of each policy, measured in the time it takes to tag victims, allowing for the possibility of improving MCI guidelines. The contributions of the paper are summarized as follows:

1) We formulate MCI emergency medical response victim tagging as an integer linear programming problem.
2) We design five distributed, on-the-go heuristic solutions considering global and local agent communication and analyze their performance through simulations. Results indicate that a local, nearest victim tagging solution minimizes victim tagging time in uncertain situations, as compared to four other practical heuristics.

The rest of this paper is organized as follows. Section II introduces the related work. Section III gives the problem definition and the mathematical formulation of the constrained multi-agent victim tagging problem. Section IV presents the five heuristics. The experimental setup and results are given in Section V, followed by the conclusion in Section VI.

II. Related Work

A problem similar to victim tagging during an MCI is search and rescue (SAR). Many studies evaluate SAR as a task allocation problem, considering a team of unmanned aerial vehicles (UAVs), heterogeneous vehicles [7], [8], [9], or generally robotic teams [10]. SAR involves the search and then the rescue phase that occurs once the positions of the individuals are known [9]. The vehicle performing the task may be allocated a particular known region to explore. Often the solutions to these problems are solved in an offline manner where tasks are allocated to agents

This paper is based upon work supported by the National Science Foundation Research Traineeship (NRT) program under Grant No.1829004.

[1]Maria Ana Cardei and Afsaneh Doryab are with the Department of Computer Science, University of Virginia, VA, 22904, USA. Correspondance: cbr8ru@virginia.edu, ad4ks@virginia.edu

2024 IEEE 20th International Conference on Automation Science and Engineering (CASE) | 979-8-3503-5851-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/CASE59546.2024.10711469

before the start of the search phase [11], [8]. The victim tagging problem is different because it requires practical on-the-go/online approaches that assume uncertainty for the responder team when they arrive on scene. While other work proposing solutions for similar problems assume complex robotic systems, our heuristics are intuitive and can be implemented in the real world by human responder teams, as well as generalized to heterogeneous or fully robotic teams.

In [9], authors propose a distributed task allocation method that employs local centralization by implementing planners to lead different areas. The technique's performance is compared to market-based consensus-based bundle algorithm (CBBA). Market, or more specifically auction-based methods, such as CBBA and the dynamic auction approach proposed in [10], consider one agent as the "auctioneer" while other agents bid on tasks to be allocated [12]. These mechanisms will not work well in decentralized cases where there is a lack of global communication between agents. Additionally, one drawback is that once they produce a global solution, they do not make any further effort to improve it. Finding solutions adaptively for each independent agent can be a way to mitigate this limitation. In [13] authors consider a task discovery and allocation problem with limited connectivity between robot agents. They consider a rendezvous point for agents to meet and share information discovered, which is helpful in communicating tasks, but can take away from valuable time constraints.

In [14], authors propose an online, decentralized genetic algorithm for multi-agent SAR. They introduce a handover value for each agent, which agents communicate to one another to decide who is allocated which task. This value needs to be communicated globally between agents. In [8], authors propose a centralized SAR task allocation algorithm based on particle swarm optimization and show that it outperforms some distributed solutions. For both [14] and [8], challenges will be faced when there is a lack of global communication. In our heuristics, we consider local and global communication constraints to test realistic challenges for the MCI responder team. Our methods are distributed, allowing for increased uncertainty and on-the-go victim tagging, making our solutions practical.

III. Problem Description

We consider the victim tagging problem generally where medical responders need to identify and tag victims, and the aim is to minimize the total time it takes to tag all victims. Medical responders are referred to as responders or responder agents throughout the paper, and can be generalized to make up human, heterogeneous, or fully robotic teams. In this section we set up the preliminaries, and define the problem mathematically.

A. Preliminaries

Let us consider a team of n responders $R = \{r_1, \dots, r_n\}$. A responder r_i is characterized by their speed (w_i) and the time it takes to triage a victim v_k (τ_{ik}). The responder agents have a goal to tag all m victims $V = \{v_1, \dots, v_m\}$. We use an $m \times m$ matrix D to represent the distances between pairs of victims. D_{jk} denotes the distance between victims v_j and v_k. If v_j and v_k are the same victim, we assume the distance is 0, i.e. D_{jk} = 0. In this paper we assume all responders start from the same position that we denote with index 0, therefore D_{0k} indicates the distance from the starting point to v_k.

B. Problem Formulation

The multi-agent victim tagging problem defined above is a combinatorial optimization, which can be formulated and solved using integer linear programming (ILP). The problem is defined as minimizing a linear objective function (the maximum time it takes to tag all victims) subject to linear constraints.

