AI Assignment 3

Theory

1.

(a) Direct sampling works by sampling from the prior distribution of the nodes in Baylsian network, following along the dependencies

Strength : Efficient ~~and~~ sampling when conditional probabilities like P(Leisure | train) and marginal probabilities like P(train) are explicitly given. Samples generated are consistent with precise values in probability distribution.

Weakness : Requires high number of samples to perform accurate sampling for events with low probabilities like P(low_stress, bus)
Requires explicit knowledge of ~~probability of condition~~ marginal or joint probabilities like P(train) to utilise conditional probabilities like P(leisure | train).

Rejection sampling works by sampling using prior distribution and rejecting samples that differ from the evidence. The remaining samples provide the required probability estimate.

Strength : Performs well if estimated distribution ~~is~~ has very less deviation from actual distribution, for e.g., P(train) is sampled as the base of P(leisure|train).
Incomplete knowledge of conditional probability like P(high_stress | air) but knowing marginal or joint probability like P(high_stress, air) allows to estimate actual distribution.

Weakness : If the estimated distribution significantly deviates from actual distribution, a large number of samples are discarded
Computationally costly when events with low probability are sampled due to large number of samples being rejected like for P(low_stress, bus)

Gibbs sampling is a MCMC method where ~~iterative~~ iterative sampling is done by conditioning on the values of all other variables in the Bayesian network

Strength : Efficient even when the number of variables are high, as compared to direct and rejection sampling.
The method can estimate low probability events with high accuracy and efficiency. It can also estimate complex

joint and conditional ~~distribution~~ probabilities with high accuracy & like P(high-stress, air, business).
Weakness: Requires high number of iterations to converge to the actual distribution. Sensitive to initial values, which can delay convergence ~~of~~ to actual distribution.
Eg: Estimating P(high-stress/air) ~~= 0.6~~, it iteratively conditions on high-stress and samples air, which ~~tapro~~ converges to actual value over time

(b) ~~P(leisure|train) = 0.4~~ P(leisure | train) = 0.4 = x
Number of people out of 100 preferring train = 30 = y

Number of people travelling for leisure given they are travelling by train = $x \cdot y$ = 0.4 × 30
= 12

(c) P(air) = 0.80, P(business | air) = 0.20

P(air, business) = P(business | air) P(air) = 0.20 × 0.80 = 0.160

(d) Larger sample sizes reduce error due to random sampling allowing better estimates that are more accurate of actual distribution. Events with high probability quickly converge to actual probability values. More number of samples increase the occurrence of samples for low probability events like (bus, low-stress), increasing both accuracy and precision. The accuracy of for conditional probabilities like P(leisure|train) also increases since the number of samples for train increases. The joint probabilities are more accurate estimate for joint probabilities as the number of samples of each variable involved increases. High number of samples reduce deviation or variance of estimated distribution from actual distribution.

2. (a) Let J represent the event that a person accesses journals.
Let C represent the event that a person goes to book clubs.
Let R represent the event that a person reads books

S1: $P(R \vee J) = 0.91$
S2: $P(\neg R \wedge \neg J) = 0.09$
S3: $P(J \mid R) = 0.4$, $P(\neg J \mid R) = 0.6$
S4: $P(J \wedge \neg R) = 0.227$
S5: $P(J) = 0.5$

S6 : $P(J \mid \neg R) = 0.716$

S7 : $P(C \mid R) = 0.32$

S8 : $P(C \wedge J) = 0.088$

S9 : $P(C \vee J) = 0.631$

S10 : $P(J \mid C) = 0.4$

S11 : $P(C \mid \neg R) = 0.0044$

(b) $P(E) \geq 0 \ \forall E \in$ Sample Space in given dataset

$$P(R \vee J) + P(\neg R \wedge \neg J) = 0.91 + 0.09 = 1 \quad \Big(\sum_{E \in \Omega} P(E) = 1\Big)$$

$\Omega \rightarrow$ Sample space

$$P(J) = P(R \wedge J) + P(\neg R \wedge J)$$

$$\Rightarrow P(J) = P(J \mid R).P(R) + P(\neg R \wedge J)$$

$$\Rightarrow 0.5 = 0.4 . P(R) + 0.227$$

$$\Rightarrow P(R) = \frac{0.5 - 0.227}{0.4} = 0.683$$

$$P(R \wedge \neg J) = P(\neg J \mid R).P(R) = 0.6 \times 0.683 = 0.410$$

$$P(R \wedge J) = P(J \mid R).P(R) = 0.4 \times 0.683 = 0.273$$

$$P(\neg J \mid R) + P(J \mid R) = 0.6 + 0.4$$

$$\Rightarrow \frac{P(\neg J \wedge R) + P(J \wedge R)}{P(R)} = 1$$

$$\Rightarrow \frac{P(R)}{P(R)} = 1$$

$$P(R \vee J) = \underbrace{P(R) + P(J) - P(R \wedge J)}_{\text{Inclusion - Exclusion principle}} = 0.683 + 0.5 - 0.273 = 0.91 \text{ (given)}$$

