

# A Multi-Robot Task Assignment Framework for Search and Rescue with Heterogeneous Teams

Hamid Osooli<sup>1</sup>, Paul Robinette<sup>2</sup>, Kshitij Jerath<sup>3</sup>, and S. Reza Ahmadzadeh<sup>1</sup>

**Abstract**—In post-disaster scenarios, efficient search and rescue operations involve collaborative efforts between robots and humans. Existing planning approaches focus on specific aspects but overlook crucial elements like information gathering, task assignment, and planning. Furthermore, previous methods considering robot capabilities and victim requirements suffer from time complexity due to repetitive planning steps. To overcome these challenges, we introduce a comprehensive framework—the Multi-Stage Multi-Robot Task Assignment. This framework integrates scouting, task assignment, and path-planning stages, optimizing task allocation based on robot capabilities, victim requirements, and past robot performance. Our iterative approach ensures objective fulfillment within problem constraints. Evaluation across four maps, comparing with a state-of-the-art baseline, demonstrates our algorithm’s superiority with a remarkable 97 percent performance increase. Our code is open-sourced to enable result replication.

## I. INTRODUCTION

Advancements in robotics enable their assistance in dangerous human-avoidant search and rescue tasks [1]. Such operations often demand coordination of multiple robots and humans, but this poses various challenges [2], [3]. These challenges encompass diverse tasks in varied environments with constraints [2], [4], [5], underscoring the importance of effective multi-robot task assignment frameworks [6]. Existing frameworks lack comprehensive inclusion of information gathering, assignment, and planning [7], [8], and optimizing them becomes more complex with increased robot diversity [9].

To address these issues, we propose a scouting-integrated framework for search and rescue encompassing Multi-Stage Multi-Robot Task Assignment (MSMRTA) and path planning. Our framework optimizes assignment of diverse-capability robots to specific tasks. Task assignment employs optimization and market-based techniques [10], enhancing practicality. Empirical evaluation across various map complexities demonstrates significant planning time reduction (97%) compared to a baseline (MRGA) [2], with a slight (61.2%) increase as robot-victim instances grow. This paper uses off-the-shelf methods for the scouting and path-planning stages of the framework, but our proposed framework can be extended to incorporate other search and planning algorithms as well.

<sup>1</sup> PeARL Lab, Richard Miner School of Computer and Information Sciences, University of Massachusetts Lowell, MA, USA {hamid.osooli, reza.ahmadzadeh}@uml.edu

<sup>2</sup> Department of Electrical and Computer Engineering, University of Massachusetts Lowell, MA, USA paul\_robinette@uml.edu

<sup>3</sup> Department of Mechanical Engineering, University of Massachusetts Lowell, MA, USA kshitij.jerath@uml.edu

## II. RELATED WORK

As a humanitarian research domain [11], the development of robot technology for search and rescue operations continues to be an important area of study. To enhance the efficiency of multi-robot search and rescue, Quattrini Li et al. [12] introduced semantic cues, while Liu et al. [13] employed hierarchical reinforcement learning. To further improve the overall performance of the system, we integrate semantics related to the capabilities and requirements of robots via binary vectors.

Multi-Robot Task Assignment (MRTA) is vital for efficient search and rescue missions, categorized as Centralized methods [14], Decentralized methods [15], and Hybrid methods [5], [16]. For instance, Zhao et al. [14] employed centralized methods to locate survivors using heterogeneous robots, while our approach utilizes a team of scouts for accelerated search.

Decentralized methods, exemplified by Mouradian et al. [15], utilize multi-objective optimization with inter-robot communication. In contrast, our framework employs a communication-independent ant colony algorithm for efficient scouting.

Hybrid methods, like Liu et al. [5], combine autonomy and human intervention, while a hybrid framework models tasks as multiple traveling salesman problems [16]. Our unique approach employs the  $A^*$  algorithm independently for diverse calculations.

Further classification of MRTA involves optimization-based [14], [15] and market-based [16] approaches. Both have strengths and weaknesses [10]. In our paper, we propose the multi-stage multi-robot task assignment (MSMRTA) algorithm, integrating both techniques for enhanced effectiveness.

