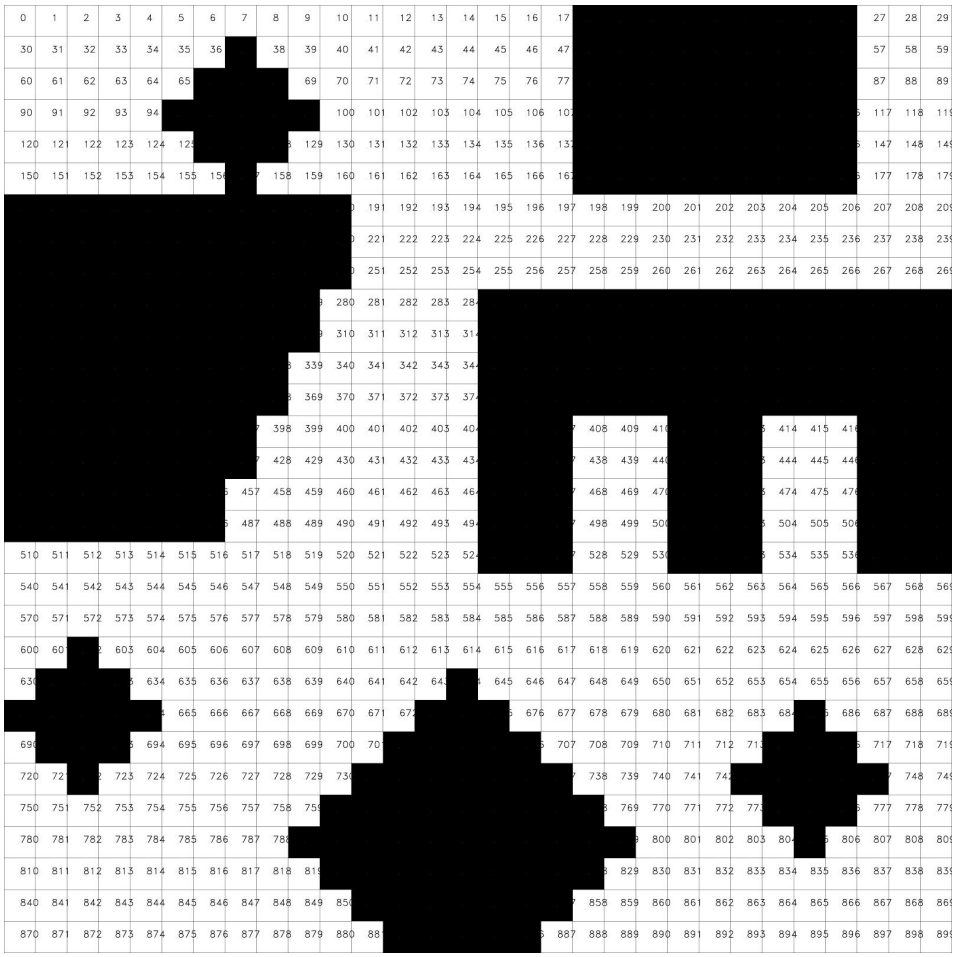
IPAS using Bayesian Optimization

The true map is:



In this case, we have 4 actors and 3 sensors. The initial position for each actor:

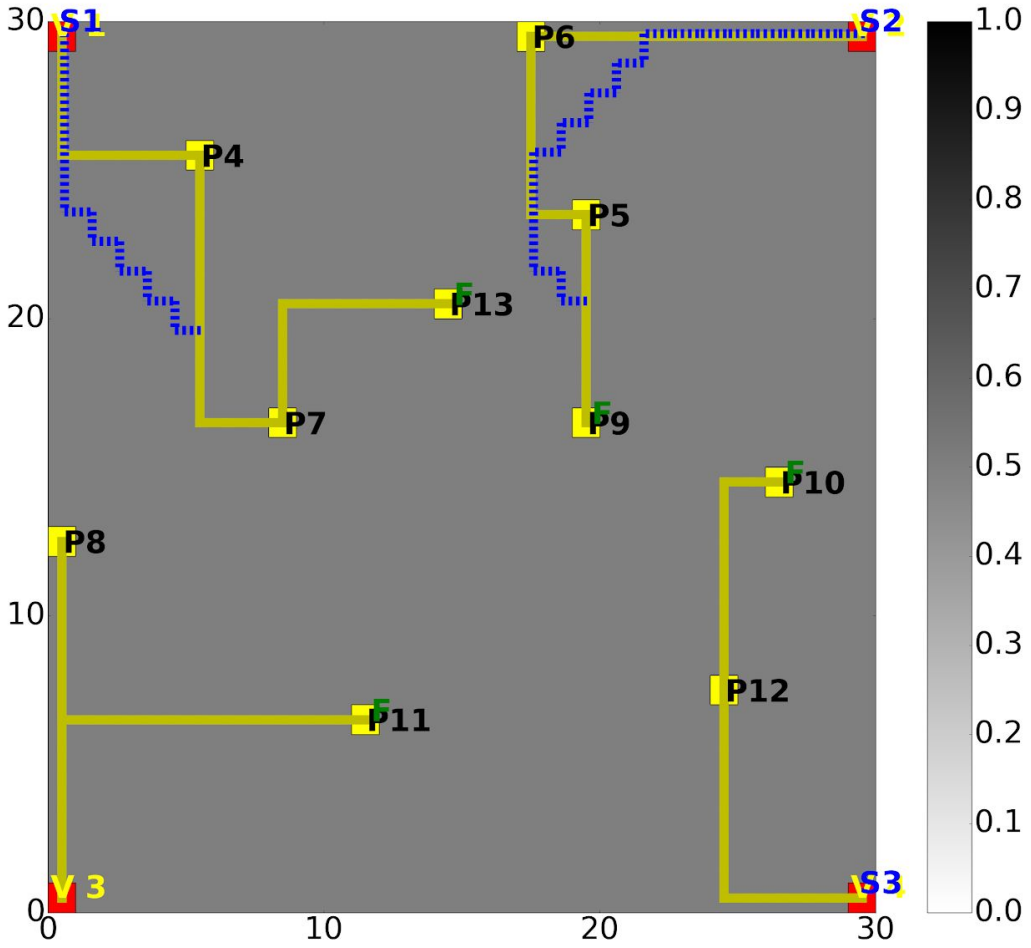
Actor	Initial Position
V1	0
V2	29
V3	870
V4	899

So we have **4 actors in the four corners of the map**.

The initial position for each sensor:

Sensor	Initial Position
S1	0
S2	29
S3	899

Initially, the map is totally unknown. 10 target positions are required to be visited by the group of actors. The initial map looks like:



The result computed by using the original IPAS (Computing the information gain for each vertex and selecting the ones with maximum information gain for measurement at next iteration.) is shown in the file:
Grid-map-IPAS.mp4 (Shared by onedrive)
IG-map-IPAS.mp4

The result computing by Bayesian optimization (Selecting the vertex for measurement via Bayesian Optimization) is shown in the file:
Grid-map-Bayesian.mp4
IG-map-Bayesian.mp4

Comparison:
Based on the result, we can see the selection of next measurement using Bayesian optimization is pretty accurate. The average error between the estimation using bayesian optimization and the actual maximum information gain is 0.068. Most of the errors appear right before the algorithm converges (Check the file: IG-map-Bayesian.mp4). You can see at the iterations approaches to the end, we put the sensors at vertices which don't have maximum information gain (the vertices with dark red). This is because, the estimation of Bayesian optimization is not accurate enough. But these error doesn't affect the convergence rate that much. So using the original IPAS, 54 iterations are required. And the interactions required by using Bayesian optimization is 55.

Meanwhile, using Bayesian optimization, at each iteration, less than 45 vertices are computed for its information gain and used as training data for bayesian optimization. Using Bayesian optimization do speed up the computation of IPAS a lot.