$$P(R \wedge J) + P(R \wedge \neg J) + P(\neg R \wedge J) + P(\neg R \wedge \neg J)$$

$$= 0.273 + 0.410 + 0.227 + 0.09$$

$$= 1 \quad \Big(\sum_{E \in \Omega} P(E) = 1\Big)$$

$\Omega \rightarrow$ Sample space

(c) $P(\neg R) = 1 - P(R) = 1 - 0.683 = 0.317$

$P(\neg J) = 1 - P(J) = 1 - 0.5 = 0.5$

(c) $P(C \wedge J \wedge R) = P(C|R) \cdot P(J|R) \cdot P(R)$

$= 0.32 \times 0.40 \times 0.683 = 0.0874$

$P(C \wedge J \wedge \neg R) = P(C|\neg R) \cdot P(J|\neg R) \cdot P(\neg R)$

$= 0.0044 \times 0.716 \times 0.317 = 0.001$

$P(C \wedge \neg J \wedge R) = P(C|R) \cdot P(\neg J|R) \cdot P(R)$

$= 0.32 \times 0.60 \times 0.683 = 0.131$

$P(C \wedge \neg J \wedge \neg R) = P(C|\neg R) \cdot (1 - P(J|\neg R)) \cdot P(\neg R)$

$= 0.0044 \times (1 - 0.716) \times 0.317 = 0.0004$

$P(\neg C \wedge J \wedge R) = (1 - P(C|R)) \cdot P(J|R) \cdot P(R)$

$= (1 - 0.32) \times 0.40 \times 0.683 = 0.186$

$P(\neg C \wedge J \wedge \neg R) = (1 - P(C|\neg R)) \cdot P(J|\neg R) \cdot P(\neg R)$

$= (1 - 0.0044) \times 0.716 \times 0.317 = 0.226$

$P(\neg C \wedge \neg J \wedge R) = (1 - P(C|R)) \cdot P(\neg J|R) \cdot P(R)$

$= (1 - 0.32) \times 0.60 \times 0.683 = 0.278$

$P(\neg C \wedge \neg J \wedge \neg R) = (1 - P(C|\neg R))(1 - P(J|\neg R)) \cdot P(\neg R)$

$= (1 - 0.0044) \times (1 - 0.716) \times 0.317$

$= 0.090$

(d) For C and J, given R

$P(C|R) = \dfrac{P(C \wedge J \wedge R) + P(C \wedge \neg J \wedge R)}{P(R)} = \dfrac{0.087 + 0.131}{0.683} = 0.319$

$P(J|R) = \dfrac{P(C \wedge J \wedge R) + P(\neg C \wedge J \wedge R)}{P(R)} = \dfrac{0.087 + 0.186}{0.683} = 0.4$

$P(C \wedge J | R) = \dfrac{P(C \wedge J \wedge R)}{P(R)} = \dfrac{0.087}{0.683} = 0.127 = 0.319 \times 0.4$

$= P(C|R) \cdot P(J|R)$

$\therefore BC \perp J | R$

For C~~B~~ and R given J,

$$P(C \mid J) = \frac{P(C \wedge J \wedge R) + P(C \wedge J \wedge \neg R)}{P(J)} = \frac{0.087 + 0.001}{0.5} = 0.176$$

$$P(R \mid J) = \frac{P(C \wedge J \wedge R) + P(\neg C \wedge J \wedge R)}{P(J)} = \frac{0.087 + 0.186}{0.5} = 0.546$$

$$P(C \wedge R \mid J) = \frac{P(C \wedge R \wedge J)}{P(J)} = \frac{0.087}{0.5} = 0.174$$

$$\neq 0.176 \times 0.546 = 0.096$$

∴ C and R not conditionally independent given J.

For J and R given C

$$P(J \mid C) = \frac{P(C \wedge J \wedge R) + P(C \wedge J \wedge \neg R)}{P(C \wedge J) + P(C \wedge \neg J)} = \frac{0.087 + 0.131}{0.088 + 0.131 + 0.001 + 0}$$

$$= 0.4$$

$$P(R \mid C) = \frac{P(C \wedge J \wedge R) + P(C \wedge \neg J \wedge R)}{P(C \wedge J) + P(C \wedge \neg J)} = \frac{0.087 + 0.131}{0.088 + 0.131 + 0.001 + 0}$$

$$= 0.991$$

$$P(J \wedge R \mid C) = \frac{P(C \wedge J \wedge R)}{P(C \wedge J) + P(C \wedge \neg J)} = \frac{0.087}{0.088 + 0.131 + 0.001 + 0}$$

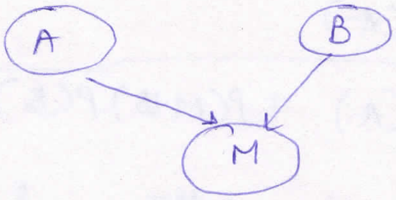
$$= 0.395 = 0.4 \times 0.991 = P(J \mid C) \cdot P(R \mid C)$$

∴ $J \perp R \mid C$

3. (a) Let A be the event that misclassification is caused by adversarial perturbation.