## III. PROBLEM DESCRIPTION

Prior to allocating victim requirements to the rescue team, essential information including location and requirements must be gathered through a scouting session using a set of uniform robots. These robots provide initial insights into the rescue operation’s aspects such as victim count, locations, and needs. The main problem addressed in this study is the task allocation for a team of heterogeneous robots in a search and rescue mission in a building. The proposed solution is applicable to both homogeneous and heterogeneous robot groups, accommodating missions with varying demands. The introduced Multi-Stage Multi-Robot Task Assignment (MSMRTA) framework combines optimization-driven and

market-based planning phases. A key aspect is the requirement analysis outlined in Algorithm 2, within which the MSMRTA, including the  $A^*$  path-planning algorithm for navigating building obstacles, allocates victim requirements to suitably capable robots. The framework’s central process is illustrated in Algorithm 1, and assignment outcomes are stored in designated variables ( $B2, B3, B4$ ). Reproducibility of results is facilitated through the open-source availability of the implementation code<sup>1</sup>.

---

**Algorithm 1:** Search and Rescue framework including Scouting, Multi Robot Task Assignment Analysis, Optimization, and Bidding stages, and Path-Planning

---

```

1 Input: map,  $R$  (List of the Robots)
2 Output: Prioritized Path to each Victim
3  $V \leftarrow \text{Scouting}(\text{map})$  // Victims’ information
4  $L^{\text{full}}, L^{\text{partial}}, L^{\text{potential}} \leftarrow \text{ReqmentAnalysis}(R, V)$ 
5  $L^{\text{unavailable}} \leftarrow \text{MissingCap}(V, L^{\text{potential}})$ 
6  $C, C^{\text{center}} \leftarrow \text{Clustering}(K, V)$  // K-means
7  $B2, \bar{V} \leftarrow \text{ClstrAsgn}(R, V, L^{\text{full}}, L^{\text{partial}}, C, C^{\text{center}})$ 
8  $T^{\text{success}}, T^{\text{failure}} \leftarrow \text{PerfAnalysis}(R, \bar{V}, L^{\text{partial}})$ 
9  $B3, \bar{V} \leftarrow \text{VictimAssign}(R, V, \bar{V}, T^{\text{success}}, T^{\text{failure}})$ 
10  $B4 \leftarrow \text{RobotAssign}(R, V, \bar{V}, L^{\text{potential}})$ 
11 Path-Planning( $B2, B3, B4$ ) //  $A^*$ 

```

---

### A. Scouting

In post-disaster emergencies, promptly assessing the environment provides vital victim distribution data [17]. We employ a 2D gridworld simulation to model building layouts and attributes (Fig. 1).

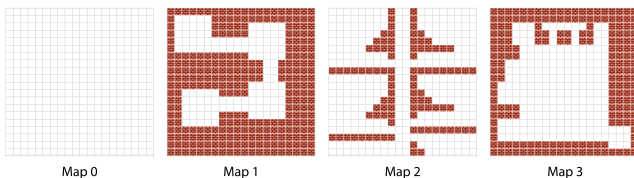


Fig. 1: Four  $20 \times 20$  maps used in our experiments. Obstacles have been depicted in red.

Our Scouting function (Algorithm1) uses the environment map to yield victim information—ID, location, requirements. Robots apply the ant colony grid search algorithm [18], mimicking ant search strategies and preventing redundant coverage via a search table. Scouts possess adaptable visual field depth (VFD) for victim detection, adjusting to obstacles. A scout identifies victims within its VFD vicinity (Fig. 2), extracting details. Agents navigate by moving forward, backward, right, or left in the grid.

### B. Task Assignment

Inspired by the work in [7], we address the task assignment problem in the context of a heterogeneous robot fleet.

