Pseudo Code for Scavenger Hunt Algorithms

October 22, 2020

A Algorithm Pseudocode

The pseudocode for all the solution algorithms described in Section 4 are presented in this section to fully specify our algorithms and to aid replicability.

Algorithm 1 Proximity-Based Search

Require: A graph $G = (N, E)$, a set of objects O, a distribution D, start location node n_0

Initialize n_0, Y_0 and τ_0 with $t = 0$
while $Y_{t,i} \neq 0, \forall i \in [k]$ **do**
  $n_{t+1} \leftarrow \underset{n \in N}{\arg\min}\{e_{n_t,n} | \exists\; o \in O, p_{t_o}^{n} > 0\}$
  Travel to n_{t+1}
  Update D, Y_{t+1} and τ_{t+1} based on the occurrence of objects at n_{t+1}
  $t = t + 1$
end while

Algorithm 2 Probability-Based Search

Require: A graph $G = (N, E)$, a set of objects O, a distribution D, start location node n_0

Initialize n_0, Y_0 and τ_0 with $t = 0$
while $Y_{t,i} \neq 0, \forall i \in [k]$ **do**
  $n_{t+1} \leftarrow \underset{n \in N}{\arg\max}\{1 - \prod_{o \in O}(1 - p_o^n)\}$
  Travel to n_{t+1}
  Update D, Y_{t+1} and τ_{t+1} based on the occurrence of objects at n_{t+1}
  $t = t + 1$
end while

Algorithm 3 Probability-Proximity Search

Require: A graph $G = (N, E)$, a set of objects O, a distribution D, start location node n_0

Initialize n_0, Y_0 and τ_0 with $t = 0$

while $Y_{t,i} \neq 0, \forall i \in [k]$ **do**

  $n_{t+1} \leftarrow \underset{n \in N}{\arg\max} \frac{1 - \prod_{o \in O}(1 - p_o^n)}{e_{n_t,n}}$

  Travel to n_{t+1}

  Update D, Y_{t+1} and τ_{t+1} based on the occurrence of objects at n_{t+1}

  $t = t + 1$

end while

Algorithm 4 DQN Algorithm

Require: A graph $G = (N, E)$, a set of objects O, a distribution D, start location node n_0, a trained policy $\pi(a|s)$

Initialize n_0, Y_0 and τ_0 with $t = 0$

while $Y_{t,i} \neq 0, \forall i \in [k]$ **do**

  $s \leftarrow$ build_observation(G,D)

  $n_{t+1} \leftarrow \underset{a \in N}{\arg\max}\, \pi(a|s)$

  Travel to n_{t+1}

  Update D, Y_{t+1} and τ_{t+1} based on the occurrence of objects at n_{t+1}

  $t = t + 1$

end while

Algorithm 5 Exhaustive Bayesian Search

Require: A graph $G = (N, E)$, a set of objects O, a distribution D, start location node n_0

Initialize n_0, Y_0 and τ_0 with $t = 0$

while $Y_{t,i} \neq 0, \forall i \in [k]$ **do**

  best_path=$\emptyset$

  best_cost=$\emptyset$

  for path in all the possible paths **do**

    expected_cost = $\sum_{X \in N^m} D(X|\tau_t) *$ compute_cost(path, X)

    if expected_cost < best_cost **then**

      best_path=path

      best_cost=expected_cost

    end if

  end for

  $n_{t+1} \leftarrow$ first node in best_path

  Travel to node n_{t+1}

  Update D, Y_{t+1} and τ_{t+1} based on the occurrence of objects at n_{t+1}

  $t = t + 1$

end while

Algorithm 6 Salesman Search

Require: A graph $G = (N, E)$, a set of objects O, a distribution D, start location node n_0

Initialize n_0, Y_0 and τ_0 with $t = 0$

shortest_path $\leftarrow$ compute_shortest_path(N, E)$\{$shortest path that visits all the nodes in $N\}$

while $Y_{t,i} \neq 0, \forall i \in [k]$ **do**

  $n_{t+1} \leftarrow$ next_node_in_path(shortest_path, n_t)

  Travel to n_{t+1}

  Update D, Y_{t+1} and τ_{t+1} based on the occurrence of objects at n_{t+1}

  $t = t + 1$

end while

Algorithm 7 Offline Optimal Search

Require: A graph $G = (N, E)$, a set of objects O, objects locations X, a distribution D, start location node n_0

Initialize n_0, Y_0 and τ_0 with $t = 0$

shortest_path $\leftarrow$ compute_shortest_path(X, E)$\{$shortest path that visits all the nodes in $X\}$

while $Y_{t,i} \neq 0, \forall i \in [k]$ **do**

  $n_{t+1} \leftarrow$ next_node_in_path(shortest_path, n_t)

  Travel to n_{t+1}

  Update D, Y_{t+1} and τ_{t+1} based on the occurrence of objects at n_{t+1}

  $t = t + 1$

end while