We use TG_{ijk} to represent the time cost of r_i moving from v_j to v_k, represented as

$$TG_{ijk} = \frac{D_{jk}}{w_i}. \tag{1}$$

The moving paths of each responder can be summarized using a three dimensional matrix $X = \{x_{ijk} | i \in \{1, \dots, n\}, j \in \{0, \dots, m\}, k \in \{1, \dots, m\}\}$. Each element x_{ijk} is a binary variable that represents whether responder agent r_i moves from v_j to v_k. The binary values can be defined as

$$x_{ijk} = \begin{cases} 1 & \text{if } r_i \text{ moves from } v_j \text{ to } v_k, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Note that j and k represent the victim index, except in the case where $j = 0$, which indicates that r_i is at the starting location ($k \neq 0$ since responder agents do not need to return to their starting location).

Then the victim tagging problem can be formulated as the following minmax ILP:

$$Obj = \min \left\{ \max_{1 \le i \le n} \left\{ \sum_{j=0}^{m} \sum_{k=1}^{m} (TG_{ijk} + \tau_{ik}) x_{ijk} \right\} \right\}, \tag{3}$$

where Obj is the objective function describing the goal to minimize the maximum time it takes for all responders to tag victims. It is subject to the following constraints:

$$\sum_{i=1}^{n} \sum_{k=1}^{m} x_{i0k} \le n \tag{4}$$

Equation (4) specifies that at most n responders leave the starting position.

$$\sum_{i=1}^{n} \sum_{j=0}^{m} x_{ijk} = 1, \quad \forall k \in \{1, \dots, m\} \tag{5}$$

Equation (5) shows that exactly one responder tags each victim v_k.

$$\sum_{i=1}^{n} \sum_{k=1}^{m} x_{ijk} \le 1, \quad \forall j \in \{1, \dots, m\} \tag{6}$$

Equation (6) specifies that at most one responder leaves each victim v_j. When all victims are tagged, responders remain at the last victim they tag.

$$\sum_{j=0}^{m}\sum_{k=1}^{m} x_{ijk} \le m, \quad \forall i \in \{1,\dots,n\} \tag{7}$$

Equation (7) illustrates that each responder r_i tags at most m victims.

$$\sum_{j=0}^{m} x_{ijk} + \sum_{\substack{a=1\\a\neq i}}^{n}\sum_{b=1}^{m} x_{akb} \le 1, \forall i \in \{1,\dots,n\}, \forall k \in \{1,\dots,m\} \tag{8}$$

Equation (8) ensures that if a responder r_i tags victim v_k, then only r_i can leave v_k.

$$x_{ijk} \in \{0,1\}, \quad \forall i \in \{1,\dots,n\}, \forall j \in \{0,\dots,m\}, \quad \forall k \in \{1,\dots,m\} \tag{9}$$

Equation (9) describes that the path of r_i from v_j to v_k is either 0 or 1.

$$u_{i0} = 0 \quad \forall i \in \{1,\dots,n\} \tag{10}$$

$$1 \le u_{ik} \le m \quad \forall i \in \{1,\dots,n\}, \forall k \in \{1,\dots,m\} \tag{11}$$

$$u_{ij} - u_{ik} + 1 \le m(1 - x_{ijk}), \quad \forall i \in \{1,\dots,n\}, \forall j \in \{0,\dots,m\}, \quad \forall k \in \{1,\dots,m\} \tag{12}$$

$$u_{ij} \in \{0,\dots,m\}, \quad \forall i \in \{1,\dots,n\}, \forall j \in \{0,\dots,m\} \tag{13}$$

Equations (10 - 13) ensure that the paths of the n responders do not contain cycles (i.e. they do not return to a previously tagged victim). Here, we extend the Miller-Tucker-Zemlin (MTZ) formulation for the Traveling Salesman Problem [15], [16] to n paths. Equation (13) specifies that u_{ij} from the MTZ-based formulation accounts for all n responders and m victims.

IV. Approach

Equations (3 - 13) describe the victim tagging problem assuming optimal conditions, with centralized, global information available and optimal communication within the responder team. It considers all possible victim tagging cases for each responder, which is laborious and time-consuming. The tagging problem for multiple responder agents is combinatorial in nature and NP-Hard [8]. The complexity of the problem grows exponentially as the number of variables increases, thus it is not suitable for real-time practical applications. Additionally, in a real MCI scenario, global information and optimal communication are not always available. Therefore, efficient heuristic solutions are presented in this section with varying degrees of uncertainty.

We focus on the practical aspect of MCI victim tagging in our solutions, and consider solutions (referred to as policies) that are commonly used in emergency response or related fields. Unlike other heuristics such as CBBA [7], [12], particle swarm [8], and genetic algorithms [14], ours are practical and intuitive approaches that are used in practice for MCIs, or could easily be applied by responder teams in chaotic, stochastic MCI scenarios on-the go, without having extensive prior knowledge of the situation. While based on previously existing methods, the five policies we explore, to our knowledge, have not been formally defined and quantitatively compared in an effort to optimize victim tagging time. The comparison and evaluation of these policies can serve as a guideline and inform emergency departments about optimal victim tagging practices.

