Ri-one

RoboCup Rescue Simulation League 2018

K-means Algorithm

- Using below algorithms for clustering
  - Before : K-means algorithm
  - After : K-means++ algorithm

K-means++ Algorithm

K-means demerits

- Improper points can be chosen
    - The solution can be worse.
    - K-means unsuited to clustering in a biased map.

K-means++ algorithm solved these problems.

- K-means++ is conscious of below points.
    - Initial centers were chosen spreadly.

K-means++ Algorithm

Initial centers are chosen by below steps.

1. Choose one center uniformly at random from all points.
2. The probability is defined by the following.

   $P(x_i) = D(x_i)^2 / \Sigma_k D(x_i)^2$

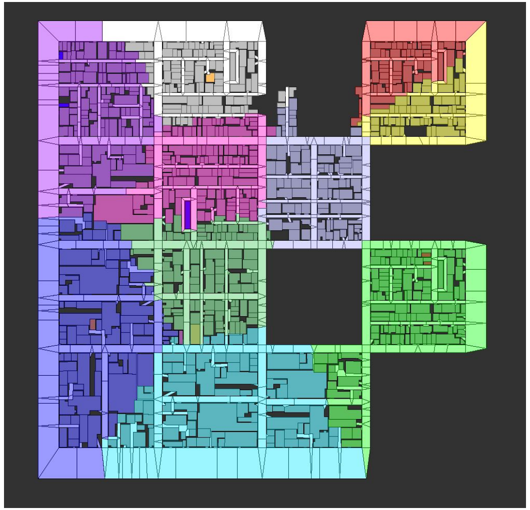
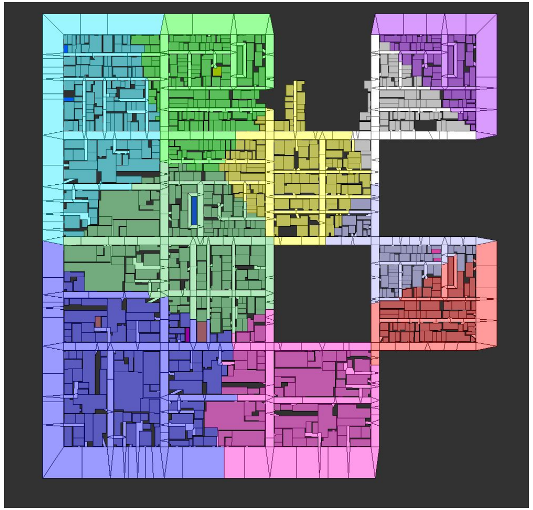
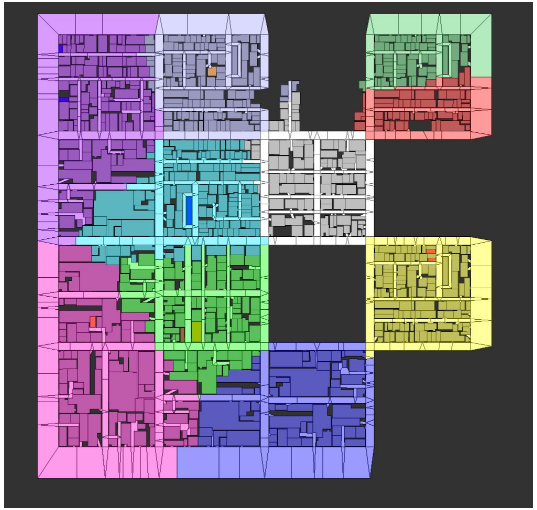
   $P(x_i)$: the probability x_i is chosen for initial center

   $D(x_i)$: the distance from the nearest center to x_i
3. New initial center, x_i is chosen by this probability.
4. Repeat those steps until 10 centers have been chosen.

As a result, the improper points were chosen only occasionally in a biased map.

