RoboCup Rescue 2020 TDP Agent Simulation CSU_Yunlu (China)

Fu Jiang*, Guanyu Cai, Ranfei Chen, Bochun Yue, Yiji Gao, Yuqing Ge, Xuxing Chen, Jing Xiong, and Wuqian Lv

Central South University, China
Jiangfu0912@csu.edu.cn

Abstract. This paper describes CSU_Yunlu's main strategies of RoboCup Rescue Agent Simulation. An estimation function based on the standard K-means algorithm and integrating the improved ideas of ideas of K-means++ and ISODATA algorithms is leveraged to divided the burning houses into different clusters. The another function based on A* enables the agent entities to choose the optimal path. The partition-based communication method improves the efficiency of information sharing. The strategies for three agents (Police Force, Ambulance Team and Fire Brigade) improve the efficiency of the rescue operation.

1 Introduction

Team CSU_Yunlu [3] has been participating in the RoboCup rescue simulation competition since 2006. Moreover we have achieved good results in both RoboCup 2016, 2017 and 2019. We hope to optimize our strategy and improve the performance of our team based on the strategies we have in previous years.

The main contents of this article are listed as follows (Figure1): clustering, path planning, communication and specific strategies for the three agents (Police Force, Ambulance Team and Fire Brigade).

2 Modules

2.1 Clustering

Purpose

In the rescue environment in RoboCup Rescue Agent Simulation, every map consists of thousands of nodes. In order to help the agents enhance the efficiency of searching the whole map, we need a clustering module to divide the whole

* Corresponding author.

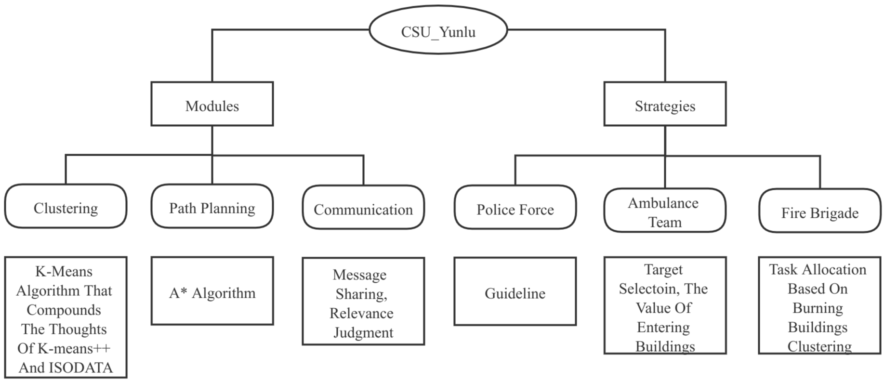

Fig. 1: Team Modules and Strategies

map into clusters. Each cluster can be seen as a smaller map, and the complexity of getting the information of each cluster is reduced greatly.

Also, using clustering algorithm to divide the ignition point of the entire map can help to extinguish the fire. It can prevent the fire from spreading in a large area and control the fire source within each divided block.

Related Works

Clustering algorithms can be roughly divided into four categories: partition-based clustering. Level-based, density-based, grid-based. In particular, Wang Zhenyu proposed the clustering algorithm based on Markov. But it needs a great number of data. K-Means is the typical partition-based algorithm, CURE is the hierarchical algorithm, DBSCAN is the density-based algorithm, and CLIQUE is the grid-based algorithm. Among them, K-means has the advantages of low complexity and fast computing speed, and is often used as the choice of many team clustering algorithms.

Proposed Approach

Our team simulated the cluster based on the standard K-means algorithm and integrated the improved ideas of K-means++ and ISODATA algorithms.

The main optimizations are in the following approaches:

1. Based on the K-means++ algorithm, the probability distribution is used to select the cluster center, and iterative modification of the ISODATA algorithm is used to obtain the best cluster groups, and the two-layer standard deviation is used to measure the effect of the cluster groups. Iteratively uses thresholds to simulate the best cluster groups.