Each policy we present acts as the overarching responder team policy, which allows for responders to individually and adaptively identify their next victim to tag. For the following policy descriptions, we refer to the previously defined term x_{ijk}, which describes whether responder agent r_i moves from a victim with index j to a victim with index k. Initially all $x_{ijk} = 0$ and $j = 0$ for all responders. The responder team's policy will determine how an individual responder locally and iteratively determines k, which then becomes j when moving to the next victim to tag. The selection of k for each responder agent happens on-the go, individually for each responder in a distributed manner. Therefore, when victim tagging has completed, a sequence of k values determines the order of tagged victims for one responder, and $k = j + 1$ in the sequence of victims tagged.

To explain our heuristics, we introduce responder attribute a_i, which is initialized as Null, then becomes the victim that r_i has selected to tag next, and continues to update as r_i tags victims based on the policy. A victim v_j is characterized by the responder that is going to tag them (f_j) and whether they have been tagged yet (g_j), where $g_j \in \{\{0,1\}|0 = \text{untagged}, 1 = \text{tagged}\}$. The following subsections formally present the five policies we analyze as methods for a responder r_i to identify the next victim to tag (v_k). For any definition of k, if k is undefined, then k = j indicating the responder is idle. For all formalizations, *dist*(a, b) refers to the distance between agents a and b.

A. Random Victim Policy (RVP)

In RVP, a responder r_i chooses a random victim v_l that has not been tagged yet, and one who is not chosen by a different responder to tag. Using previously defined responder and victim attributes, the victim's index can be mathematically formulated as

$$k = \{\text{random } v_l \in V \mid ((v_l \notin a_i, \forall r \in R) \wedge g_l = 0)\}. \tag{14}$$

RVP represents the tagging method that assumes all globalized and centralized information, with minimal uncertainty,

TABLE I: Variables used to formalize our solutions.

Symbol	Description
R	Set of responder agents
n	Number of responders
r_i	The ith responder
a_i	Selected victim for r_i
τ_{ik}	Time it takes r_i to triage v_k
w_i	Speed of r_i
c_i	Cell for r_i
s_i	State of r_i
V	Set of victim agents
m	Number of victims
v_j	The jth victim
f_j	The responder that selected v_j
g_j	Whether v_j has been tagged
h_j	Health state of v_j
p_j	Position of v_j

and therefore acts as our reference scenario serving as our benchmark. The four following policies build on this one and allow for increasing uncertainty.

B. Nearest Victim Policy (NVP)

In NVP a responder r_i chooses the nearest victim that has not been tagged yet, and is not chosen by a different responder as their target victim to tag. This is represented as

$$k = \arg\min_{v_l \in V}\{\text{dist}(r_i, v_l) \,|\, ((v_l \notin a_i, \forall r \in R) \wedge g_l = 0)\}. \tag{15}$$

NVP is loosely based on a practical heuristic in [6]. This policy continues to be applicable in situations that allow for global communication and perception.

C. Local Nearest Victim Policy (LNVP)

LNVP describes the situation where responder r_i locally chooses the nearest victim as their next victim to tag. r_i can choose a victim that is chosen by a different responder, in which case the other responder may need to replan their next steps based on this on-the-go iterative process. This is formulated below:

$$\begin{aligned} k = \arg\min_{v_l \in V}\{&\text{dist}(r_i, v_l) \,|(g_l = 0 \wedge (f_l = \text{NULL} \\ &\vee (\text{dist}(v_l, f_l) > \text{dist}(r_i, v_l) \wedge \text{dist}(v_l, f_l) > \epsilon)))\}, \end{aligned} \tag{16}$$

where ϵ acts as a threshold and can be any value or function indicating how far a victim's current tagger has to be from the victim so that they can reroute if needed. LNVP is similar to NVP, except it is a heuristic that allows for increased uncertainty. In LNVP, we assume responders do not have the ability to communicate globally, only locally.

D. Local Critical Victim Policy (LCVP)

In LCVP we introduce a new victim attribute h_j, which indicates victim v_j's health status. This can be defined using any criteria, but we represent it following the START (Simple Triage and Rapid Treatment) algorithm, which is commonly used in MCIs to correspond triage tags to health status [17]. For v_j, if $h_j \in [0, 1]$, we can denote (1) Black (dead) when $h_j \in [0, 0.25)$, (2) Red (immediate) when $h_j \in [0.25, 0.50)$, (3) Yellow (delayed) when $h_j \in [0.50, 0.75)$, and (4) Green (mobile) when $h_j \in [0.75, 1]$. Health status is an internal measure to the victim, so we make an assumption that if a responder sees a victim, they can visually determine whether the victim is critically injured (dead or almost dead) or less injured (mobile or slightly injured). Thus, for LCVP we define a critical victim as a victim with attribute $h < 0.5$.

