RoboCup Rescue 2023 TDP Agent Simulation MRL (Iran)

Mahdi Chatri[1], Mohamad H. Shayesteh[1]

[1]Islamic Azad University of Qazvin, Electrical, IT & Computer Sciences Department, Mechatronics Research Laboratory,
Qazvin, Iran

{chatrimahdi, m.h.shayesteh}@gmail.com

Abstract. This paper outlines our latest research and progress in the Agent Simulation Competition for the upcoming RoboCup 2023 event. In the previous year, we successfully implemented the k-means clustering algorithm to partition the map for the clustering task. We also optimized our strategies for ambulance, fire brigade, and police force agents. This year, we have further improved our clustering algorithm by eliminating the need for initializations and parameter selection. Moving forward, we aim to enhance our search strategies for ambulance task allocation.

Keywords RoboCup, Rescue Simulation, Map Clustering, Search Strategies.

1 Introduction

The Rescue Agent Competition involves a team of agents, including Ambulances, Police Forces, and Fire Brigades agents, working together to rescue civilians and extinguish fires in cities affected by an earthquake. The primary objective is to identify optimal strategies for multi-agent cooperation that can be applied across different scenarios. To achieve this goal, the agents must possess planning, learning, and information exchange capabilities to coordinate their efforts effectively and complete the rescue mission as a cohesive team [1].

The MRL Agent team has actively participated in RoboCup competitions, including the IranOpen, WorldCup, and Asia Pacific, since 2003. Our primary area of expertise is in developing multi-agent systems for rescue missions. Our team has achieved several victories in various world competitions from 2005 to 2021 in RoboCup competitions, cementing our position as a leading contender in the field [2].

Additionally, the system has been developed using a high-fidelity, open-source project called the Agent Development Framework (ADF), which is executed on the RoboCup Rescue Agent Simulator [3][4]. All teams must adhere to their code within the specified RCRS ADF classes.

This year, the Technical Committees (TCs) have introduced significant rule changes to (1) enhance the realism of simulation scenarios and (2) focus on a pertinent rescue challenge, specifically civilian rescue [5]. While teams previously implemented their code exclusively in the Java environment [4], they can now develop their agents using either Java or Python [6]. Additionally, Docker support has been made available, allowing teams to utilize a pre-configured Docker environment. To further increase the realism of simulation scenarios, fires have been removed, and the capacity of refuges has been limited to a fixed number of victims that can be accommodated on-site. These modifications demonstrate the organizers' dedication to ensuring the competition's evolution and relevance to real-world rescue operations.

In this study, we present our system development efforts concentrated on two primary aspects: map clustering and multi-objective search strategies for various types of agents in response to the recent rule changes. The newly introduced limitation on refuge capacity necessitates enhancing our search strategies for ambulance task allocation to identify optimal solutions. Furthermore, we propose a novel, parameter-free initiation clustering algorithm inspired by [7], compatible with large and small maps. The subsequent sections detail our investigations into clustering algorithms and search strategies.

2 Team Members

Our research laboratory comprises Ph.D., M.Sc., and B.Sc. students from diverse fields, including Artificial Intelligence, Software Engineering, and Information Technology Engineering. The Mechatronics Research Laboratory (MRL) also supports the Islamic Azad University of Qazvin. The following section introduces our esteemed leadership members:

- **Mohammad H. Shayesteh:** Map Clustering Algorithms
- **Mahdi Chatri:** Search Strategies Algorithms
- **Amir Hossein Moshfegh:** Base-Code Platform

3 Map Clustering

3.1 Related Works

Clustering is an unsupervised machine-learning technique that identifies and groups similar data points into natural clusters. Within each cluster, data points should exhibit minimal dissimilarity while maintaining maximum distance between different clusters. Clustering has various applications across various scientific domains, including image, video, text, web documents, and map segmentation [8]. Clustering approaches can be categorized into hard or soft clustering, point-assignment, or hierarchical models.

In hard clustering, each data point is wholly assigned to a single cluster, whereas in soft clustering, probabilities or likelihoods of data points belonging to different clusters are determined. Clustering algorithms can be classified as point-assignment models [9] or hierarchical models. The k-means algorithm is a widely-used point-assignment method with applications in color quantization, data compression, and image segmentation.

Alternative clustering models include distribution-based clusters that employ statistical distributions [10], density-based models such as DBSCAN and OPTICS [11], subspace models (also known as co-clustering or two-mode clustering) [12], and graph-based models [13]. In map clustering, image segmentation techniques are often applied to geo maps to classify regions within unknown environments. Related studies have utilized the k-means algorithm for segmentation tasks [14].