<sup>1</sup>Github repository: <https://github.com/hamidosooli/Multi-Robot-Task-Assignment>

Throughout the paper, we use the terms ‘tasks’ and ‘victims’ interchangeably to refer to the same entity. The robots can either completely fulfill the victim’s requirements or fulfill part of the requirements, depending on their sensing and actuating capabilities. In line with the arguments presented in [10], we propose a hybrid multi-stage task assignment framework integrating optimization and market-based methods. The proposed framework follows the taxonomy in [19] and finds solutions for a group of single robot tasks (SR) through time-extended assignment (TA) of the tasks to multi task (MT) robots. To better describe our system, we have created a list of capabilities/requirements for both the robots and victims. For the robots, we represent these capabilities as a binary vector, indicating which capabilities each robot possesses. Similarly, we have created a binary vector for the victims to represent their individual requirements ( $\vec{q}$  in Table I and  $\vec{p}$  in Table II).

#### 1) Victim Requirement and Robot Capability Analysis:

The ReqmentAnalysis (Algorithm 2 and line 4 in Algorithm 1) focuses on determining three key sets of information through calculations. These sets include: victims that their requirements can be fully met by a given robot ( $L^{\text{full}}$ ), victims that their requirements can be partially fulfilled by a given robot ( $L^{\text{partial}}$ ), and potential robots that can meet a victim’s requirements ( $L^{\text{potential}}$ ). When a robot fulfills all

---

**Algorithm 2:** The ReqmentAnalysis function identifies victims fully satisfied ( $L^{\text{full}}$ ), partially satisfied ( $L^{\text{partial}}$ ) by a robot, and potential robots meeting a victim’s requirements ( $L^{\text{potential}}$ ).

---

```

1 Input:  $P, Q$ 
2 Output:  $L^{\text{full}}, L^{\text{partial}}, L^{\text{potential}}$ 
3  $U \leftarrow QP^{\top}$  // Matrix Multiplication
4  $S \leftarrow \sum_{j=0}^{nr-1} Q[:, j]$  // nr = number of requirements
5  $L^{\text{full}} \leftarrow (U > 0)$  and  $(U == S)$ 
6  $L^{\text{partial}} \leftarrow (U > 0)$  and  $(U < S)$ 
7 for  $i$  in  $(L^{\text{full}}$  and  $L^{\text{partial}})$  do
8 |  $L^{\text{potential}} \leftarrow Q[i(0), :]$  and  $Q[i(1), :]$ 
9 end

```

---

victim requirements, the victim ID is included in the set  $L^{\text{full}}$ . If a robot satisfies only a subset of victim requirements, the corresponding victim ID is added to the robot’s  $L^{\text{partial}}$  set. Conversely, robots capable of meeting a victim requirement contribute their IDs to the  $L^{\text{potential}}$  set for that requirement. By representing victim requirements ( $\vec{q}$ ) and robot capabilities ( $\vec{p}$ ) in matrix form, the system performs a matrix operation (Algorithm 2, line 3) to compare the outcome with a larger vector encompassing all requirements ( $S$ ) (Exemplified in Tables I and II). These binary vectors signify agent possession of each attribute. The algorithm’s inputs are matrices  $Q = [\vec{q}_0^{\top}, \dots, \vec{q}_{M-1}^{\top}]^{\top}$  and  $P = [\vec{p}_0, \dots, \vec{p}_{N-1}]$  containing information, presented as matrices to mitigate time complexity.

Using the data in  $L^{\text{potential}}$ , the MissingCap function (Algorithm 1, line 5) identifies instances where no suitable

robots exist to fulfill specific requirements. Subsequently, it appends the relevant victim IDs and requirements to  $L^{\text{unavailable}}$ , accounting for deficient capabilities.

2) *Clustering of Victims and Allocation of Victim Clusters to Robots*: In order to facilitate the task assignment process, victims are grouped into clusters based on their map locations using the `Clustering` function, employing the K-means algorithm [20]. These clusters, equal in number to the robots ( $K = N$ ), enable efficient robot-to-victim group assignments for proximate victims.

