# RoboCup Rescue Asia Pacific YEAR 2021
# TDP Agent Simulation
# Team Doodd (Singapore)

James Kusuma Dewa Halim, Tan Jun Yong, and Marc Choo

Singapore University of Technology and Design, Singapore
[james_halim@sutd.edu.sg, junyong_tan@mymail.sutd.edu.sg,
marc_choo@mymail.sutd.edu.sg]

**Abstract.** This paper will describe the main methods and strategies of Team Doodd in this year's RoboCup Rescue Asia Pacific Agent Simulation. The agents includes Graham Scan convex hull method and new constraints to help their path planning capabilities. The buildings are now divided into different clusters using the modified optimized k-means++ algorithm. The parameters optimization, improves existing algorithms and decreases time complexity.

## 1 Introduction

We focused on implementing 2 types of changes, parameter optimization on each agent and improving their existing algorithm to make it more efficient. The parameter optimization focuses on reallocating agent resources to focus on activities that maximises our score, which leaves injured civilians with low health to pass on earlier. The base code for our submission is based on CSU_Yunlu 2021 Robocup Agent TDP submission. The algorithms that we modified are as follows:

- Modifying the k-means++ clustering such that takes into account of possible empty clusters
- Modifying the guideline creator, that is used by the police force to clear obstacles, to ensure a straighter guideline being produced
- Modifying the map dimension determination algorithm, that is used by the ambulance team, to be based on average distance from the center of the map to all the building and road entities.
- A computationally less expensive Graham-Scan is used for the Convex Hull algorithm.

## 2 Modules

### 2.1 Clustering

The standard k-means++ clustering algorithm might be unable to form the number of cluster group desired. Thus overlapping of agent assignment to a cluster

might occur [1] reducing the overall efficiency of splitting the map effectively. To prevent this, we implemented an empty cluster strategy where we split a cluster with the highest variance into more clusters to ensure that the desired number of cluster group is always formed.

### 2.2 Convex Hull

Instead of using the Standard Jarvis March Convex Hull algorithm, it is proposed that the Graham 2D Scan method be used in its stead as it is computationally faster than the aforementioned algorithm. This is because Jarvis March method has a worst case scenario of $O(N^2)$ and an average time complexity of $O(NM)$, where M is the number of output or hull points, assuming $M <= N$.

However, in most cases, $O(logN) <= O(M)$ even if N is 2 orders of magnitude larger than M, an example would be $N = 1000$ and $M = 10, log(1000) < 10$. Hence, the proposed changes of Jarvis March method to Graham 2D Scan is chosen instead [2].

| | Average | Worst |
|---|---|---|
| Jarvis March | $O(NM)$ | $O(N^2)$ |
| Graham Scan | $O(NlogN)$ | $O(NlogN)$ |

Table 1: Time complexity table

## 3 Strategies

### 3.1 Police Force

The main changes in the police force were in the Guideline creator algorithm. The guideline is used by the police force to clear obstacles along the guide-line. Previously, CSU_Yunlu implemented a guideline creator based on forming a straight line in between 2 mid point of edges of a road as shown in Fig1.

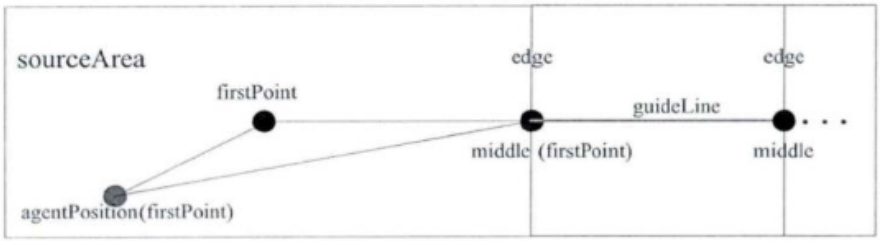



Fig. 1: CSU_Yunlu guide line creator

This might cause a zig-zag pattern, $d_1$ and $d_2$, as shown in Fig2. This could be avoided if the next midpoint is also considered as a potential guide-line, $d_3$ from $p_1$ to $p_3$. This way, the guide-line produced would be straighter.