In previous years, we extended the k-means algorithm to guide the decision-making system for each agent. The similarity level in a standard two-dimensional space was assessed using Euclidean distance. Our strategy incorporated a combination of point density, building significance, and the probability of citizens' and agents' presence within each region. Subsequently, clusters were assigned to each agent using the Hungarian algorithm.

However, the current clustering approach requires tuning hyperparameters for different map sizes, which implies that the system's performance depends on the input maps' size.

3.2 Proposed Method for Map Clustering

This year, we investigate a novel clustering approach that does not require initialization or parameter tuning and applies to different map sizes. Based on [7], the proposed method can automatically determine the optimal number of clusters according to the number of agents and the available information in each competition schema setting for the map. Generally, clustering is an iterative process with certain conditions that aim to minimize an objective function (e.g., Euclidean Distance) for each cluster center and its associated data points:

$$a_k = \frac{\sum_{i=1}^{n} z_{ik} x_{ij}}{\sum_{i=1}^{n} z_{ik}} \text{ and } z_{ik} = \begin{cases} 1, if \ \|x_i - a_k\|^2 = \underset{1 \le k \le c}{min} \|x_i - a_k\|^2 \\ 0, otherwise \end{cases} \quad (1$$

Where $\|x_i - a_k\|$ represents the Euclidean distance between data points and the cluster a_k. In existing models, the number of clusters is typically unknown, and parameter initializations heavily influence the algorithm's performance. To address these issues, cluster validity has garnered increased attention for determining the optimal number of clusters, c. In the proposed method, a novel unsupervised k-means objective function is introduced by combining the z_{ik} and a_k values, utilizing the fundamental concept of cross-entropy theory as follows:

$$J_{UKM_2}(z, A, \alpha) = \sum_{i=1}^{n} \sum_{k=1}^{c} z_{ik} \|x_i - a_k\|^2 - \beta n \sum_{k=1}^{c} \alpha_k \ \ln \alpha_k - \gamma \sum_{i=1}^{n} \sum_{k=1}^{c} z_{ik} \ln \alpha_k \quad (2$$

In the case where β and γ are set to zero, we obtain the original k-means algorithm. To enhance the standard k-means algorithm, we compute the Lagrangian Equation 2 to derive new expressions for the z_{ik} and a_k parameters as follows:

$$z_{ik} = \begin{cases} 1, if \ \|x_i - a_k\|^2 - \gamma \ \ln \alpha_k = \underset{1 \le k \le c}{min} \|x_i - a_k\|^2 - \gamma \ \ln \alpha_k \\ 0, otherwise \end{cases} \quad (3$$

$$\alpha_k^{(t+1)} = \sum_{i=1}^{n} z_{ik} \,/\, n \ + \left(\frac{\beta}{\gamma}\right) \alpha_k^{(t)} - \sum_{s=1}^{c} \alpha_s^{(t)} \ln \alpha_s^{(t)} \quad (4$$

Where t represents the iteration number in the proposed method, and $\sum_{s=1}^{c} \alpha_s^{(t)}$ is the weighted mean of α_k with $\alpha_1, \dots \dots, \alpha_c$. To determine the optimal value for c, we carry out the clustering process iteratively to achieve a stable number of clusters for c by:

$$c^{(t+1)} = c^{(t)} - \left| \left\{ \alpha_k^{(t+1)} | \alpha_k^{(t+1)} < \frac{1}{n}, k = 1, \dots., c^{(t)} \right\} \right| \quad (5$$

Where |{}| denotes the cardinality of the set {}. Additionally, we must learn β and γ through a standard experiment. The proposed method is assessed using experimental data from previous maps of rescue agent competitions, and the algorithm's performance is demonstrated in terms of accuracy and computational cost measures in Section 4.

4 Search Strategies

4.1 Previous Works

As discussed in Section 1, rescue agents are categorized into the ambulance, police force, and fire brigade teams. In previous years, we assigned each ambulance agent to a specific partition, where they would initiate search and rescue operations. However, this approach occasionally led to ambulances traveling long distances to reach their designated partitions. To address this issue, we introduced a new strategy called "Sticky Move," based on potential field motion planning, which aims to overcome problems associated with lengthy paths. Furthermore, if agents detected a civilian within a certain proximity threshold along their route, they would attempt to rescue that civilian before proceeding to their target partition.

We also employed a Monte Carlo-based algorithm to estimate the remaining time before a civilian's demise and calculate the optimal number of ambulances required for rescue. By allocating only the necessary number of ambulances to rescue tasks, we enabled the remaining units to continue toward their designated partitions.