Robot-cluster matching employs a linear assignment approach [8], our assignment formulation (Equation 1) incorporates two assignment weights:  $w_{rc}^f$  (denoting victim requirement sets fully satisfied by robot  $r$  in cluster  $c$ ) and  $w_{rc}$  (indicating total victim requirements met by robot  $r$  in cluster  $c$ ) for  $N$  robots and  $K$  clusters.

$$\begin{aligned} \max_{\{x_{rc}\}} & \sum_{r=0}^{N-1} \sum_{c=0}^{K-1} [(1-\psi)w_{rc} + \psi w_{rc}^f] x_{rc} \\ \text{s.t.} & \sum_{r=0}^{N-1} x_{rc} \leq 1, \quad \forall c = 0, 1, \dots, K-1 \\ & x_{rc} \in \{0, 1\}, \quad \forall r, c \end{aligned} \quad (1)$$

The importance coefficient  $\psi \in [0, 1]$  balances fulfilling victim's requirements fully versus partially. A high  $\psi$  prioritizes robots in clusters satisfying one or more victims fully, while a low  $\psi$  favors clusters partially meeting requirements of more victims. Binary variable  $x_{rc}$  decides robot assignment to clusters. The equation's second line ensures a cluster is assigned to at most one robot [8].

Once robots join clusters, they move to cluster centers, calculated as victims' location averages. Manhattan distance from center to victims guides prioritization within clusters. The `ClstrAsgn` function (Algorithm 1 line 7) computes this, storing result in  $B2$ .

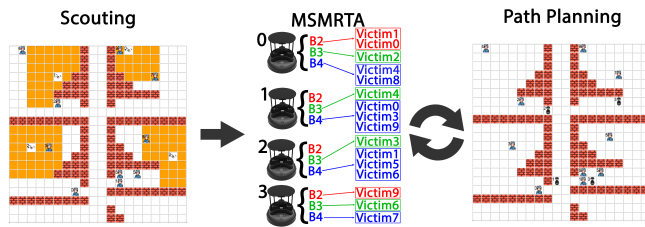


Fig. 2: Various steps of our proposed framework. The locations of victims and scouts (left). Each scout's Visual Field Depth (VFD) has been plotted in orange. The output of the different assignment stages (middle). The path planning step (right).

A clustering method that relies solely on location ([7], [21]) can be insufficient because it may fail to account for obstacles in the environment. For example, victims on opposite sides of a wall may be grouped together in the same cluster, leading to suboptimal robot travel times. To overcome this issue, we devised a method that examines each cluster for obstacles by comparing the Manhattan distance from the cluster center to each victim with that of an empty

environment. If the distance in the empty environment is greater than the distance in the environment with obstacles (walls), it suggests the existence of an obstacle, and the victim is excluded until the subsequent stage.

3) *Analysis of Robot Performance and Reliability*: Robots' performance in assigned clusters involves individual interactions with victims, either succeeding or failing. Evaluating reliability relies on analyzing time allocation for successful ( $T_{rv}^{\text{success}}$ ) and unsuccessful ( $T_{rv}^{\text{failure}}$ ) tasks. Robot reliability is computed to minimize both time categories, enhancing overall reliability. This optimization employs linear assignment to manage busy time for  $N$  robots and  $M$  victims using Equation 2. Binary variable  $x_{rv}$  ensures exclusive victim-to-robot assignment [8].

A coefficient  $\beta \in [0, 1]$  balances the time cost trade-off. Success and failure times are calculated using `PerfAnalysis` function (Algorithm 1 line 8). Subsequently, `VictimAssign` (Algorithm 1 line 9) employs this data to assign remaining victims to robots, stored as variable  $B3$ .

$$\begin{aligned} \min_{\{x_{rv}\}} & \sum_{v=0}^{M-1} \sum_{r=0}^{N-1} [(1-\beta)T_{rv}^{\text{success}} + \beta T_{rv}^{\text{failure}}] x_{rv} \\ \text{s.t.} & \sum_{r=0}^{N-1} x_{rv} \leq 1, \quad \forall v = 0, 1, \dots, M-1 \\ & x_{rv} \in \{0, 1\}, \quad \forall r, v \end{aligned} \quad (2)$$

4) *Bidding Based on Distance*: We employ the potential robot list ( $L^{\text{potential}}$ ) from the `ReqmentAnalysis` function to match unattended victim requirements with the nearest robot. The proximity is determined using Manhattan distance computation among potential robots. This approach employs a sealed bid scheme wherein robots submit concealed bids until auction culmination. The winner is the highest bidder, which in this context is the closest robot to the victim. The outcome, denoted as  $B4$ , is computed via the `RobotAssign` function (Algorithm 1, line 10), ensuring comprehensive fulfillment of victim requirements. Additionally, we verify  $L^{\text{full}}$  to prevent duplication. Specifically, if a robot is capable of fulfilling all of a victim's requirements, then that victim should not be assigned to any other robots.