Let B be the event that misclassification is caused by backdoor attack.

Let M be the event that a misclassification alarm is observed.



Bayesian network

Initially $P(A \cap B) = P(A)\,P(B)$

Reports indicate that backdoor attacks are becoming prevalent, This implies an increase in $P(B)$.

Objective is to apply Bayesian inference and find the effect of ~~incre~~ $P(B)$ on likelihood that an adversarial perturbation caused the misclassification alarm, that is $P(M|A)$.

(b) Prior probability:

$P(A)$: ~~Initia~~ Initial probability of adversarial perturbation ~~causing misclassification~~

$P(B)$: Initial probability of backdoor attack

Likelihood:

$P(M|A)$: Probability of observing ~~a~~ misclassification alarm given the occurrence of adversarial attack

$P(M|B)$: Probability of observing misclassification alarm given the occurrence of backdoor attack

Posterior probability:

$P(A|M)$: Probability that ~~an~~ adversarial perturbation caused misclassification given the misclassification alarm

$P(B|M)$: Probability that backdoor attack caused misclassification given the misclassification alarm.

$$P(M) = P(M|A)\,P(A) + P(M|B)\,P(B) \quad \text{— ①}$$

$$P(A|M) = \frac{P(M|A)\,P(A)}{P(M)} \quad \text{where } P(M) \text{ from ①}$$

$$P(B|M) = \frac{P(M|B)\,P(B)}{P(M)} \quad \text{where } P(M) \text{ from ①}$$

(c) $$P(A|M) = \frac{P(M|A)\,P(A)}{P(M|A)\,P(A) + P(M|B)\,P(B)}$$

An increase in $P(B)$ increases the value of denominator, decreasing $P(A|M)$.

A and B are independent causes of M, but observing M creates a dependency between A and B.

$$P(A|M,B) < P(A|M)$$

The conditioning on B reduces the effect of M on A, decreasing $P(A|M)$.

AI ASSIGNMENT 3 - Uncertainty, Bayesian Nets, HMM and Kalman Filtering

Coding

4.

```
Runtime for loading datasets: 0.03125786781311035 s
```

1.

<u>Initial Bayesian network</u>

```
Runtime for initial Bayesian network: 11.01029634475708 s
```

```
Total Test Cases: 350
Total Correct Predictions: 350 out of 350
Model accuracy on filtered test cases: 100.00%
```

Zones_Crossed = |End_Stop_ID – Start_Stop_ID|

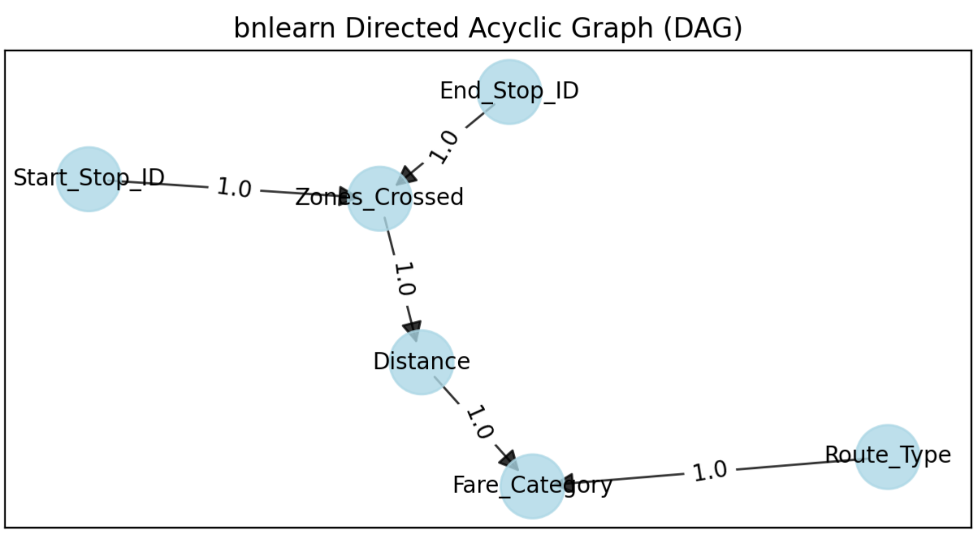
Zones_Crossed depends on Start_Stop_ID and End_Stop_ID

Distance depends on Zones_Crossed (More zones crossed implies longer distance)

Fare_Category depends on Route_Type (Faster route type implies higher fare) and Distance (Longer route type implies higher fare)

A DAG is created using these dependencies and the Bayesian network model is fitted on the DAG.