For police force agents, we introduced a straightforward method to identify blockades as potential traps for agents. Previously, we employed a guideline-based strategy to address the police force challenge of clearing blocked roads along the guidelines, leaving no residues in the 2019 RoboCup Competition. Furthermore, we enhanced the prioritization of police clearing tasks based on 1) traffic, 2) the importance of the buildings, and 3) the significance of the partition.

Additionally, we developed a modified version of the Artificial Bee Colony (ABC) optimization algorithm for fire brigade agents to search for fires on the map. Our proposed approach structured the system into distinct exploration and exploitation phases. The employed bees were dedicated to exploration, while the onlooker bees focused on exploitation, as illustrated in the following equation:

$$v_{i,j} = a_1 \cdot x_{i,j} + a_2 \cdot Gbest + a_3 \cdot \left(x_{k,j} - x_{i,j}\right) \qquad (6$$

Where X_i represents the current solution, Gbest refers to the global best solution found thus far, and X_k is a randomly selected solution ($i \neq k$). The weights a_1, a_2 and a_3 are associated with three random values, and $a_1 + a_2 + a_3 = 1$, where the a parameters are generated randomly.

Our procedure was implemented in five steps:

1) Generating N random solutions and computing their fitness values for FEs = N that FEs was the number of fitness evaluations

2) A new solution was generated V_i according to Equation 6, for each X_i

3) The probability p_i was calculated by: $p_i = \frac{fit_i}{\sum_{i=1}^{N} fit_i}$

4) All solution X_i was updated by the equation $x_{i,j} = low_j + rand_j \cdot \left(up_j - low_j\right)$ If is $max\{trial_i\} > limit$ where $\left[low_j, up_j\right]$ is the constraint box, and $rand_j$ is a random value generated in the range [0,1].

5) The procedure was stopped if $MaxFEs$ were the maximum value of FEs.

We employed this algorithm to assign tasks to fire brigade agents like the ABC algorithm in the task allocation process. Our approach was based on three types of bees: onlookers, employed, and scouts. Onlookers were responsible for deploying both scout and employed agents.

This year, the TC members decided to remove fires from the scenarios, emphasizing the role of ambulance teams. As a result, ambulances must now determine where to transport victims in their new mission rather than merely taking them to the nearest refuge. Furthermore, the capacity of refuges is limited in the new version, as outlined in the following items:

- **Bed Capacity:** Total number of available beds in the refuge
- **Occupied Beds**: Number of occupied beds in the refuge
- **Waiting List Size:** Number of civilians already waiting in the queue for treatment in the refuge

Following the new rules for refugees, ambulances can unload a civilian at a refuge if the number of occupied beds exceeds the bed capacity. The "Damage" property of civilians will decrease at each step as long as they remain in a waiting queue based on the First-In-First-Out strategy. It should be noted that civilians' "Damage" property increases when they are left outside the refuge without assistance. Consequently, in the next section, we will introduce a new optimization algorithm to address this issue.

4.2 Proposed Search Method for Ambulance Task Allocation

In line with the new rules for ambulance agents, they must determine the destination for transporting victims rather than merely taking them to the nearest refuge. Consequently, they must unload a victim at the closest refuge with available beds or shorter waiting times. We frame this challenge as a multi-objective optimization problem in which the agent makes decisions based on multiple factors.

Multi-Objective Optimization (MOO), or multi-objective programming, vector optimization, multi-criteria optimization, multi-attribute optimization, or Pareto optimization, is an effective tool for decision-making in mathematical optimization problems that require optimization of one or more objective functions [15]. In recent years, MOO has been applied in various fields of science, including engineering, economics, and logistics [16]. MOO necessitates making optimal decisions while balancing trade-offs between two or more conflicting objectives.

In mathematical terms, the MOO problems can be formulated as follows:

$$\min\left(f_1(\vec{x}), f_2(\vec{x}), \dots \dots \dots, f_n(\vec{x})\right) \quad (7$$

When there are more than two objective functions, a set of X∈R represents the feasible set of decision vectors. In these cases, multiple solutions can optimize each objective simultaneously. Pareto optimality is a well-known approach for such situations, as it considers a solution Pareto optimal if no other solution dominates it.