### C. Path-Planning

The `Path-Planning` function in Algorithm 1 line 11 uses the  $A^*$  search algorithm, that is a widely used path-planning algorithm and particularly well suited for grid-based environments.  $A^*$  calculates the shortest path from the robot's initial location to the assigned victim's address by taking into account both the distance traveled and the estimated distance to the goal, also known as the heuristic cost. In our system, we use the Manhattan distance as the heuristic cost. The  $A^*$  algorithm ensures that the robot's path is efficient, while also avoiding obstacles and ensuring safe navigation. The calculated paths are then used by the robots to navigate to the assigned victims.

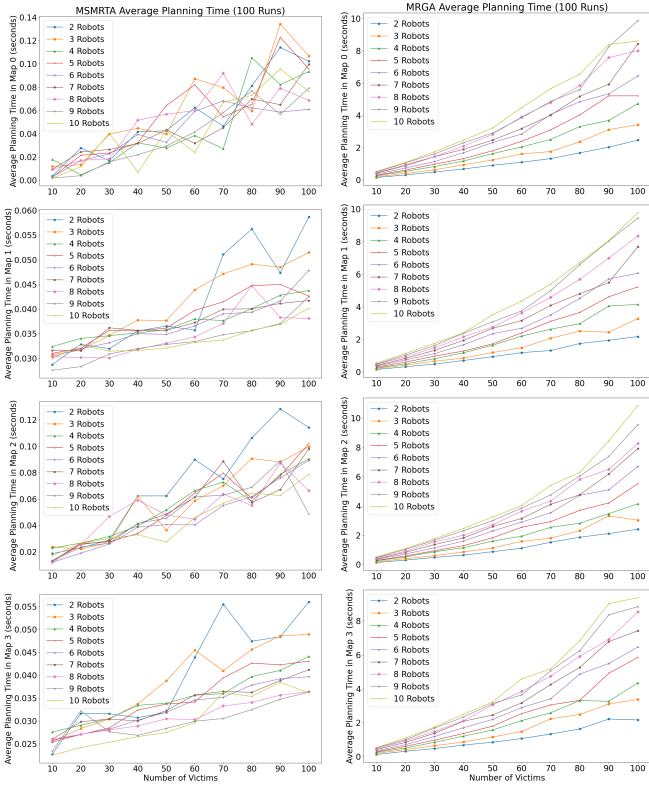


Fig. 3: Average planning time for the proposed algorithm (MSMRTA) on the left and the baseline algorithm (MRGA) on the right.

#### IV. RESULTS & DISCUSSION

In this section, we present the experimental results of our proposed framework on *Map 2*. Ten victims with diverse requirements (Table I) were assigned to four robots with varying capabilities (Table II). The Ant colony algorithm was employed by the scouting team for environment coverage before assignment. All ten victims were successfully located, and pertinent data such as ID, location, and requirements were gathered and recorded for task allocation (Table I). The Cluster ID in Table I indicates clustering outcomes (Algorithm 1, line 6), while  $L^{\text{potential}}$  denotes a requirement/capability analysis output (Algorithm 1, line 4). The remaining columns exhibit scouting team-acquired victim information.