(c)



2.

Pruned Bayesian network

```
Runtime for pruned Bayesian network: 0.09171152114868164 s
```

```
Total Test Cases: 350
Total Correct Predictions: 350 out of 350
Model accuracy on filtered test cases: 100.00%
```

(b)

Route_Type = 3 for each row in the given train_data.csv (Fare_Category independent of Route_Type)

The node Route_Type is pruned.

Edge from Zones_Crossed to Distance is pruned since Distance is itself sufficient to classify Fare_Category.

The nodes Start_Stop_ID, End_Stop_ID and Zones_Crossed are pruned.

For each row in train_data.csv,

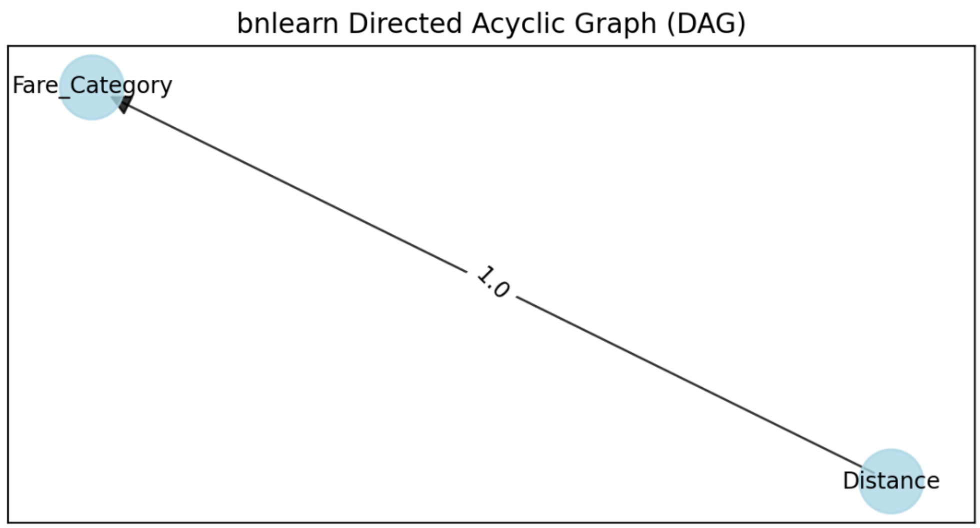
Distance	Fare_Category
Short	Low
Medium	Medium
Long	High

Fare_Category depends only on Distance.

A DAG is created using the single dependency after performing the above described pruning steps and the Bayesian network is fitted on the DAG.

The pruning method explained above improves the model's efficiency, that is reduces the time taken to fit the model as compared to initial Bayesian network (A) by

(c)



3.

<u>Optimized Bayesian network</u>

```
Runtime for optimized Bayesian network: 0.6042571067810059 s
```

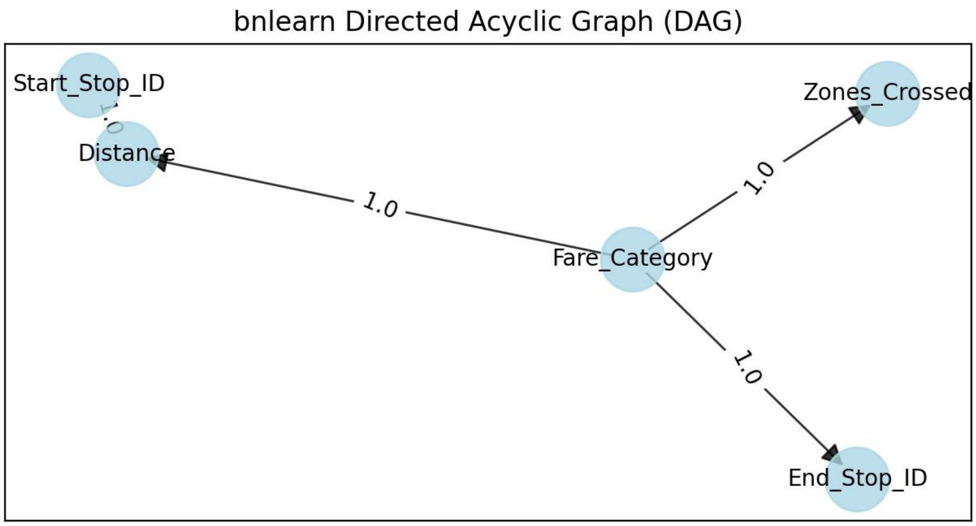
```
Total Test Cases: 350
Total Correct Predictions: 350 out of 350
Model accuracy on filtered test cases: 100.00%
```

(b)

The initial Bayesian network (A) is optimized by applying structure learning using Hill Climbing with Bayesian Information Criterion as scoring metric.

The optimization technique explained above improves the model's efficiency, that is reduces the time taken to fit the model as compared to initial Bayesian network (A) by

(c)



5.