For LCVP, a responder r_i chooses to tag the nearest critical victim, and then follows LNVP when all critical victims are tagged. This is similar to [6] where they utilize memory guided finding, except that in our case the responder agents are not aware of the exact injury severities, and they have not previously seen the MCI environment. This is formally defined here:

$$\begin{aligned} k = \arg\min_{v_l \in V}\{&\text{dist}(r_i, v_l) \,|\, (g_l = 0 \wedge h_l < 0.5 \\ &\wedge (f_l = \text{NULL} \vee (\text{dist}(v_l, f_l) > \text{dist}(r_i, v_l) \\ &\wedge \text{dist}(v_l, f_l) > \epsilon)))\}, \end{aligned} \tag{17}$$

$$\text{if } (\neg\exists v_l | h_l < 0.5) \wedge (\exists v_l | g_l = 0), \text{ then use } LNVP. \tag{18}$$

Here, ϵ acts the same as in LNVP. Equation (18) indicates that after all critical victims are tagged, and there are still victims that have not been tagged, then r_i will follow LNVP to find k. For LCVP we assume responders do not have the ability to communicate globally, they must communicate ad hoc locally.

E. Local Grid Assignment Policy (LGAP)

For LGAP, the MCI environment is divided into cells, so that each responder is assigned to a cell in the grid. For this policy, we introduce a set of cells $C = \{c_1, \ldots, c_n\}$ describing cell areas for n responders, a responder attribute c, and a victim attribute p. In LGAP a responder r_i tags the nearest victim within their cell c_i until all victims in their cell are tagged. A victim v_j's position in the MCI environment is represented as p_j. LGAP is formalized as follows:

$$k = \arg\min_{v_l \in V}\{\text{dist}(r_i, v_l) \,|(p_l \in c_i \wedge g_l = 0)\}. \tag{19}$$

We designed LGAP based on commonly used practical SAR heuristics where different specified areas should be explored [11]. For each aforementioned policy we assumed that responders have a global view of the MCI environment and the victims within it. LGAP allows for the most uncertain conditions, where only local communication and limited perception exist. We assume that the perception of a responder agent is equal to or greater than their cell size. The variables used to summarize our heuristics are summarized in Table I.

V. Experiments and Results

This section presents the experimental setup and performance evaluation of the five victim tagging policies to demonstrate their effectiveness in comparison to one another.

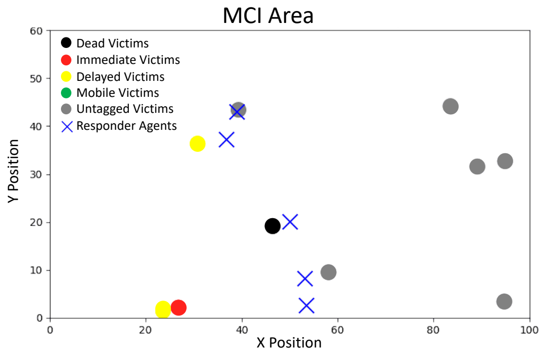

Fig. 1: Simulation frame from scenario with 5 responder agents and 10 victims agents. The responders enter the scene from the bottom left-hand corner and tag victims.

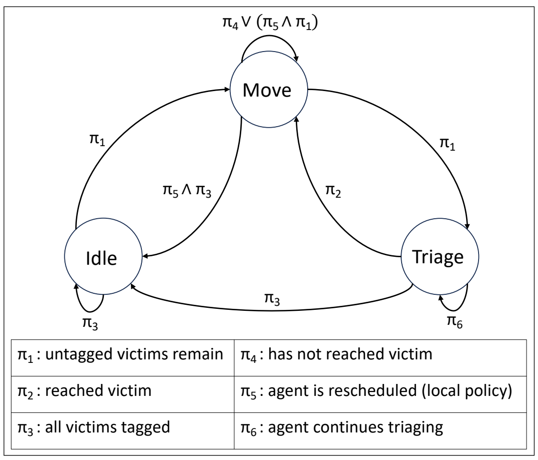

Fig. 2: Responder agent finite state machine.

A. Simulation Setup

For our experiments, we designed a multi-agent system (MAS) consisting of responder and victim agents in an MCI environment, and created simulations utilizing agent-based modeling to extensively test various responder policies. We programmed the simulations in Python (v. 3.8.8) and used the open-source agent framework Mesa [18] for the MAS design. The MAS environment has continuous space, runs in discrete time steps, and utilizes a scheduler that activates agents in a random order at each time step. The simulation iterates forward by one time step until all victims are tagged.