TABLE I: Summary of victims’ locations, requirements, and potential robots for fulfillment of the requirements, as well as the corresponding cluster ID for each victim

Victim ID	Location	Cluster	$\vec{q}$	$L^{\text{potential}}$
0	(16, 7)	$C_0$	$[0, 1, 1, 0, 0, 0]^T$	$[\emptyset, [0], [1], \emptyset, \emptyset, \emptyset]$
1	(15, 12)	$C_0$	$[0, 0, 0, 1, 1, 0]^T$	$[\emptyset, \emptyset, \emptyset, [0, 3], [1, 2], \emptyset]$
2	(6, 5)	$C_2$	$[0, 1, 0, 0, 0, 1]^T$	$[\emptyset, [0], \emptyset, \emptyset, \emptyset, \emptyset]$
3	(11, 4)	$C_2$	$[0, 0, 1, 0, 1, 0]^T$	$[\emptyset, \emptyset, [1], \emptyset, [1, 2], \emptyset]$
4	(0, 1)	$C_1$	$[1, 1, 0, 0, 0, 0]^T$	$[[1, 3], [0], \emptyset, \emptyset, \emptyset, \emptyset]$
5	(14, 14)	$C_0$	$[0, 0, 0, 0, 1, 1]^T$	$[\emptyset, \emptyset, \emptyset, \emptyset, [1, 2], \emptyset]$
6	(14, 12)	$C_0$	$[1, 0, 0, 0, 1, 0]^T$	$[[1, 3], \emptyset, \emptyset, \emptyset, [1, 2], \emptyset]$
7	(3, 16)	$C_3$	$[1, 0, 0, 1, 0, 0]^T$	$[[1, 3], \emptyset, \emptyset, [0, 3], \emptyset, \emptyset]$
8	(10, 15)	$C_0$	$[0, 1, 0, 1, 0, 0]^T$	$[\emptyset, [0], \emptyset, [0, 3], \emptyset, \emptyset]$
9	(0, 12)	$C_3$	$[1, 0, 0, 0, 1, 0]^T$	$[[1, 3], \emptyset, \emptyset, \emptyset, [1, 2], \emptyset]$

This representation indicates that victim 0 has requirements that include the second and third items of the capability/requirement list, and robot 0 and robot 1 are potential candidates to fulfill these requirements, respectively. Additionally, victim 0 is located at position (16, 7), which belongs to cluster  $C_0$ .

TABLE II: Rescue Robots’ locations and capabilities. List of the victims IDs whose requirements can be fully or partially met by each robot, as well as the result of the victims IDs assigned to each robot in each stage of the task assignment process.

Robot ID	Location	$\vec{p}$	$L^{\text{full}}$	$L^{\text{partial}}$	Cluster
0	(0, 9)	$[0, 1, 0, 1, 0, 0]^T$	[8]	[0, 1, 2, 4, 7]	$C_0^{\text{center}}$
1	(0, 10)	$[1, 0, 1, 0, 1, 0]^T$	[3, 6, 9]	[0, 1, 4, 5, 7]	$\emptyset$
2	(19, 9)	$[0, 0, 0, 0, 1, 0]^T$	$\emptyset$	[1, 3, 5, 6, 9]	$\emptyset$
3	(19, 10)	$[1, 0, 0, 1, 0, 0]^T$	[7]	[1, 4, 6, 8, 9]	$C_3^{\text{center}}$

Table II shows robot 0’s capabilities and assignments. Victims 2 and 5 have unavailable requirements as indicated by  $L^{\text{unavailable}}$ . Fig.2 illustrates assignment stages with color-coded outcomes. Robot 2 lacks clusters in B2 but gains tasks in bidding. Initially, robot 0 is assigned to victim 8, later adjusted for spatial constraints. Redundancy in B4 is minimized, except for nearest qualified robot.

Our MSMRTA algorithm was evaluated against Carreno et al.’s MRGA on 2 to 10 robots, 10 to 100 victims, and 4 maps. MSMRTA reduces planning times significantly: 97.70%, 97.00%, 96.93%, and 97.34%. Planning time increased with more robots and victims: 92.26%, 31.80%, 80.85%, and 39.89%. Fig. 3 depicts these outcomes, highlighting MSMRTA’s potential in enhancing search and rescue efficiency.

#### V. CONCLUSION & FUTURE WORK

This paper introduces a novel search and rescue framework integrating scouting, multi-stage multi-robot task assignment, and path-planning. The framework leverages robot capabilities and victim requirements during scouting, optimizing task assignment based on this data and robots’ past performance. Notably, it accommodates multiple assignments by different robots to fulfill all requirements. The framework’s effectiveness was validated through simulations, exhibiting a 97% average reduction in planning time and overall performance enhancement. Future work entails refining clustering techniques, analyzing environment maps, exploring influencing factors, and validating the algorithm via real-world experiments with diverse robotic fleets.