(c), (d)

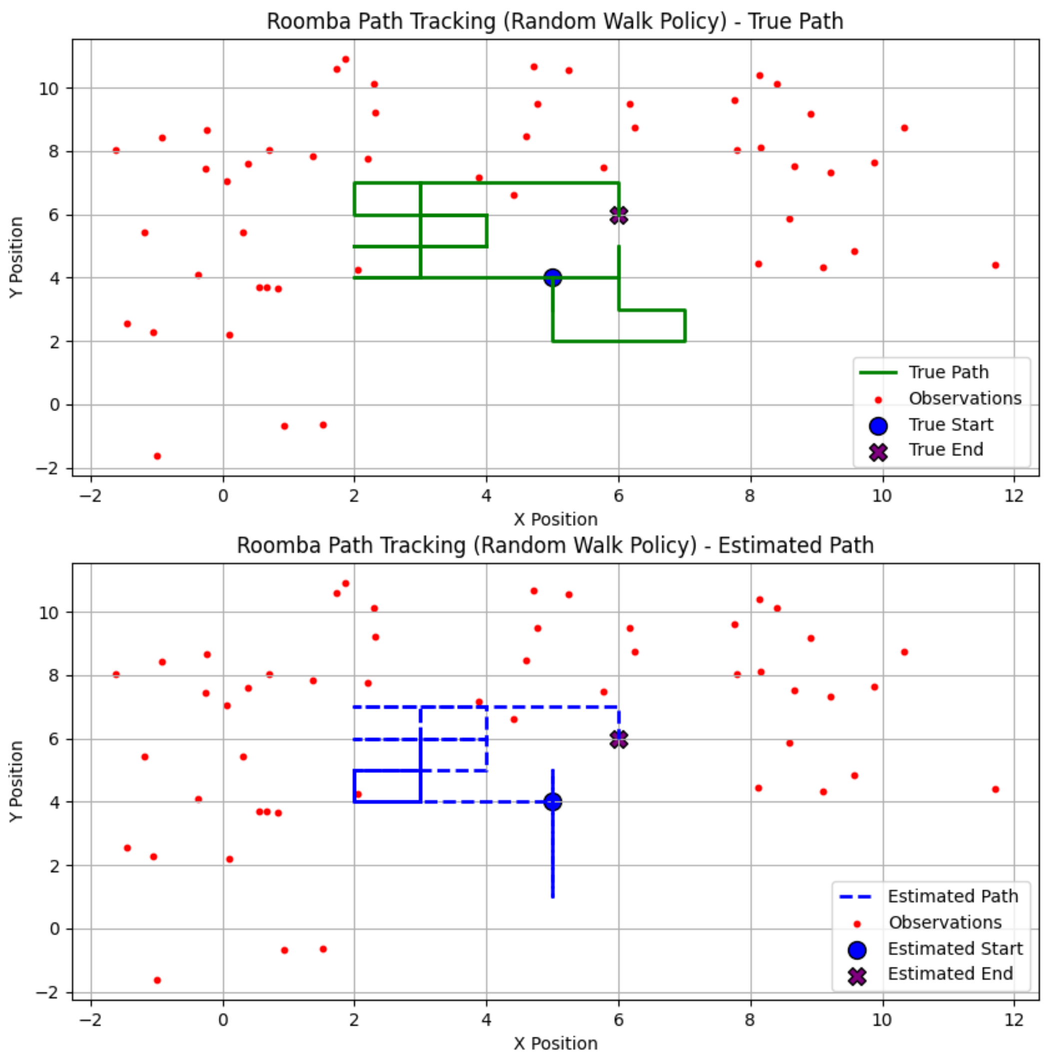
The Straight Until Obstacle Policy is more accurate since the Roomba's movement is deterministic in moving in the current direction until an obstacle is encountered, in which case it randomly selects a new direction to move in. This combination of deterministic movement until a problem/obstacle is encountered and a random response in case of problem allows it to have more accurate movement as compared to Random Walk Policy. The random movement predictions at each step lead to high levels of random, not useful movement for the Random Walk Policy.