K-means++ Algorithm

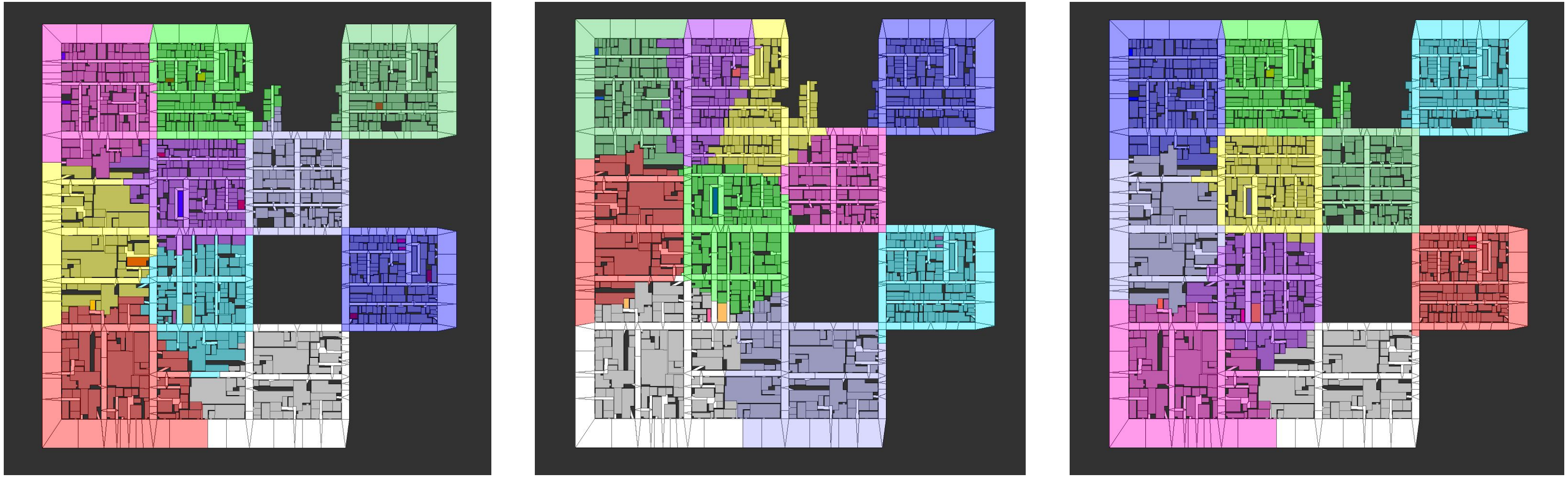
Results by K-means

Note : Map is modified map from VC1

K-means++ Algorithm

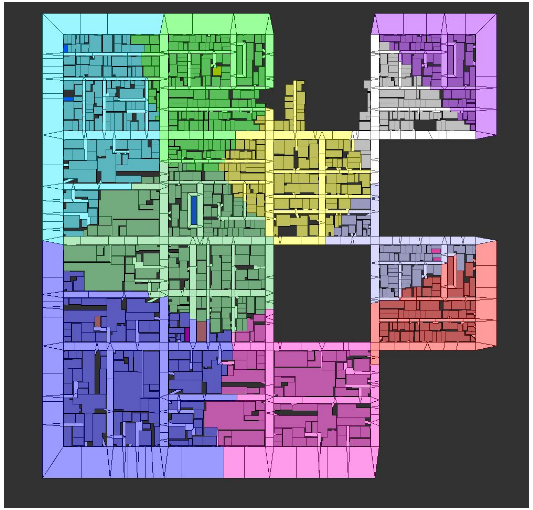
Results by K-means++



Note : Map is modified map from VC1

K-means++ Algorithm

K-means



K-means++

The clustering work adequately.

Problems of our team’s FB

- FB was limited
- Fire occurred at multiple locations
- The situation of the disaster was changing by every moment

→Considering allocation of FB according to the situation of the fire.