This year, we devised a new decision-making algorithm for ambulance task allocation inspired by [17]. Our objective functions incorporate factors such as the distance to the nearest refuge, the capacity of the refuge, and the waiting queue time at the refuge, for each agent. This can be expressed mathematically as follows:

$$ORS = \arg\min\{\min[CR_r, FR_r, WR_r], min[CR_d, FR_d, WR_d]\}$$

Here, CR represents the distance to the closest refuge, FR represents the capacity of the refuge, and WR represents the waiting queue time at the refuge. The objective functions are to be minimized, and the total cost is the sum of all objective functions computed using equations 8 to 10.

$$CR = \sum_{i=1}^{n} \sum_{j=1}^{k} \left|r_j - a_i\right|^2 \quad (8$$

$$FR = \min_{1<j<k} r_j(nb - nc) \quad (9$$

$$FR = \min_{1<j<k} r_j(qwt) \quad (10$$

Where r_j is refuge j, a_i is agent i, nb is the total number of beds in r_j, nc is occupied several beds in r_j, and qwt is the queue waiting time for each r_j. The proposed method is based on NP-complete to search for optimal global compromise, and Pareto optimal is selected for the best solution.

5 Experimental Results

In this section, we evaluate the performance of the proposed methods in sections 3 and 4 using the overall accuracy score for the proposed clustering and map score measurement for our new system based on a multi-objective decision-making system.

The experiments were conducted on a clustered system with three nodes, where each PC had a Core i7-8700K processor and 32 GB of RAM resources. The overall accuracy of the clusters was calculated as follows:

$$accuracy = \frac{number\ of\ recognized\ clusters\ by\ proposed\ method}{number\ of\ truth\ clusteres\ in\ the\ map} \tag{11}$$

In addition to the overall accuracy score and the evaluation of the proposed methods, the system's performance was also tested on various maps from the RoboCup 2021 Rescue Simulation. The maps used in the experiments were Montreal, sf2, sydney2, berlin3, kobe3, sakae2, eindhoven3, and paris3.

The system's performance on each map was evaluated based on the ranking section of the rules document [5], which provides a final score for each team. The team's score is determined based on several criteria, including the number of victims rescued, the time to complete the task, and the amount of damage incurred.

In the following sections, we will discuss the results of our experiments on each map mentioned above and provide a comprehensive analysis of the system's performance.

5.1 Performance of Proposed Map Clustering against Standard k-means Algorithm

In this experiment, we compared the performance of our proposed clustering algorithm with the standard k-means algorithm using a static value for the parameter k on seven different maps used in our project. The accuracy of both methods was measured, and the results are shown in Table 1.

Our proposed clustering algorithm was based on a new unsupervised k-means objective function that automatically finds the optimal number of clusters according to the number of agents and feasible information on each map's competition schema setting. The algorithm also does not require parameter tuning and works on different sizes of maps. The results showed that our proposed method achieved an average accuracy score of 90% in clustering the regions, indicating its effectiveness. This suggests that the new unsupervised k-means objective function and the absence of parameter tuning enable the algorithm to perform well on different maps and accurately cluster the regions.

On the other hand, the standard k-means algorithm performed well in maps with small sizes, but the accuracy score reduced significantly in larger maps. However, it should be noted that the standard k-means algorithm can be tuned for larger maps, but this would reduce its performance on smaller maps.

In contrast, our proposed method is not sensitive to the maps' scale and provides a generalized model for each scenario. The algorithm can be used on different maps without specific parameter tuning, making it more efficient and practical for real-world applications. Overall, these results demonstrate the superiority of our proposed clustering algorithm over the standard k-means algorithm in the context of rescue agent competitions.

Table 1. Performance analysis of the proposed method against a standard K-Means algorithm

Map Name	Standard K-Means	Proposed Method
Montreal	71%	91%
SF2	91%	93%
Sydney2	72%	90%
Berlin3	70%	94%
Kobe3	92%	91%
Sakae2	89%	90%
Eindhoven3	90%	95%
Paris3	87%	91%

5.2 Performance of Proposed Search Method against Artificial Bee Colony Algorithm

In this experiment, we aimed to compare the effectiveness of our proposed multi-objective search method for ambulance agents with the optimized Artificial Bee Colony (ABC) algorithm used in our previous projects while considering a new rule that limits the capacity of refugees. Firstly, we implemented our previous search method for the ambulance search strategy and then applied the proposed method to find the optimal refuge for unloading victims.