The selected seed values are 36, 97, 101

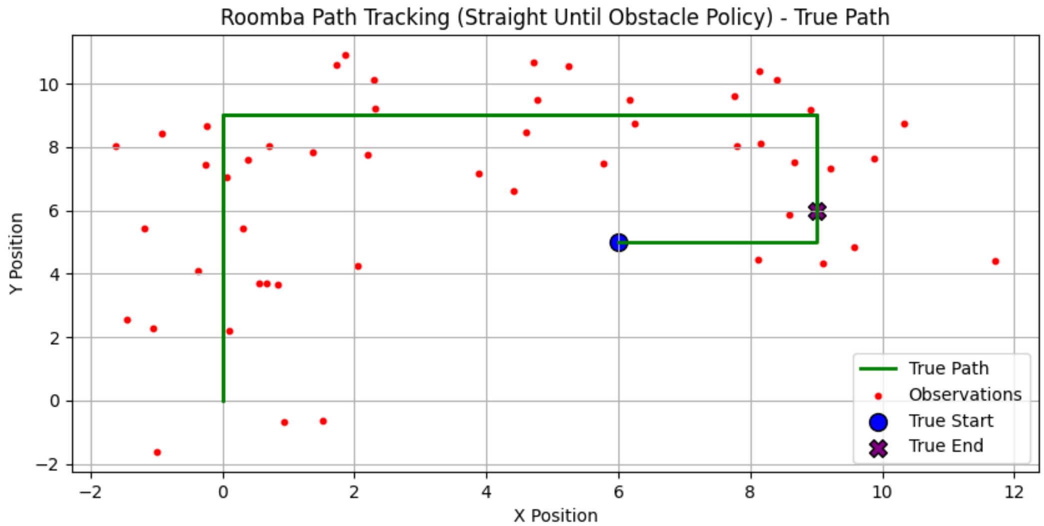
The value of seed variable was manually changed each time before the file HMM_question.py was run to get the plot for true path and estimated path and populate the estimated_paths.csv file with the estimated_path for respective seed value.

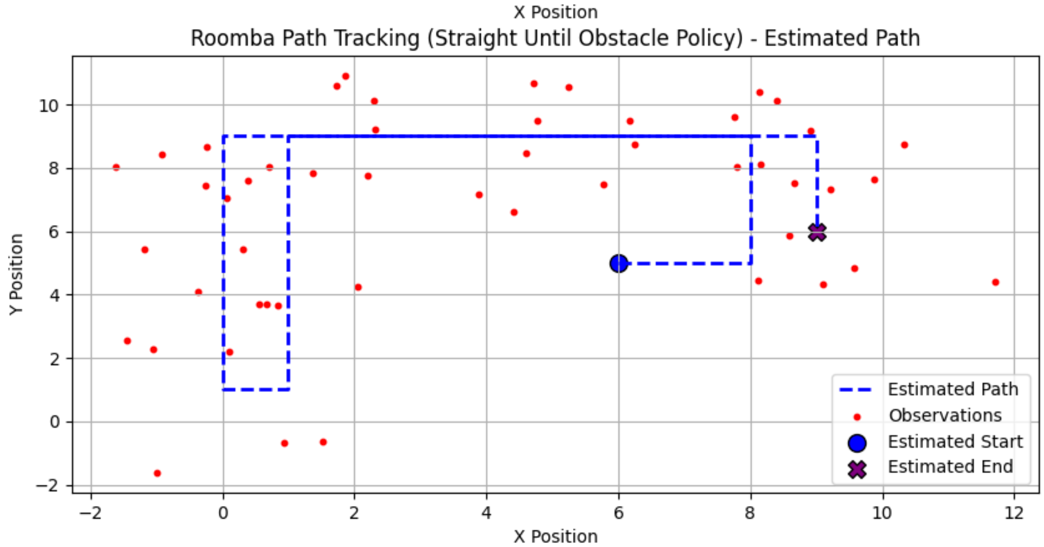
seed = 36

```
Processing policy: random_walk
Tracking accuracy for random walk policy: 32.00%
```