The MCI environment is staged as a 2-dimensional rectangular space with area A, as shown in Fig. 1. The responder agents tag victims following the responder policy until they are all tagged. When victims are visited by a responder, they are triaged and tagged as either black (dead), red (immediate), yellow (delayed), or green (mobile) following the START triage algorithm [17].

Victim positions are randomly, uniformly distributed throughout the MCI environment, and we assume they are idle. For the purpose of our experiments, we design a finite state machine (FSM) to represent the responder agents' state and transitions between them, observed in Fig. 2. This lends to an additional responder agent attribute s_i for responder

TABLE II: Simulation parameters, where t is time and values are arbitrarily chosen.

Parameter	Symbol	Value
MCI Area	A	100 x 60 units
Time to triage any victim	τ	3 time steps
Victim health state	h	random value $\in [0, 1]$
Victim position	p	random value $\in A$
Responder speed	w	1 unit/time
Responder start position	$p_{t=0}$	0
Responder start state	$s_{t=0}$	*idle*

TABLE III: Number of responder (No. R) and victim (No. V) agents for each experiment are shown. Results highlight average time steps it takes to tag all victims for each experiment, over fifty iterations. Bold values highlight the most efficient policy for each experiment.

Experiment	1	2	3	4	5	6	7	8	9
No. R	5	5	5	5	20	20	80	80	320
No. V	10	20	100	1000	100	1000	100	1000	1000
Results (average time steps)									
RVP	136	226	956	9,135	299	2,381	122	657	231
NVP	124	152	298	1,328	164	**411**	132	218	152
LNVP	118	**145**	**288**	**1,316**	**153**	420	**111**	**190**	**118**
LCVP	**115**	174	375	1,573	197	506	117	258	136
LGAP	126	161	317	1,383	177	470	148	238	146

r_i's state, where $s_i \in \{idle, move, triage\}$. All responder agents begin in the idle state and then transition between the move and triage states following transitions denoted with π. When in the triage state, a responder r_i is in the process of tagging a victim v_k, and is done after some number of triage steps determined by τ_{ik}. The responder attribute p_i for r_i additionally describes their position at any given time. Our simulation parameters are summarized in Table II, which can be easily altered based on simulation goals.

B. Evaluation

We devise experiments with varying numbers of agents to represent nine MCI scenarios of different scales to evaluate which policy optimally minimizes victim tagging time, shown in Table III. The number of victims chosen are drawn from one MCI incident plan [19], and we choose responder numbers to be less than the number of victims. For each of these nine experiments, we test each of the five policies, for a total of forty-five experimental scenarios. For each of these experiments, fifty iterations are run and we report the average results. Each experiment is evaluated by using the objective function in Equation (3) to find the minimal time it takes to tag all victims.

Table III reports the average number of time steps it took to tag all victims for each experiment, comparing policies. Policies with global communication between responder agents (RVP and NVP) are distinguished from the policies with local communication (LNVP, LCVP, and LGAP). LNVP performed optimally, with the minimum average time step values across almost all experiments, while

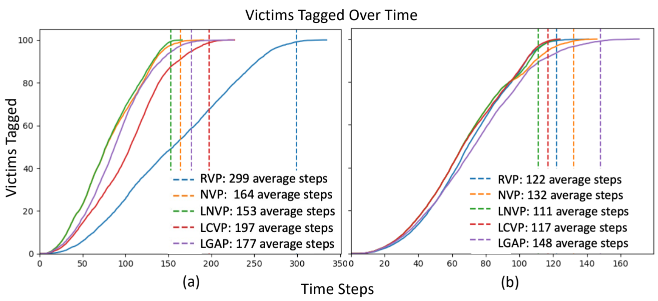

Fig. 3: Policy comparison for number of victims tagged over each time step for 100 victims and (a) 20 or (b) 80 responders. Each color curve denotes a different policy, and the dotted vertical line shows the average time it takes to tag all victims for each policy.

RVP performed the least efficiently. Each policy's average time to tag all victims increases as the ratio between the number of responder and victim agents grows. We notice RVP's values elevate increasingly compared to other policies. For the tests with 5 responders (experiments 1-4), when the number of victims increase, the values for RVP increase from 136, 226, 956, to 9,135 respectively, while values are smaller for others. This is because of RVP's nature; responders choose to tag a random victim next, which could be the furthest victim away, and this results in more time steps when the number of victims increases. This emphasizes the importance in selecting a quality policy for responders to adaptively tag victims. We also note that although policies LNVP, LCVP, and LGAP assume increasingly uncertain conditions compared to RVP and NVP, they perform well. This indicates that with uncertain conditions that arise from a lack of global communication, local, iterative, and distributed victim tagging for responders can provide efficient victim tagging time.