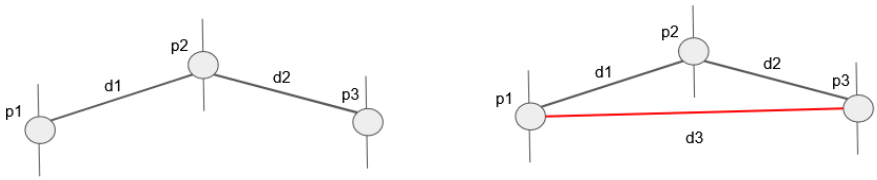



Fig. 2: Default vs Improved guide-line creator

We also implemented a constraint to only consider $d_3$ as the guideline if the condition at 1 is fulfilled. Otherwise it will only be the same as the default algorithm where $d_2$ will be considered the guideline. This ensure that the guideline is only created on a straight road and not when the agent is about to turn.

$$d_1 + d_2 > d_3 * 1.1 \tag{1}$$

Where $d_1$, $d_2$ and $d_3$ are distances from $p_1$ to $p_2$, $p_{p2}$ to $p_{[3}$ and $p_1$ to $p_3$ respectively. 1.1 is a constant we can adjust to sensitivity of the algorithm.

### 3.2 Ambulance Team

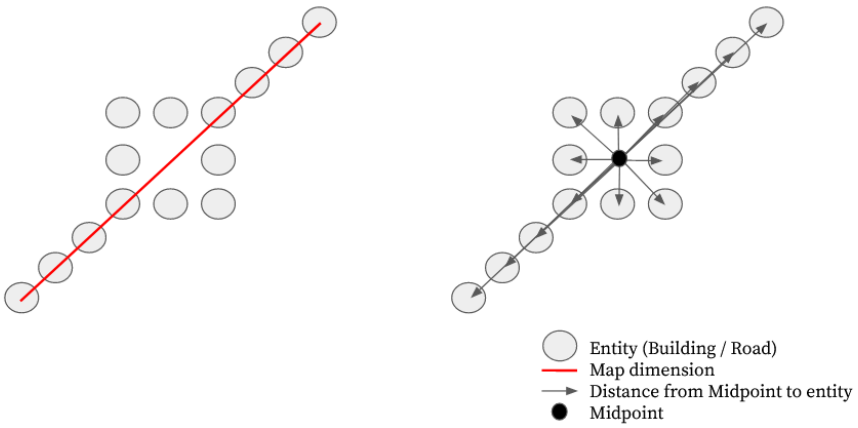
The range of rescue of the Ambulance Team is dependent on the size of the map. There are 3 sizes are used, small, medium and large. Previously, the size of the map is determined by iterating through all entities in the map and finding the $max_x$, $min_x$, $maxy$, $min_y$ values. After finding these values, the dimension of the map is determined by the hypotenuse where:

$$\begin{aligned} x_{dim} &= max_x - min_x \\ y_{dim} &= max_y - min_y \\ map_{dim} &= \sqrt{x_{dim}^2 + y_{dim}^2} \end{aligned} \tag{2}$$

While this approach is effective in finding the map size, it might be inaccurate in the case of entities which are clustered together closely with a few entities spread out such as in Fig**??** and thus possibly skewing the results. Thus we change the determination of the map size to the average euclidean distance from of all the entities to the mid of x and the mid of y to the rest of the entities. This way, entities which are found to be much further away than a cluster will not skew the determination of map size too much.

### 3.3 Fire Brigade

In the event that there are multiple injured civilians the Fire Brigade agent will prioritize and detect civilians with health above a certain threshold. This is to avoid the scenario where the severely injured civilians passed on during




their transport to the refuge by the ambulance agent. Hence, if the health of the civilian is beneath the threshold, we will mark them as dead since they would not reach the refuge in time.

## 4 Results

Due to technical constraints, we are unable to test our algorithm on a large map. However we created a smaller test map and tested our algorithm on it and results show minor improvements.

| Team name | Test Map |
|---|---|
| Original CSU_Yunlu | 33.8212 |
| TeamDoodd Modified | 34.8254 |

Table 2: Comparison of Scores

## References

1. M. Pakhira, "A Modified k-means Algorithm to Avoid Empty Clusters," *International Journal of Recent Trends in Engineering*, 2009.
2. T. Pereira, "Convex hull algorithms," 2008.