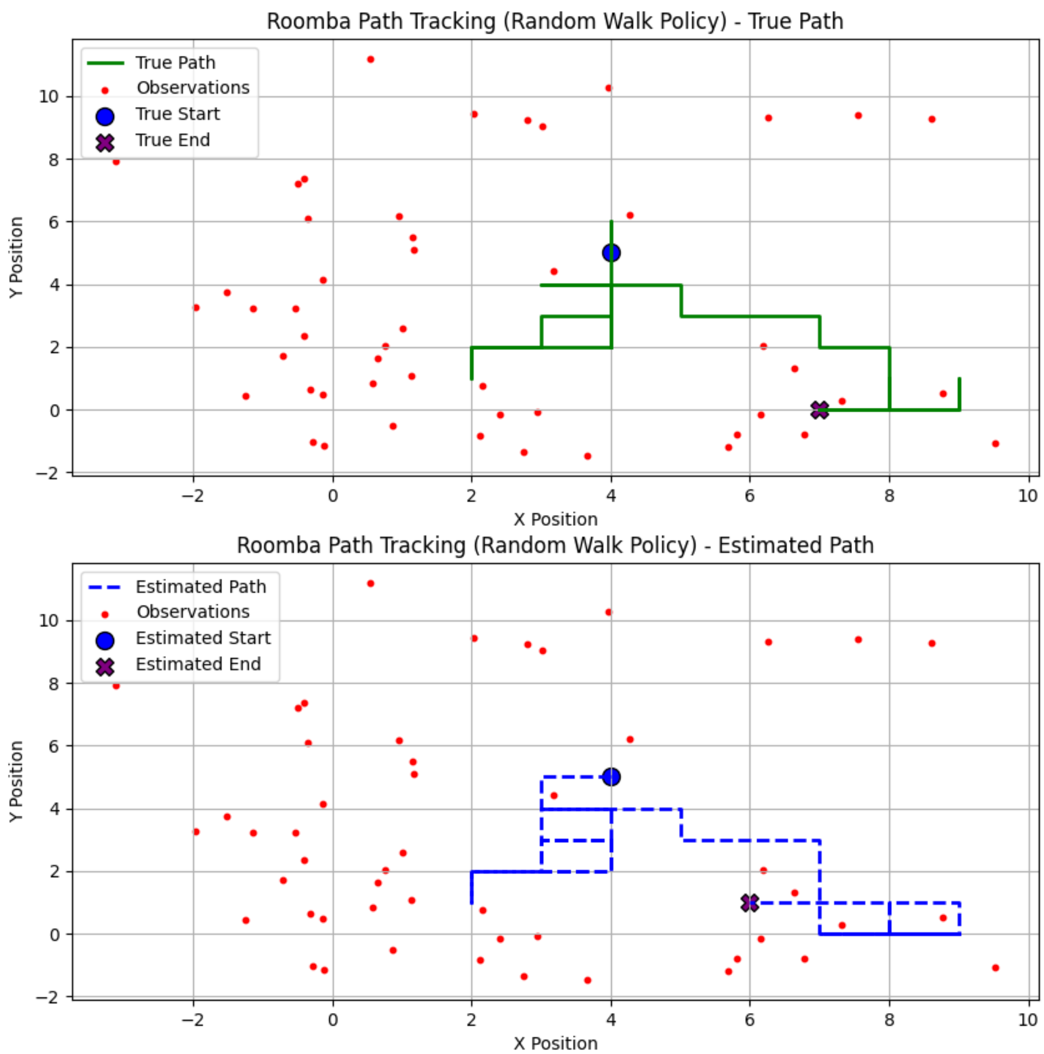
```
Processing policy: straight_until_obstacle
Tracking accuracy for straight until obstacle policy: 68.00%
```