Graphs (a) and (b) in Fig. 3 show a policy comparison in how many victims are tagged over each time step for 100 victims, and 20 or 80 responders, respectively. It is interesting to see that for both cases, each policy's curve is an S-curve, indicating a sigmoidal relationship between the victims tagged and time. In Fig. 3(a) NVP, LNVP, and LGAP have curves that are steeper, which suggests that these policies are more effective in tagging victims quicker as more victims are tagged per each time step. Additionally, the NVP and LVP curves overlap from the start ($t = 0$) until about $t = 75$, where about 50 victims have been tagged. This indicates that, for any number of victims up to 50, 20 responders tag the same number of victims each time step regardless of whether in a global or local communication scenario. Similarly, other parts of the curves overlap, such as the NVP and LGAP curves at around $t = 112$, suggesting that policies have similar effectiveness at specific time steps with a particular number of victim and responder agents.

In both graphs of Fig. 3, curves overlap, demonstrating that if that particular number of victims and responders exists, then some policies can be interchangeable, thus giving more freedom in the choice of responder policy. In Fig. 3(b), LGAP performs the least efficiently taking an average of 148 time steps until all victims are tagged. In LGAP, responders only tag victims within their designated cell, thus some tag more than others. This result highlights the inconsistency of LGAP's performance, as the locations of victims is very influential. Fig. 3(b) shows that, in comparison to 3(a), all policies' curves are similar at this scale, indicating that with 80 responders, tagging victims is smoother and more efficient than with 20 responders when there are 100 victims. This also suggests that paying importance to policy selection is most relevant when the ratio between victims and responders is larger.

Fig. 4 presents results that give additional insights into each policy. Row (I) shows total victims tagged over each time step with 5, 20, and 80 responders shown for each policy. All policies have similar curves with slight variations, except RVP has drastic separation between each responder amount, and it has a much larger range of time steps. Comparing the policies, LCVP's curves have a greater disparity at the start of the experiment, as well as a larger difference between the case of 20 and 80 responders. This indicates that LCVP could be less effective at the start of tagging 100 victims and there makes a bigger difference between the ratio of responders to victims for this policy. Results also suggest that if a responder team wants to implement LCVP as a way to prioritize critical victims, then they should strongly consider the number of responders to dispatch in an MCI scenario involving 100 victims.

Row (II) in Fig. 4 shows the average time steps it takes to tag 100 victims for 5, 20, and 80 responders for each policy. NVP results look visually similar to those of LNVP, however for the case with 20 responders the NVP curve increases with a greater slope indicating an inability to tag victims as quickly as LNVP. The ability for responders to communicate locally proves to benefit their iterative process of identifying the next victim to tag.

Row (III) in Fig. 4 shows the average time steps it takes to tag 10, 20, and 100 victims for up to 5 responders, for each policy. The graphs look very similar, indicating the common trend of a decrease in time it takes to tag victims when there are more responders. However, RVP is a much larger scale with it taking over 4,000 time steps to tag all victims when there is 1 responder. Graphs like these can be used as a valuable tool to determine the number of responders to dispatch, based on the number of victims estimated in an MCI. For example, analyzing all policies except for RVP, the curves all converge at similar values. The curves for the cases with 10 and 20 victims look very similar, and at around the point where there are 3 responders the lines converge. Therefore 3 responders could be a sufficient amount to dispatch for the case where there are either 10 or 20 victims across these policies. This is especially relevant when resources are scarce and scheduling and resource allocation come into play. Similarly, if there is an existing deadline, such as 1,500 time steps, if there are 1-100 victims, it may be sufficient to dispatch 2-5 responders