Table 2. Performance analysis of the proposed method against the Artificial Bee Colony algorithm

Map Name	Artificial Bee Colony	Proposed Method
Montreal	37.00	40.85
SF2	21.00	23.68
Sydney2	04.85	07.14
Berlin3	17.81	34.60
Kobe3	18.15	34.05
Sakae2	54.04	58.45
Eindhoven3	43.36	45.90
Paris3	57.07	76.23

Table 2 presents the results of our experiments, demonstrating the improvement in the overall project score. We set the refuge capacity to random values, and the number of agents was consistent with the RoboCup 2021 setting. The table highlights that the proposed method outperformed the previous method, although it showed a weaker performance on larger maps. On the other hand, the ABC algorithm performed better in all scenarios. Hence, we concluded that conventional decision-making methods might not be suitable for scenarios with limited refuge capacity, and multi-objective methods, such as our proposed method, may offer better performance in such situations. Overall, this text describes the experiment and its results more clearly, outlining the approach and conclusions drawn from the findings.

6 Conclusion

This study aimed to improve the performance of the search strategy for finding optimal regions in small and large maps, with the incorporation of a new multi-objective decision-making system and a free-of-initialization map clustering module. The experimental results showed a significant improvement in the map clustering system for both small and large maps, and the scores were shifted through adjustments to the new search function. Future research will focus on improving the computation cost of the proposed map clustering system to enable its application to real-time tasks. Overall, this study presents a promising approach for enhancing the efficiency and effectiveness of search strategies for optimal region detection.

References

1. Agent Simulation – RoboCup Rescue Simulation. http://rescuesim.robocup.org/competitions/agent-simulation-competition/. Accessed 27 Feb 2022
2. Competitions – RoboCup Rescue Simulation. https://rescuesim.robocup.org/past-events/past-competitions/. Accessed 27 Feb 2022
3. roborescue/rcrs-server: RoboCup Rescue Simulator. https://github.com/roborescue/rcrs-server. Accessed 27 Feb 2022
4. roborescue/adf-sample-agent-java: A sample team using RCRS Agent Development Framework. https://github.com/roborescue/adf-sample-agent-java. Accessed 27 Feb 2022
5. Rescue R, Organizers S (2022) Agent Simulation Competition Rules
6. roborescue/rcrs-sample-agent-python. https://github.com/roborescue/rcrs-sample-agent-python. Accessed 27 Feb 2022
7. Sinaga KP, Yang MS (2020) Unsupervised K-means clustering algorithm. IEEE Access 8:80716–80727. https://doi.org/10.1109/ACCESS.2020.2988796
8. Ghosal A, Nandy A, Das AK, et al (2020) A Short Review on Different Clustering Techniques and Their Applications. Springer Singapore
9. McCall MR, Mehta T, Leathers CW, Foster DM (2002) Effect on the metabolism of apolipoprotein B in African green monkeys. Am J Clin Nutr 56:385–393. https://doi.org/10.1093/ajcn/56.2.385
10. Lu X, Lu H, Yuan J, Wang X (2020) An improved K-means distributed clustering algorithm based on spark parallel computing framework. In: Journal of Physics: Conference Series
11. Kriegel HP, Kröger P, Sander J, Zimek A (2011) Density-based clustering. Wiley Interdiscip Rev Data Min Knowl Discov 1:231–240. https://doi.org/10.1002/WIDM.30
12. ParsonsLance, HaqueEhtesham, LiuHuan (2004) Subspace clustering for high dimensional data. ACM SIGKDD Explor Newsl 6:90–105. https://doi.org/10.1145/1007730.1007731
13. The Constrained Laplacian Rank Algorithm for Graph-Based Clustering | Proceedings of the AAAI Conference on Artificial Intelligence. https://ojs.aaai.org/index.php/AAAI/article/view/10302. Accessed 27 Feb 2022
14. Dhanachandra N, Manglem K, Chanu YJ (2015) Image Segmentation Using K -means Clustering Algorithm and Subtractive Clustering Algorithm. Procedia Comput Sci 54:764–771. https://doi.org/10.1016/J.PROCS.2015.06.090
15. Amidpour M, Khoshgoftar Manesh MH (2021) Optimization of cogeneration and polygeneration systems. In: Cogeneration and Polygeneration Systems. Elsevier, pp 287–323
16. Ivanov SY, Ray AK (2019) Application of multi-objective optimization in the design and operation of industrial catalytic reactors and processes. Phys Sci Rev 1:. https://doi.org/10.1515/PSR-2015-0017/ASSET/GRAPHIC/J_PSR-2015-0017_FIG_006.JPG
17. Meng K, Lou P, Peng X, Prybutok V (2017) Multi-objective optimization decision-making of quality dependent product recovery for sustainability. Int J Prod Econ 188:72–85. https://doi.org/10.1016/j.ijpe.2017.03.017