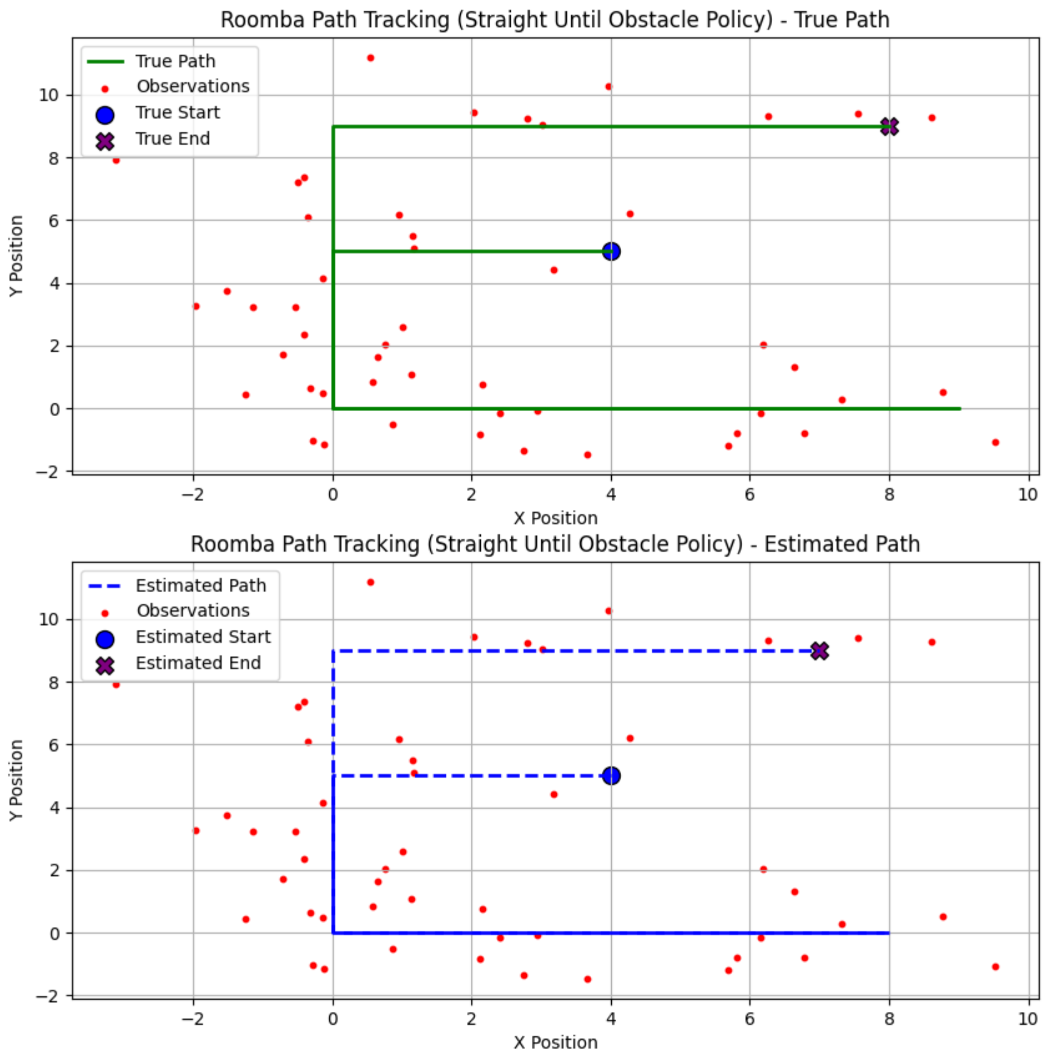


seed = 97

```
Processing policy: random_walk
Tracking accuracy for random walk policy: 60.00%
```

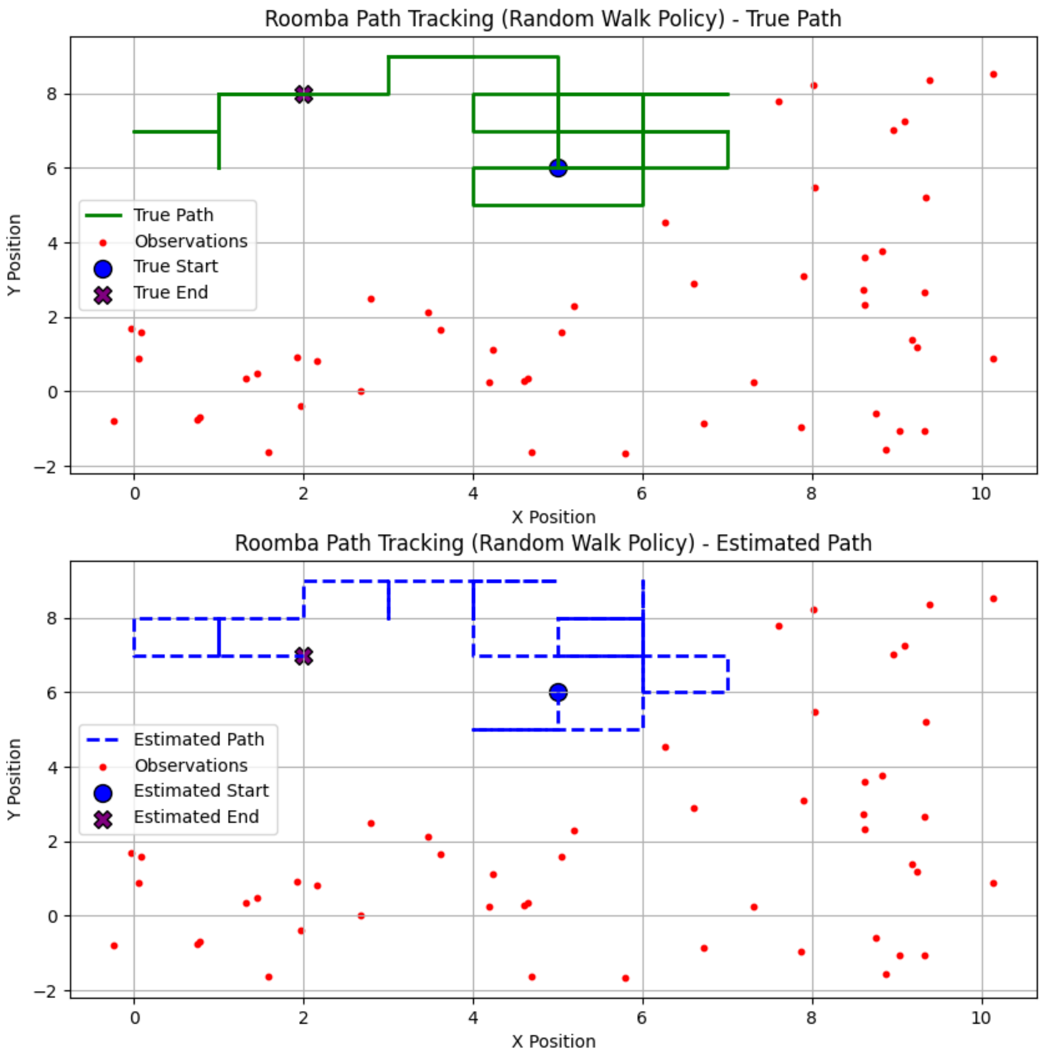


```
Processing policy: straight_until_obstacle
Tracking accuracy for straight until obstacle policy: 82.00%
```



seed = 101

```
Processing policy: random_walk
Tracking accuracy for random walk policy: 36.00%
```



```
Processing policy: straight_until_obstacle
Tracking accuracy for straight until obstacle policy: 64.00%
```