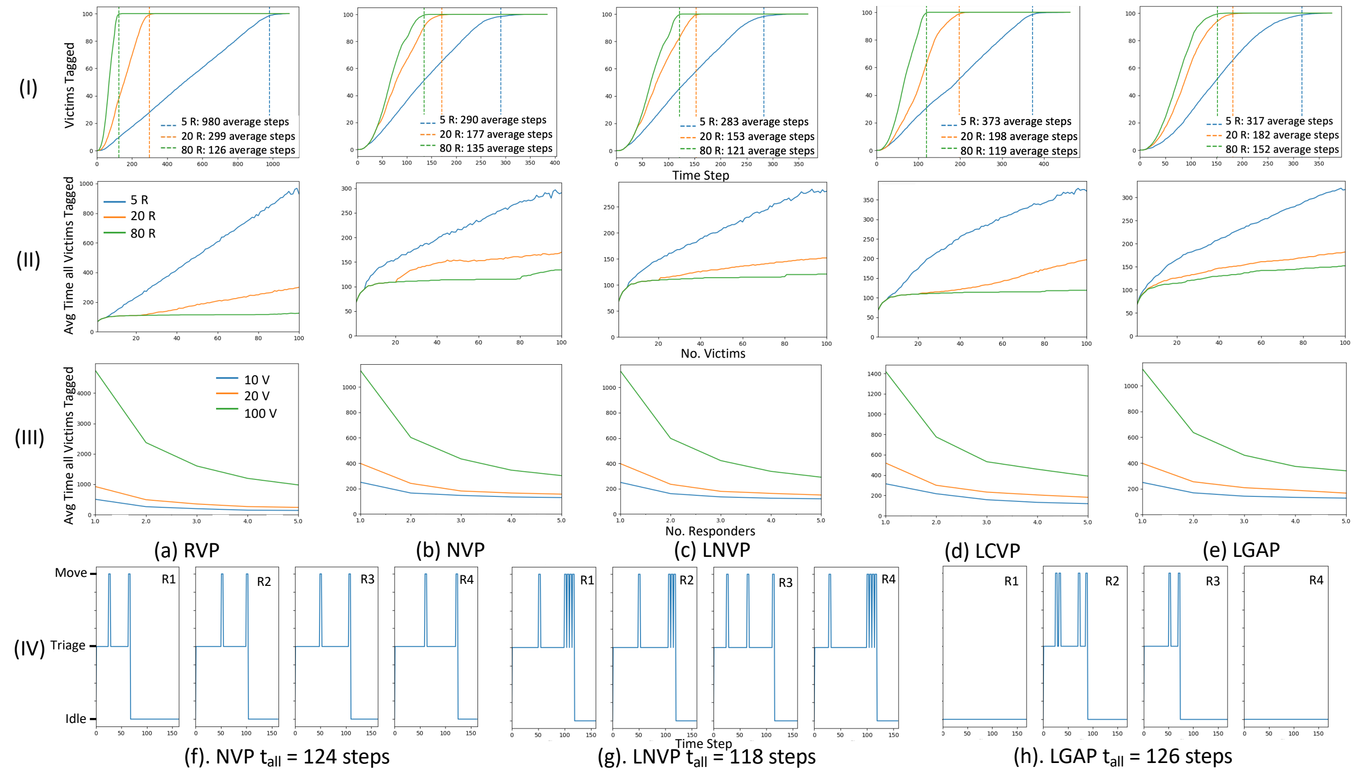

Fig. 4: (I) Number of victims tagged over time for 5, 20, and 80 responders. (II) Average time all victims are tagged for up to 100 victims for 5, 20, and 80 responders. (III) Average time all victims are tagged for 10, 20, and 100 victims for up to 5 responders. Columns (a-e) illustrate different policies. (IV) Responder agents' states over time. 4 responders are shown for experiment 1 (5R, 10V). Parts (f-h) demonstrate different policies and the time when all victims are tagged (t_{all}).

for any of the policies, excluding RVP, based on the graphs. However, if there are more victims, or a tighter deadline to accomplish, more responders would be needed to fulfill the time requirement.

Row (IV) in Fig. 4 depicts a responder agent r_i's state s_i for each time step in the case of 5 responders tagging 10 victims for NVP, LNVP, and LGAP (Fig. 2). The NVP graph indicates that each responder was in $s_i =$ triage for 2 steps, illustrating each responder tagged 2 victims. On the other hand, the local policies, LNVP and LGAP, had varying number of triage transitions, demonstrating the adaptation that occurred between responders as they communicated locally and identifies victims to tag. The adaptation resulted in improved efficiency for tagging victims quickly as LNVP and LGAP resulted in 118 and 126 time steps to tag all victims, respectively. Results for LGAP further highlight the inconsistency in efficiency due to the locations of victims. Responder 1 and 4 remained in the idle state the entire time, while responders 2 and 5 tagged all of the victims.

VI. Discussion and Conclusion

This paper aims to solve the multi-agent victim tagging problem of minimizing the time it takes to tag all victims during an MCI. We formalized the victim tagging problem using ILP, and proposed five applicable, distributed, on-the-go heuristics considering local and global communication constraints. Our solutions were evaluated through a series of simulation experiments. The results demonstrated that local policies performed most efficiently for adaptively identifying the next victim to tag for each responder, in an on-the-go fashion. Specifically, LNVP consistently performed most efficiently and RVP performed least efficiently.

Individual policy analyses provided further insights. For policies assuming global communication, NVP performed significantly better than RVP. The employment of local adaptive tagging proved valuable for selecting the next victim to tag. LCVP performed well but not the most efficiently. If the goal was instead to prioritize critical victims with a performance metric involving lives saved, LCVP should be explored as a potential optimal policy. LGAP performed poorly in comparison to the rest of the policies, as its performance was dependent on the locations of victims. An extension of LGAP where the responder cells consider victim population locations could be an improvement to consider in future work.