Combine the idea that the distance between the cluster centers should be as far as possible. By first selecting the cluster center points, we use Euclidean distance to calculate the distance $D(x)$ from the cluster center to each point in the candidate point set.

$$D(i) = \sqrt{\delta x^2 + \delta y^2} \qquad sum = \sum_{i=1}^{n} D(i)$$

Calculate the distance of each point of the candidate point set, and the sum of these distances.Subtract the distance from each point by using the distance sum until the distance sum decreases to zero,then ans is the right point.

$$sum - \sum_{i=1}^{ans} D(i) < 0$$

2. Whenever we get a cluster group, we calculate the standard deviation of the distance between the cluster center of each block and each vertex as the evaluation value of the group, and then calculate the whole by the calculated evaluation value of each block The standard deviation of cluster groups, and set a metric to determine if the current fit meets the clustering criteria.

$$C(i) = \sqrt{\frac{\sum_{i=1}^{n} (D(i) - \frac{\sum_{j=1}^{n} D(j)}{n})^2}{n-1}} \qquad val = \sqrt{\frac{\sum_{i=1}^{n} (C(i) - \frac{\sum_{j=1}^{n} C(j)}{n})^2}{n-1}}$$

   (a) If it meets the criteria, it will jump out and use the cluster group as the criterion for subsequent processing at this time.
   (b) If it does not meet the criteria, delete the cluster center of the cluster with the largest standard deviation (that is, the largest degree of dispersion) in the cluster, then iterate again and reinstall until the installation criteria are met. At the same time, we set an upper limit on the number of iterations to prevent over-iteration.

Pros and Cons

By iteratively modifying the fit, we can obtain a better quality cluster, but at the same time, we need to pay more time for each iteration.

2.2 Path Planning

Purpose

The pathfinding strategy is to help the agent and the citizen find the optimal path. The purpose is to make their movements more efficient and enable them to reach their destination faster. The efficiency of mobility is the most important foundation for rescue operations. Improving the efficiency of mobility means increasing the efficiency of all rescue operations.

Related Works

A* [2] is a graph traversal and path search algorithm, which is often used in pathfinding due to its completeness, optimality, and optimal efficiency. In many cases, A* is the best solution. What's more, some other algorithms such as D* [5] and ARA [6] which based on A* can perform even better than A* in certain situations.

MRL team [8] from Iran adopted the A* algorithm in 2019. With respect to the existence of the blockades, if the navigation doesn't work properly, it can cause a huge time waste. They implemented new graph based on area passable edges and available blockades. Their A* algorithm considers these passable and blocked nodes to finding shortest reachable route, if available.

Proposed Approach

Compared with breath-first algorithms, A* algorithm has the following characteristics:

a) Not only record the cost from one point to source point, but also estimate the cost from this point to target point. It is a heuristic algorithm and can be regarded as a depth-first algorithm.

b) The selection of heuristic function is very important. If the estimated cost is larger than the actual cost sometimes, we can not ensure that the final path is the shortest. However, at one extreme, when the estimated is always zero, the A* algorithm is equivalent to the Dijkstra algorithm.

Specific methods:

Step 1. Define two lists named open and closed. Open List is used to store all the blocks considered to find the path, closed Lists are used to store blocks that are no longer considered.

Step 2. A is the starting point, B is the target point. Start from the source point A. Put point A into the open list and the closed list is initialized to be empty.

Step 3. Check the point n adjacent to a (n is called the child point of a, A is called the parent of n). Passable gird n can be added to the open list. Calculate their F (the estimated cost from the starting point via node n to the target point), G (the actual moving cost from starting point to point n), and H (estimated cost of moving from point n to target point) values. Remove A from open and add to the closed list.

Step 4. Judge whether the open list is empty. If yes, the search fails. If not, perform the next step.

Step 5. Remove n from the open list and add it to the closed list. Judge whether n is the target point B, if yes, it means that the search is Work and the algorithm operation ends, jump to the Step 8.

Step 6. If not, expand the search for n's child point:

a) If the child point is not passable or in the close list, ignore it.

b) If the child point is not in the open list, add it to the open list and set the current point as its parent, record the F, G and H values of the point.