Modification of our method of allocation

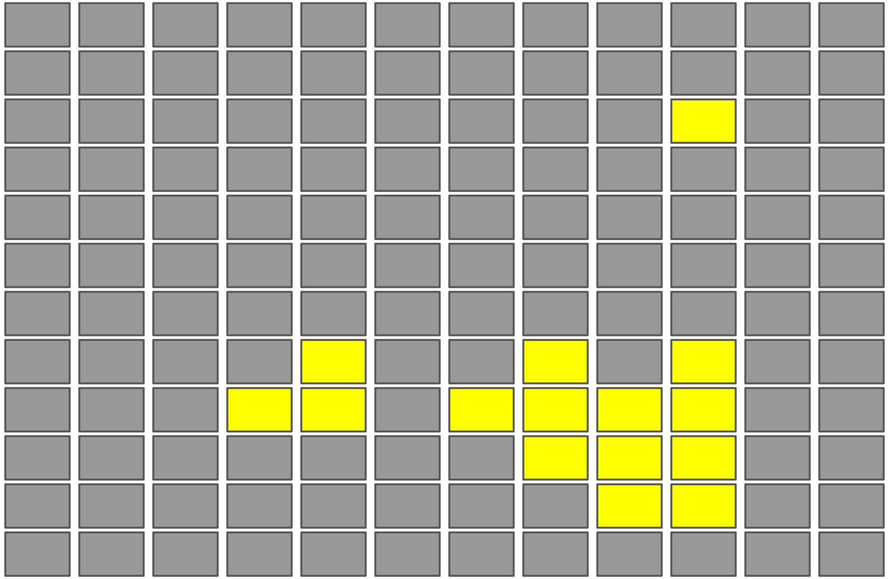
Our team’s method last year

- Static cluster
  - This didn’t take the state of the fire which changed every moment into consideration.
  - FB’s movement might have been wasteful.
  - FB may be short.

Our team’s method this year

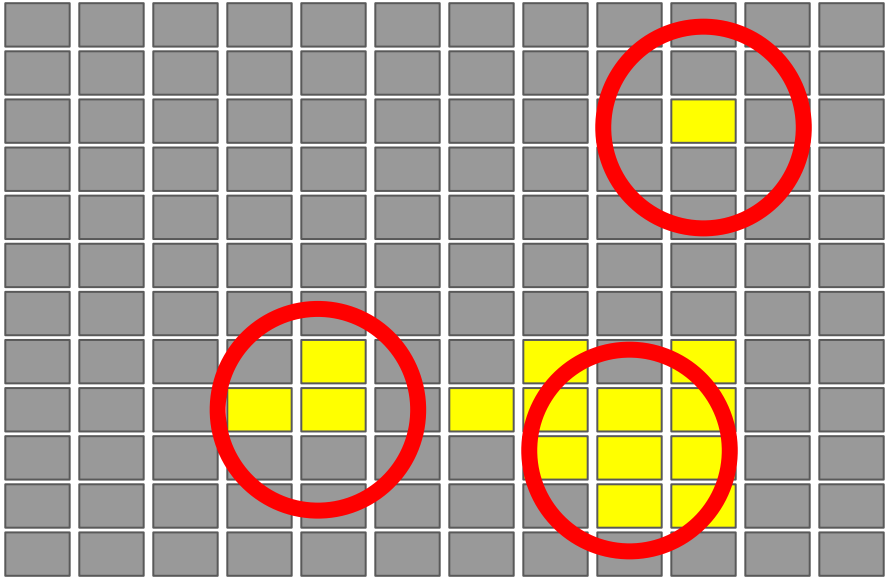
- Dynamic cluster

Algorithm of Dynamic Clustering



A fire station collected information of firing buildings from messages by agents and processed them.

Algorithm of Fire Clustering



If a firing building was detected within a certain distance from the center of the cluster, a fire station contained its entity into the cluster.

Priority of allocation of FB

80% of the agent's targets to extinguish fire were determined.

- The top priority is the initial firing.
  - It is difficult to suppress the fire spread once.
- According to the ratio of the number of buildings of each cluster among the total number of buildings of all clusters.

The remaining 20% of the FB searched to find new fires.

Target Allocators (AT)

We analyzed strategies of other teams to improve to make a better score.

