the minimize boundary as the candidate intersections. Figure 3 shows some internal states and Figure 4 present the result. In this situation, there is only one candidate intersection.

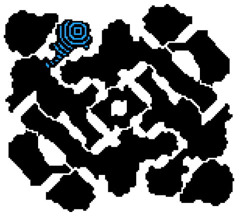

Figure 3: Radar-like Internal states

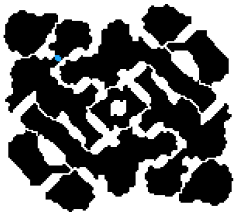

Figure 4: Radar-like Detection Result

2.1.3 Validity of Intersections. After getting the candidate intersections, we need to verify that the detected intersections can separate the map into multiple parts. Our method compares the number of the reachable grids from the start point before and after removing the intersections to determine whether the intersection is valid. If the number is significantly decreased, this intersection is valid.

2.2 Building Arrangement Search

After detecting of the critical intersections in the map, we utilize the building-arrange algorithm based on A* algorithm to find the optimal building arrangement for blocking the intersection.

2.2.1 Border of the intersection. In order to block the intersection, the building must be close to the border, the unpassable areas and unbuildable areas.

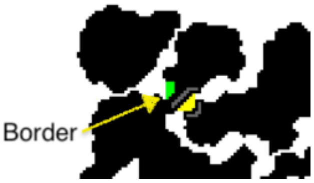

Figure 5: Border and Possible Building Location

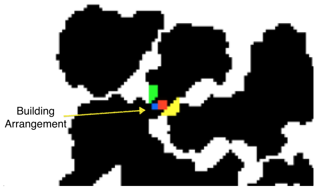

Figure 6: Building Arrangement Result from A* Search

We located the border areas near the intersection by finding all unpassable and unbuildable grids within three grids from the intersection. The green and yellow areas in Figure 5 show the two border areas of the intersection.

2.2.2 Building-arrange Algorithm for Building Arrangement. In StarCraft II, Terran usually use Supply Depot (2×2 building) and Barracks (3×3 building) for blocking the intersection, and Protoss usually use Pylon (2×2 building) and Gateway (3×3 building) for blocking.

The blocking arrangement of the building should have two properties. First, at least one building is close to each of the border; otherwise, the unit can pass through the gap between the border and the building. Second, each building must be adjacent to at least one building; otherwise, the unit can pass through the gap between the buildings.

We designed the building-arrange algorithm based on A* search algorithm to search the building arrangement (as the node) from one side of the border until reaching to another side of the border (goal). Algorithm 1 shows the algorithm process. The *candidate*() obtain the candidate building locations near the current border (Gray color areas in Figure 5). The *minManh*() gain the minimize Manhattan distance between buildings in the current state and another part of the border (the heuristic function). Moreover, the *boundary*() acquires the length of the exposure boundary of the current state's buildings (the cost function).

Algorithm 1 Building-Arrange Algorithm

```
Input: start, goal(), candidates(), boundary(),
if goal(start) = true then
    return arrangment(start)
$open \leftarrow start, closed \leftarrow \emptyset$
while $open \neq \emptyset$ do
    sort(open)
    $n \leftarrow open.pop()$
    $candidates \leftarrow candidates(n)$
    for all $cand \in candidates$ do
        $cand.f \leftarrow cand.boundary + minManh(cand)$
        if $goal(cand) = true$ then
            return arrangment(cand)
        if $goal(cand) = false$ then
            $open \leftarrow cand$
    $closed \leftarrow n$
return $\emptyset$
```

The minimize Manhattan distance is the lower bound of the exposure boundary for any blocked buildings arrangements, which can make sure that the algorithm obtains the optimal solution [1]. The time complexity in the worst case is $O(b^d)$, b is the number of the candidates and d is the number of buildings in the arrangement result. Figure 6 shows the building arrangement result from the building-arrange search.

3 EXPERIMENTS

We conducted experiments on all maps from the StarCraft II Ladder 2017. For the original base location and the closest expand resource location, our approach could find the right intersection and plan the appropriate building locations for all maps. The number of nodes expanded in search for all map is less than one hundred, which meet the real-time requirement. For the start point in the middle of the map, due to the increase in connectivity paths and the variety of the terrain, our approach build lots of redundant buildings for blocking the location and need to expand hundreds of nodes to get the building arrangement.

REFERENCES

[1] P. E. Hart, N. J. Nilsson, and B. Raphael. 1968. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics* 4, 2 (July 1968), 100–107.

[2] O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, J. Quan, S. Gaffney, S. Petersen, K. Simonyan, T. Schaul, H. van Hasselt, D. Silver, T. P. Lillicrap, K. Calderone, P. Keet, A. Brunasso, D. Lawrence, A. Ekermo, J. Repp, and R. Tsing. 2017. StarCraft II: A New Challenge for Reinforcement Learning. *CoRR* abs/1708.04782 (2017).