Step 7. Jump to the step 4.

Step 8. cycle ends and save the path. Starting from the end point, each point moves along the parent node to the start point, which is the optimal path.

We decide to divide Road into four categories:

a) Passable: the condition of the road has been detected and can be passed by agents.

b) Impassable: the condition of the road has been detected and can not be passed by agents.

c) Unknown: undetected road

We choose the Manhattan distance to implement the Heuristic function H. Moreover, we choose different coefficients of H for these three kinds of roads above. 1 for passable roads, 1.2 for unknown roads and a large number for impassable roads to make sure agents hardly choose a path which may can not be passed.

When agents have a new target specified by center during the process of moving to the previous target, at this time, instead of immediately changing to the target that the center assign, we use Manhattan distance to compare the distance of the agent with the previous target and the new target assigned by center and then make a wiser choice.

Pros and Cons

Advantages:
a) It can get optimal solution theoretically.
b) It is easy to implement A* algorithm.
Disadvantages:
a) It is difficult to set optimal parameters.
b) It will search the entire map when the targets are isolated points.

2.3 Communication

Purpose

Communication is an important factor for disaster relief. The information that an agent can obtain is limited. Sharing the information with other agents can improve the efficiency of rescue task. The key to communication lies in how to properly deal with the communication within partitions and the communication among partitions.

Related Works

For the communication method, there are not many descriptions in the previous Team Description Paper (TDP). In fact, there are many implementation methods at present. The difficulty lies in how to ensure the transmission efficiency while transmitting information to as many agents as possible. In some studies, information is identified by the importance levels. Then the agent chooses to filter part of the information for reception based on the importance levels, which improves the efficiency of delivery.

Proposed Approach

The communication strategy is generally divided into communication within partitions and communication among partitions.

Communication Within Partitions: Agent will judge whether the message is related to itself or not when receiving a message. If it is related, the agent will give priority to the task in this message. If it is not relevant, this message will be broadcast to the nearby agents. Other agents also reacted after making similar judgments.

Communication Among Partitions: Different agents have totally different tasks. However, the messages they catch from environment are helpful for all kinds of agents. For example, the Police Force agent may find buried humans and buildings on fire. So They must immediately notice other kinds of agents once they get theses information to help control the fire and save humans.

Pros and Cons

Advantages:
a) Ensuring that the message is delivered in place.
b) Ensuring that the message is spread rapidly.
Disadvantages:
a) A large number of message senders will reduce the message processing efficiency.

3 Strategies

3.1 Police Force

Purpose

Police's clean behavior plays an important role. In order to help the other agent to carry out the work in time and promote the efficient operation of the whole simulation system, we must ensure that the police agent behavior of high efficiency. Its efficiency is shown as the following three aspects: correctness, effectiveness, smooth.

Related Works

Team SEU-Jolly [7] uses the grid division method to disperse the police in the early stage of police intelligence. At the same time, the distributed algorithm was used for task assignment and personnel scheduling.

Team MRL [1] used a guideline to help police force agents clear a road smother and prevent to create jagged blockades in a long straight path. What's more, it will greatly improves the efficiency of road cleaning in early stage of rescue simulation.

Proposed Approach

Last year, we developed the idea of using guideline to improve the efficiency of removing obstacles. But at last we didn't adopt the idea because it may be not conducive to ambulance team's work. However, in fact, using guideline method is of great help to control fire because it can help to clear a road for agents to pass rapidly. And fire fighting is far more important than saving humans at earlier stage.

Firstly, for task allocation, we divide task assignment into two categories: distributed task assignment in the absence of central agent and centralized task assignment based on central agent. According to these two categories, we propose a task assignment strategy based on task priority and an another one based on the Hungarian algorithm. Finally, for the clearing module, we use the Guideline model, which allows the police agent to move along the guideline and makes the walking trajectory as simple as possible.