Table. The results of analysis of VC3's log data of the top 4 in RoboCup2017

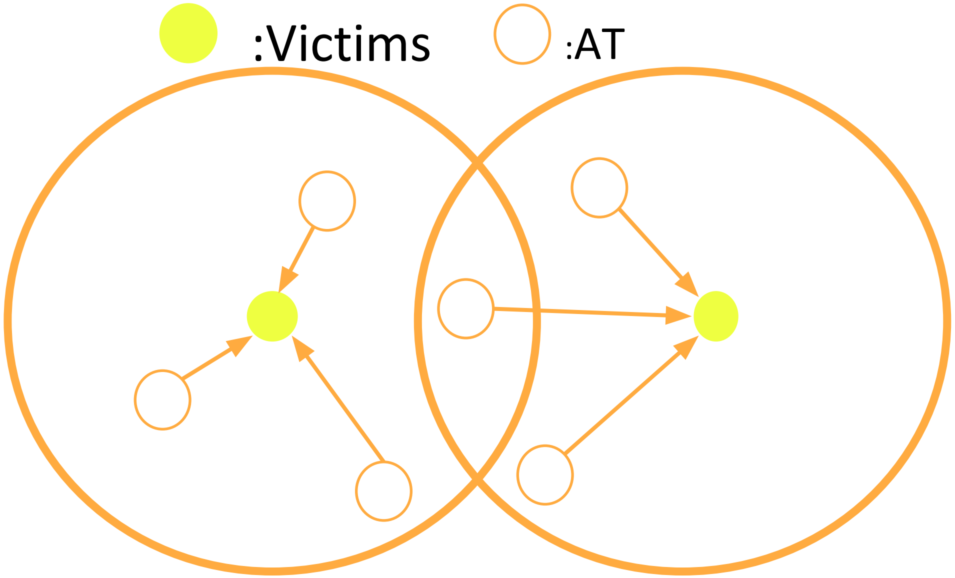
Team	Rate of rescued victim	Rate of rescue action	Rate of *cooperation*
Aura	10%	20%	23%
MRL	22%	17%	22%
RoboAKUT	13%	18%	21%
SEU-UniRobot	9%	22%	19%

Other team built groups of AT dynamically and reduced time to rescue victims.

Target Allocators (AT)

- Team Algorithm (Rescue Simulation)
- Input :
  - N:Ambulance Team
  - M:Victims
  - Priority list for each AT and victims
- Output:
  - {N,M} pairs

Target Allocators (AT)



・AT set victim as a candidate for allocation in order of highest priority.

・Selected AT： existing in the circle centering on a victim in order of closeness.

The above was operation when a victim(M) is more than AT(N).
The validity was not lost when AT(N) is more than the victim(M).

Target Allocators (AT)

Possibility AT wasn’t assigned victim occurred in the case of the algorithm of group action.

- Greedy Algorithm (Rescue Simulation)
- Input :
  - One Ambulance Team
  - M Victims
  - Preference list for each AT and Victim
- Output:
  - One pairs

Utilizing this greedy algorithm, we allocated the highest priority victim to the AT that did not occur.
Although it is a simple algorithm, AT with no matching can be set to 0.

Target Allocators (AT)

- Greedy algorithm (Rescue simulation)
- Advantages :
  - List of priority can be reused in *team-algorithm*.
  - The calculation is up to O(N).

PF’s strategy

In order to reflect information PF got to a target for the detector,

a hashmap was implemented,

inserted a priority as a value into using each EntityID of all roads a key.

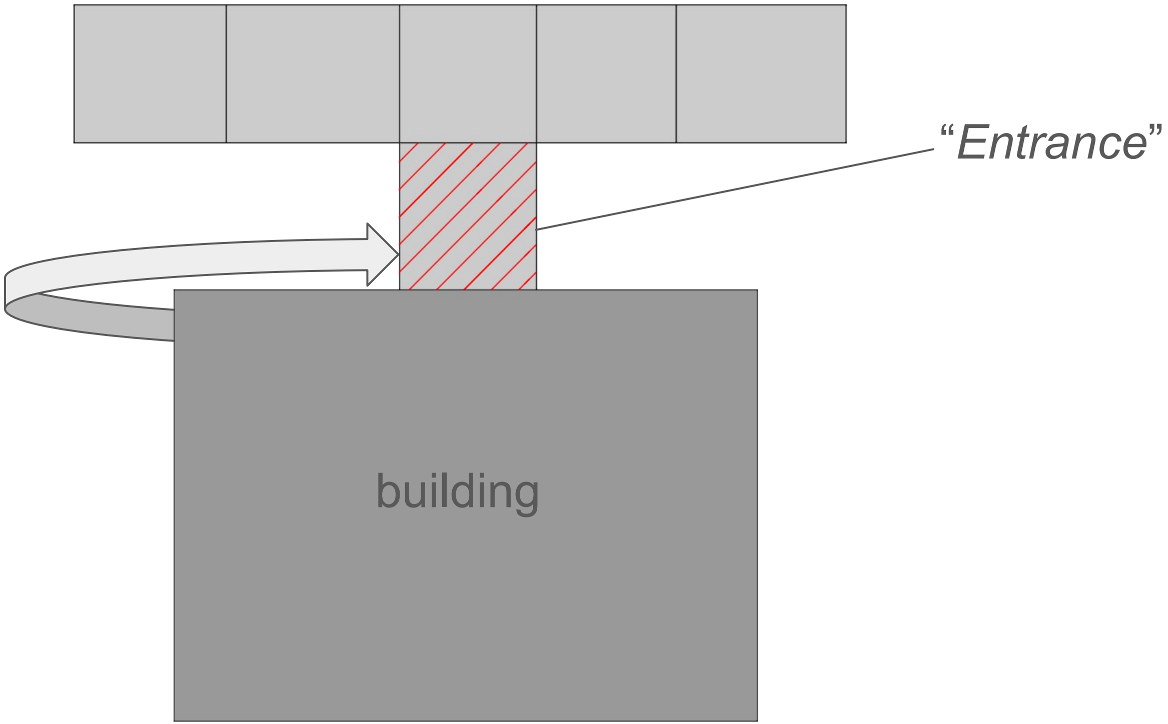
The priority corresponds to the following 7 items