#### VI. ACKNOWLEDGMENTS

This research is supported by the Army Research Lab contract W911NF20-2-0089.

#### REFERENCES

- [1] R. R. Murphy, “Rescue robots at the world trade center from sept. 11–21, 2001,” *IEEE Robotics Automat. Mag.*, vol. 13, pp. 11–21, 2001.
- [2] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, “Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision,” *Ieee Access*, vol. 8, pp. 191 617–191 643, 2020.

- [3] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," *Cooperative robots and sensor networks 2015*, pp. 31–51, 2015.
- [4] M. Rodriguez, A. Al-Kaff, A. Madridano, D. Martín, and A. de la Escalera, "Wilderness search and rescue with heterogeneous multi-robot systems," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 110–116.
- [5] Y. Liu and G. Nejat, "Multirobot cooperative learning for semiautonomous control in urban search and rescue applications," *Journal of Field Robotics*, vol. 33, no. 4, pp. 512–536, 2016.
- [6] S. Grayson, "Search & rescue using multi-robot systems," *School of Computer Science and Informatics, University College Dublin*, pp. 231–432, 2014.
- [7] Y. Carreno, È. Pairet, Y. Petillot, and R. P. Petrick, "Task allocation strategy for heterogeneous robot teams in offshore missions," in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, 2020, pp. 222–230.
- [8] L. Luo, N. Chakraborty, and K. Sycara, "Distributed algorithm design for multi-robot generalized task assignment problem," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 4765–4771.
- [9] —, "Distributed algorithms for multirobot task assignment with task deadline constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 876–888, 2015.
- [10] M. Badreldin, A. Hussein, and A. Khamis, "A comparative study between optimization and market-based approaches to multi-robot task allocation," *Advances in Artificial Intelligence (16877470)*, 2013.
- [11] J. L. Casper, M. Micire, and R. R. Murphy, "Issues in intelligent robots for search and rescue," in *Unmanned ground vehicle technology II*, vol. 4024. SPIE, 2000, pp. 292–302.
- [12] A. Quattrini Li, R. Cipolleschi, M. Giusto, and F. Amigoni, "A semantically-informed multirobot system for exploration of relevant areas in search and rescue settings," *Autonomous Robots*, vol. 40, no. 4, pp. 581–597, 2016.
- [13] M. Liu, K. Sivakumar, S. Omidshafiei, C. Amato, and J. P. How, "Learning for multi-robot cooperation in partially observable stochastic environments with macro-actions," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1853–1860.
- [14] W. Zhao, Q. Meng, and P. W. Chung, "A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario," *IEEE transactions on cybernetics*, vol. 46, no. 4, pp. 902–915, 2015.
- [15] C. Mouradian, J. Sahoo, R. H. Glietho, M. J. Morrow, and P. A. Polakos, "A coalition formation algorithm for multi-robot task allocation in large-scale natural disasters," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 1909–1914.
- [16] A. Hussein, M. Adel, M. Bakr, O. M. Shehata, and A. Khamis, "Multi-robot task allocation for search and rescue missions," in *Journal of Physics: Conference Series*, vol. 570, no. 5. IOP Publishing, 2014, p. 052006.
- [17] A. V. Nazarova and M. Zhai, "The application of multi-agent robotic systems for earthquake rescue," in *Robotics: Industry 4.0 Issues & New Intelligent Control Paradigms*. Springer, 2020, pp. 133–146.
- [18] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.
- [19] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International journal of robotics research*, vol. 23, no. 9, pp. 939–954, 2004.
- [20] J. MacQueen, "Some methods for classification and analysis of multivariate observation," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.
- [21] F. Janati, F. Abdollahi, S. S. Ghidary, M. Jannatifar, J. Baltes, and S. Sadeghnejad, "Multi-robot task allocation using clustering method," in *Robot Intelligence Technology and Applications 4: Results from the 4th International Conference on Robot Intelligence Technology and Applications*. Springer, 2017, pp. 233–247.