Guideline Model
Determine the starting point of the bootstrap
Step1: obtain the road area where the agent is located, denoted as sourceArea;
Step2: get the connection boundary of the next region adjacent to the SourceArea and path, denoted as edgeTo.Get the middle of the boundary, call it middle;
Step3: make three straight lines: the connection line from the location of the agent to the boundary center point, the connection line from the location of the agent to the center of the road area where the agent is located, and the connection line from the road center to the boundary center, respectively denoted as agentEdgeLine, agentAreaLine and areaEdgeLine;
Step4: calculate the included Angle between agentAreaLine and areaEdgeLine, agentAreaLine and agentEdgeLine, respectively denoted as theta and alpha;
Step5: compare the size of θ and α, if $\alpha < 80$ and $\theta > 80$, the intelligent body position is the starting point of the guideline; Otherwise, the starting point is still the center of the road area where the agent is located.

The points, lines, angles and areas used in the above steps are shown in the figure below:

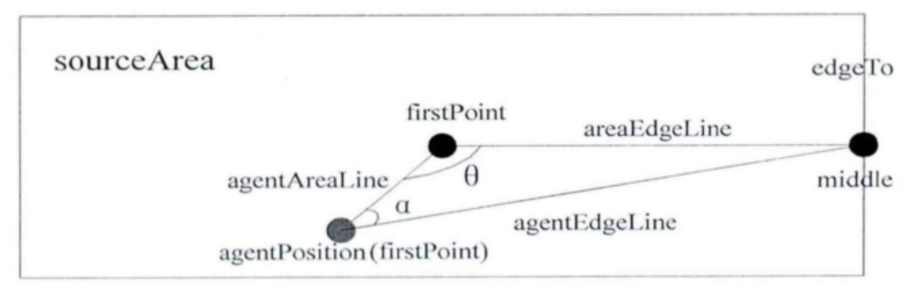

Fig. 2: The starting point of the path bootstrap

Plan the intermediate connection process of the path guideline
Step1: traverse from the first road area to the penultimate area of the entire path;
Step2: determine the starting point of the current region and the midpoint of the adjacent boundary between the current region and the next region;
Step3: establish the regional guidance line from the starting point to the middle point of the boundary, and add it to the list of path guidance lines;
Step4: take the middle point of the current boundary as the starting point of the next regional guideline;
Step5: repeat steps Step2 through Step5.

A schematic diagram of the above process is shown below:

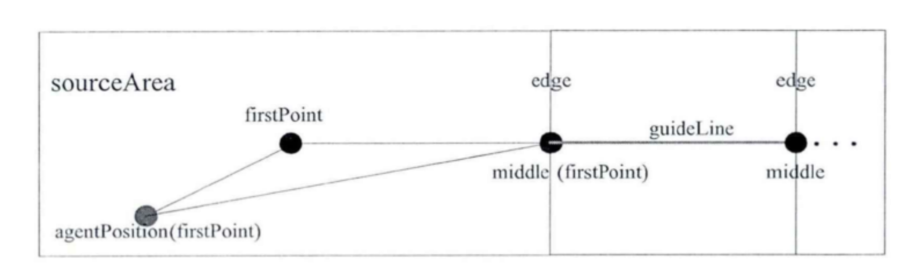

Fig. 3: The intermediate connection process of the path guideline

Determine the starting point of the bootstrap

All the intermediate and indirect processes of the path guidance line above start from the midpoints of two adjacent boundaries until the last area, that is, the road area where the target point is located.The path to be cleared by the police is planned to the target point, but not the area where the target point is located. At this point, the target point is used as the end point of the last area guideline and the area guideline is added to the list of path guidelines. Schematic diagram is shown as follows:

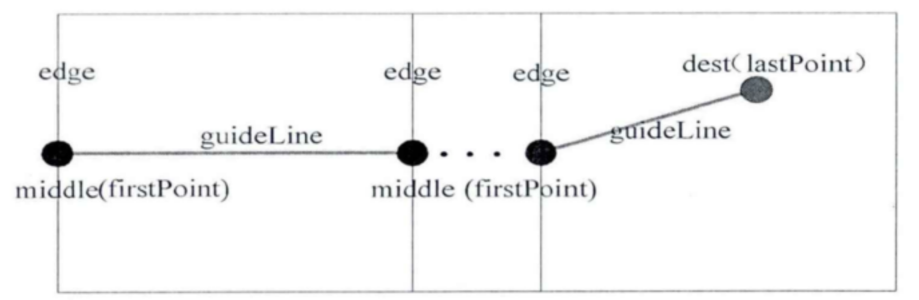

Fig. 4: Connect the Path Guidelines endpoint lastPoint

Experimental result

The figure below shows a comparison of the model with and without the guideline

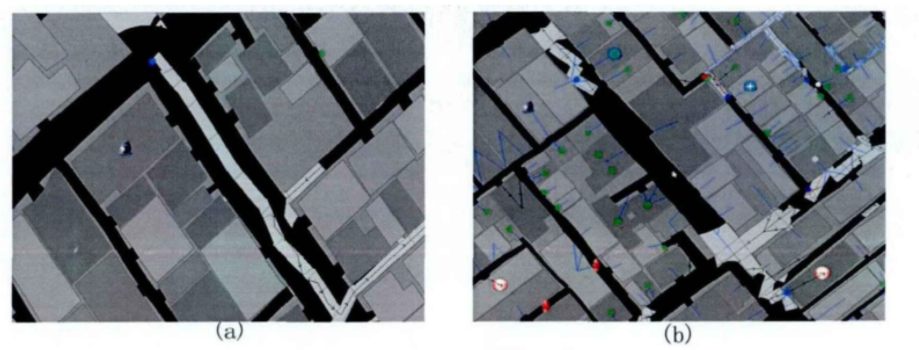

Fig. 5: Use the before and after guidelines

In figure (a), we used the guideline model. Obviously, compared with figure (b), figure (a) shows a smoother result of police obstruction clearance, with a significant decrease in the number of burrs and irregular patterns.

Pros and Cons

Advantages:
a) The agent is adaptive to the dynamic rescue environment.
b) The trajectory of the police agent can be optimized.
c)The clear action save plenty of time result in the Convenience of other agents' rescue action.

Disadvantages:
a) The quality of the communication mechanism seriously affects the results of task assignment.
b) The Guideline model requires a more rational design when dealing with intersections and map edges.

3.2 Ambulance Team

Purpose

The task of the ambulance team is to rescue, treat the trapped wounded and transport them to the refuge quickly and effectively. Based on the strategies in previous years, we plan to improve the efficiency of rescue.

Related Works

In order to implement such ambulance, team Aura [4] took full advantage of the information of travel time to the refuge, civilian death time and the fire simulator. They use the data above to simulate the process of rescuing the civilian and calculate the possibility of rescue.

Team MRL [8] use a sticky move method for ambulance team agents. In this way they can move to their clusters and in those they can rescue any civilian nearer than the threshold distance.

Proposed Approach

At the beginning of simulation, sometimes there are lots of ambulance teams, fire brigades and police forces trapped in a building. For a better rescue efficiency, we decide to first rescue ambulance teams, fire brigades and police forces instead of civilians. However, as time goes on, more and more civilians are in danger of death and this will greatly impact the results. So we will pay more attention to civilians with as time goes by.

we didn't perform well in human search last year. This year, we want to improve search efficiency by enhancing communication among agents and firstly searching the buildings where humans are calling for help. Also, when ambulance is in front of a building, we estimate the probability if ambulance can correctly judge whether there is anyone in it from the current location. If the probability above the threshold, ambulance doesn't need to enter the building.

Pros and Cons

Advantages:
a) It can guaranteed the quantity of agents that can work to improve efficiency.
b) It can speed up the search for houses.

Disadvantages:
a) It may lead to increasing in deaths of civilians at the early stage.
b)It may leading to the omission of some houses where humans in.

3.3 Fire Brigade

Purpose

The purpose of setting this strategy is to better control the fire and reduce casualties. Determine the edge building by constructing a convex hull of the point set, and then extinguish the fire from the outside to the inside to prevent the fire from spreading.