High priority

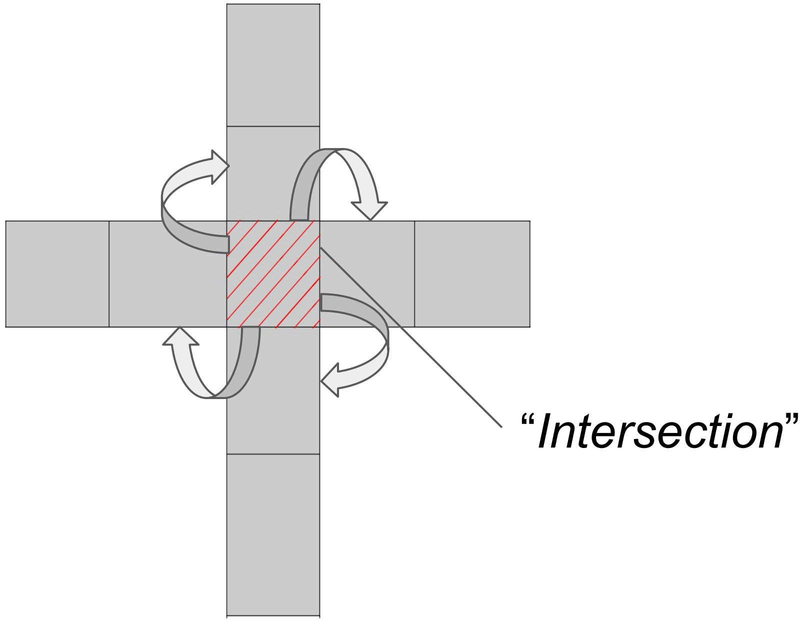
・*Entrances* of refuge

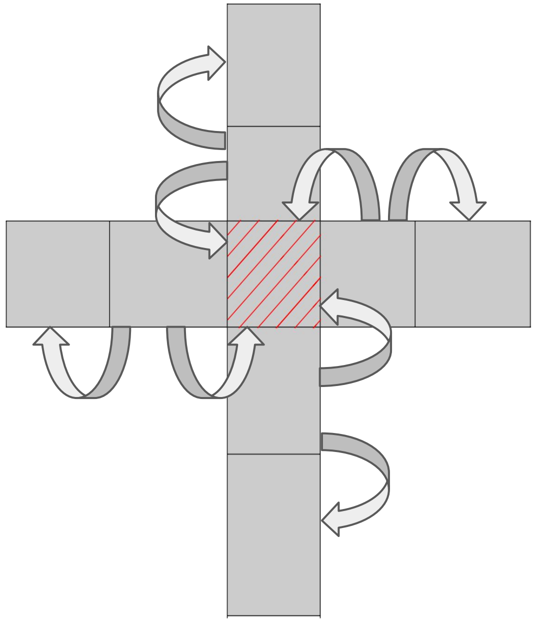
・Roads regarded as an *intersection*

Because refuges and intersections had a high traffic volume.



An *entrance* is a road adjacent to a building.





An *intersection* neighbors 4 roads.

The 4 roads neighbor 2 roads each.

An *intersection* satisfies both conditions.