Future work could improve realistic global policies so that the responder's choice of a next victim is more efficient than finding the nearest victim on-the-go. Also, further analysis on local policies should be conducted to identify more optimal policies. Instead of finding the next victim to tag in an iterative and adaptive fashion, work could be done to find methods that could be more efficient, and still practical.

Our results could be utilized to create guidelines for the victim tagging procedure. Currently, each emergency department has individualized methods for responding to MCIs, without a focus on victim tagging algorithms. If medical responders have an estimate of the number of victims involved in an MCI, they can compare our policy results in choosing an optimal victim tagging method for their situation, as well as deciding on the number of responders needed to dispatch in order to tag victims within a timeframe.

REFERENCES

[1] Renee L. DeNolf and Chadi I. Kahwaji. Statpearls, Jan 2023. A service of the National Library of Medicine, National Institutes of Health.

[2] Glenn I. Hawe, Graham Coates, Duncan T. Wilson, and Roger S. Crouch. Agent-based simulation of emergency response to plan the allocation of resources for a hypothetical two-site major incident. In *Innovative Artificial Intelligence Solutions for Crisis Management*, volume 46, pages 336–345, 2015.

[3] Rafael A. Gonzalez. *A Framework For ICT-Supported Coordination in Crisis Response*. PhD thesis, Delft University of Technology, Delft, Netherlands, 2010.

[4] Rafael A. Gonzalez. Analysis and design of a multi-agent system for simulating a crisis response organization. In *Enterprises & Organizational Modeling and Simulation*, number 6, pages 1–15, 2009.

[5] Yu Wang, Louis K. Luangkesorn, and Larry Shuman. Modeling emergency medical response to a mass casualty incident using agent-based simulation. *Socio-Economic Planning Sciences*, 46(4):281–290, 2012.

[6] Narjès Bellamine Ben Saoud, Tarek Mena, Julie Dugdale, Bernard Pavard, and Mohamed Ben Ahmed. Assessing large scale emergency rescue plans: an agent based approach. *Int. J. Intell. Control Syst*, 11, 12 2006.

[7] Joanna Turner, Qinggang Meng, and Gerald Schaefer. Increasing allocated tasks with a time minimization algorithm for a search and rescue scenario. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3401–3407, 2015.

[8] Na Geng, Qinggang Meng, Dunwei Gong, and Paul W. H. Chung. How good are distributed allocation algorithms for solving urban search and rescue problems? a comparative study with centralized algorithms. *IEEE Transactions on Automation Science and Engineering*, 16(1):478–485, 2019.

[9] Wanqing Zhao, Qinggang Meng, and Paul W. H. Chung. A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario. *IEEE Transactions on Cybernetics*, 46(4):902–915, 2016.

[10] Jieke Shi, Zhou Yang, and Junwu Zhu. An auction-based rescue task allocation approach for heterogeneous multi-robot system. *Multimedia Tools Appl.*, 79(21–22):14529–14538, Jun 2020.

[11] Jinchao Chen, Chenglie Du, Xu Lu, and Keke Chen. Multi-region coverage path planning for heterogeneous unmanned aerial vehicles systems. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, pages 356–3565, 2019.

[12] Chen Luo, Qinwei Huang, Fanxin Kong, Simon Khan, and Qinru Qiu. Applying machine learning in designing distributed auction for multi-agent task allocation with budget constraints. In *2021 20th International Conference on Advanced Robotics (ICAR)*, pages 356–363, 2021.

[13] Lauren Bramblett, Rahul Peddi, and Nicola Bezzo. Coordinated multi-agent exploration, rendezvous, & task allocation in unknown environments with limited connectivity. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12706–12712, 2022.

[14] Martin Pallin, Jayedur Rashid, and Petter Ögren. A decentralized asynchronous collaborative genetic algorithm for heterogeneous multi-agent search and rescue problems. In *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 1–8, 2021.

[15] Gábor Pataki. Teaching integer programming formulations using the traveling salesman problem. *SIAM Rev.*, 45:116–123, 2003.

[16] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *J. ACM*, 7(4):326–329, oct 1960.

[17] Los Angeles Fire Department. Start - simple triage and rapid treatment, 2005. Accessed: 2024-02-28.

[18] Jackie Kazil, David Masad, and Andrew Crooks. Utilizing python for agent-based modeling: The mesa framework. In Robert Thomson, Halil Bisgin, Christopher Dancy, Ayaz Hyder, and Muhammad Hussain, editors, *Social, Cultural, and Behavioral Modeling*, pages 308–317, Cham, 2020. Springer International Publishing.

[19] Rappahannock EMS Council. Regional mass casualty incident plan. Accessed: 2024-02-28.