Related Works

The Concave Hull is also a tool that can be used to compute the envelope of a set of points in the plane, by generating convex or non-convex polygons that represent the area occupied by the given points.We also plan to incorporate this method to improve the accuracy and the fire suppression efficiency.

Proposed Approach

Our team optimizes in processing of fire source cluster and the dynamic processing when new ignition sources are added.

1. We use Graham-Scan algorithm to preprocess the set of fired buildings, and we overload the sorting algorithm by using $\cos\theta$(θ is the angle with polar edge) as standard to ensure that the cos of the buildings and poles that are on fire are dynamically added to the correct position of the set. Let us set the pole point as A_1, the other point on the other side of pole is A_2, and all the other points $B_1, B_2, ...B_n$, maintain the building set $B_1, B_2...B_n$ by the following method.
$$\cos(\overrightarrow{A_2B_{m-1}}, \overrightarrow{A_2A_1}) \leq \cos(\overrightarrow{A_2B_m}, \overrightarrow{A_2A_1})$$
2. To ensure that the newly added burning buildings are inside the convex hull, we use multiplication cross to calculate. For example, we assume two burning buildings B_1, B_2, then we judge the burning building C by the following formula.
$$(B_1.y - B_2.y) \cdot (B_1.x - C.x) - (B_1.x - B_2.x) \cdot (B_1.y - C.y) > value?B_1 : C$$
   (a) For the case where the building is exactly horizontal or vertical on a straight line, we need to make a special judgment and compare it by calculating the distance to prevent the far point from being selected as the next point on the convex hull.

(b) The extreme case occurs when the building is just a bit off on a straight line but the value of mutiplication cross point shows that they belong to different lines. To judge this situation, we set the value of the parameter *value* to 10^{-6} to optimize it. And then this situation turns into the above situation(a).

Pros and Cons

Advantages:
a) The fire is easy to control and the area affected is small.
Disadvantages:
a) The internal building has a high probability of overburning.

4 Preliminary Results

* means we have modified the map.

Team	Map		
	Kobe	**Sydney***	**Paris***
My Team	197.332	211.564	190.663
CSU_Yunlu_2019	194.582	190.467	114.805
Sample	145.661	193.347	91.005

5 Conclusions

Different agents have different responsibilities and behaviors, so the priorities of tasks are different. Path Planning and Clustering are the most basic and important modules, so finding a better way to achieve them is a constant topic. We should design different strategies for different agents for an efficient solution. At the same time, we can't ignore the cooperation between agents. Effective cooperation strategies can generate greater benefits. Communication is the foundation of cooperation, so we try to use a reasonable communication strategy to improve the score.

In the coming period, we plan to absorb the advantages of other team agents, and then improve our strategies. Also, we will try to introduce more advanced methods.

References

1. Ardestan, P., Taherian, M., MohammadAliZadeh, P., JazebNikoo, E.: Rescue Simulation League MRL Team Description (2017)

2. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE transactions on Systems Science and Cybernetics **4**(2), 100–107 (1968)
3. Jiang, F., Lv, W., Zhang, J., Peng, J., Fu, Y., Lou, Y., Zhang, X.: Rescue Simulation League Team Description CSU_Yunlu(China). RoboCup 2019 (2019)
4. Kandeh, A., Absalan, A.: Rescue Agent Simulation Aura. RoboCup 2018 (2018)
5. Likhachev, M., Ferguson, D.I., Gordon, G.J., Stentz, A., Thrun, S.: Anytime dynamic a*: An anytime, replanning algorithm. In: ICAPS. vol. 5, pp. 262–271 (2005)
6. Likhachev, M., Gordon, G.J., Thrun, S.: Ara*: Anytime a* with provable bounds on sub-optimality. In: Advances in neural information processing systems. pp. 767–774 (2004)
7. Niu, W., Wu, J.: Rescue Simulation League SEU_Jolly Team Description (2015)
8. Zarei, S., Chatri, M., Mirfattahi, S.: Rescue Agent Simulation MRL. RoboCup 2019 (2019)