The priority corresponds to the following 7 items

・Roads on which AT or FB are in the blockages

AT and FB were important for a disaster relief, so the priority of roads was increased in the case AT or FB were caught in blockades on the road.

The priority corresponds to the following 7 items

・Roads in a cluster

・*Entrances* of the buildings in which humans are buried

It is inefficient that several PF clear the same road.

Therefore, the priority of the road in a cluster was increased to share work sections.

The priority corresponds to the following 7 items

Low priority

・Roads close to burnt-out buildings

Few civilians were lived around burnt-out buildings.

AT and FB were also not there.

So the priority of roads around burnt-out buildings was decreased.

The priority corresponds to the following 7 items

・Roads without blockades

In the case aftershocks occurred, the memory was initialized.

The priority corresponds to the following 7 items

In addition to the former 7 conditions, the priority of value was determined by a type of message received.

In the case the message received was a message that AT or FB were moving, the priority of the destination was increased.

In other cases, the priority of roads with AT or FB was decreased.

Thank You!

Clustering

$P(x_i) = D(x_i)^2 / \Sigma_k D(x_i)^2$

This probability is decided to choose Initial centers spreadly as possible. It means the farther point x is from other center,larger the point’s probab.

AT

The rate of cooperation :
calculated according to the ratio of the number of rescue teams rescuing the same target among the number of rescues of all steps.

Tree AT:
We executed a code on various maps many times. As a result, the optimal number of AT regarding agent's movement on average is three.

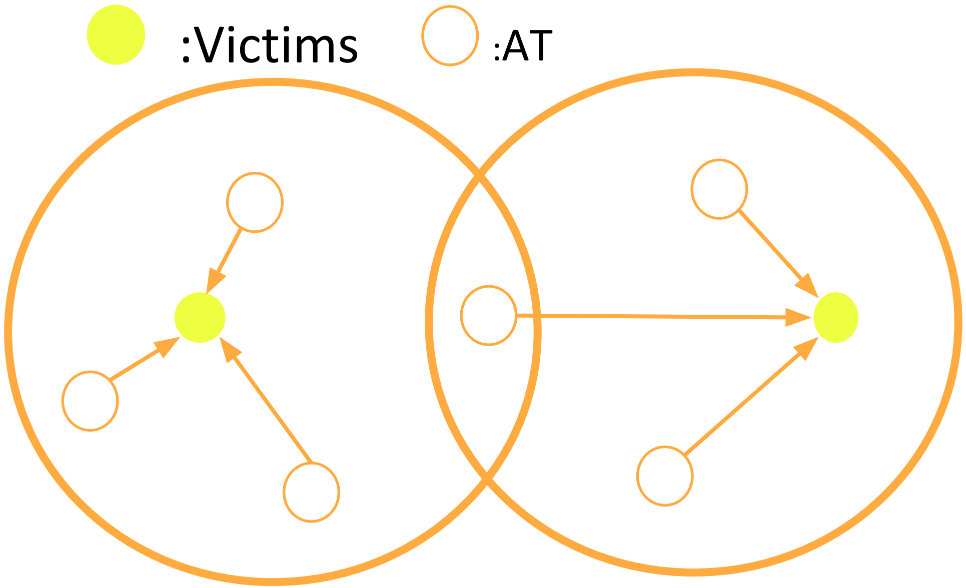
AT

Priority of victim:

First, we decide the priority following HP of each victim. This is because it is possible that civilians which have low HP die while AT rescue them.

Also, we decide the priority how many victims are in the same building and the more victims there are in it, the larger the priority is. This enables an agent to rescue the next target rapidly without moving after rescuing the target in a group.

AT



First, we take a list of AT that are in the range of circles centered on a Victim.
Next, we sort the list in descending order of priority, extract the upper 3 agents, and assign them.

AT

O(N):

We focused on the number of loop nesting and complexity of API which was used in the loop. We accelerated O(N) with changing to a data structure and reviewing an algorithm. As a result, this enable complexity to be O(N).

FB

Initial firing：
If the size of the generated cluster is less than 3, we defined it as initial firing.

Allocation:
FB are allocated to extinguish initial firing completely and make slow spread the large scale fire. It is not contradictory because they have different role.

It was assumed that a large number of FB could not reach the fires. We allocate FB not with a distance but in random.

PF

Entrance:

PF enable AT to rescue buried people.