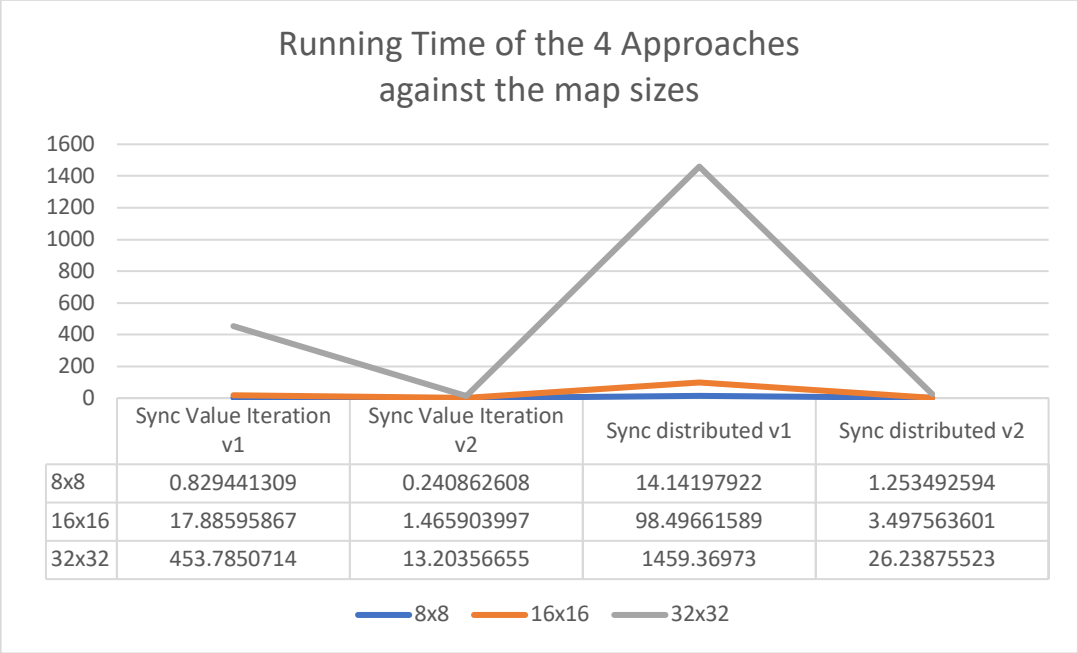
(CS 533) Intelligent and Decision Making

Homework 2:
Distributed
Synchronous Value Iteration

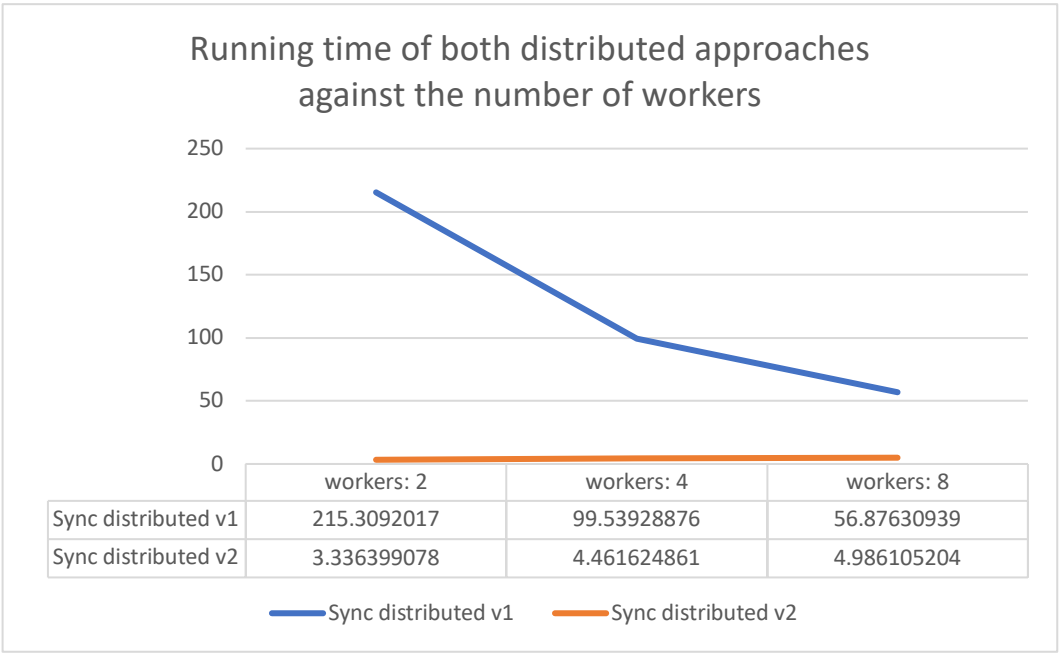
A-young Kang (Student ID: 933-610-350)

1. A plot that shows the running time of the above 4 approaches against the map sizes of 8, 16 and 32.



	Sync Value Iteration v1	Sync Value Iteration v2	Sync distributed v1	Sync distributed v2
8x8	0.829441309	0.240862608	14.14197922	1.253492594
16x16	17.88595867	1.465903997	98.49661589	3.497563601
32x32	453.7850714	13.20356655	1459.36973	26.23875523

2. A plot that shows the running time of both distributed approaches against the number of the workers with 2, 4 and 8 workers.



	workers: 2	workers: 4	workers: 8
Sync distributed v1	215.3092017	99.53928876	56.87630939
Sync distributed v2	3.336399078	4.461624861	4.986105204

3. Briefly explain why the second distributed method is faster than the first one?

In the first distributed method, a distinct worker performs the Bellman Backup for a state and update the value and policy for the state. On the other hand, the second distributed method partitions the states into batches so the update of the value and policy occurs every once per batch. Therefore, the second distributed approach is faster than the other one.

4. Compare the best distributed method with the best non-distributed approach. Which one is better? Briefly explain why.

As we can interpret from the graph above, resulted time of every map size in Sync_Value_Iteration_V2 is shorter than Sync_Value_Iteration_V1. This is because the former does not loop through the next states with zero transition probability. This method also applies to the Sync Distributed cases, as we can see the result from the sync distributed v2 is faster than sync distributed v1. For the three cases we tested, all the results from distribute_v2 took more time than non-distributed v2. However, let's compare the rate of increase. For the case of non-distributed v2, when the map size is 16, the result is 1.46 and for 32, it's 13.20. For the case of distributed v2, when the map size is 16, the result is 3.49 and for 32 it's 26.23. When we compare the rate of increase of both cases, we can see that distributed v2 is lower than the non-distributed one. So, referred to above result, the bigger the map size get sync distributed v2 will get much faster.