
Algorithm 1 DRL-TSBC 的学习过程

Require: 超参数: 经验池大小 M , 批次大小 B , 折扣因子 γ , 随机动作概率 ϵ , 最大模拟次数 E , 学习频率 P , 模型更新频率 O , 首班车时间 t_s , 末班车时间 t_e

Ensure: 训练好的 DQN 参数 θ

```
1: 初始化经验池  $D$ , 主网络参数  $\theta$ , 和目标网络参数  $\theta^- = \theta$ 
2: 使用  $\theta$  初始化主网络  $Q(s, a; \theta)$ , 并且使用  $\theta^-$  初始化目标网络  $Q(s, a; \theta^-)$ 
3: for  $episode = 1$  到  $E$  do
4:   初始化公交仿真环境
5:   for 所有  $i = t_s$  转到  $t_e$  进行 do
6:     if 在  $[0, 1]$  区间内随机取出的一个实数小于  $\epsilon$  则 then
7:       随机选择动作  $a$ 
8:     else
9:        $a = \arg \max_a Q(s, a; \theta)$ 
10:    end if
11:    通过  $a$  计算  $I^{up}$  and  $I^{down}$ 
12:    if  $I^{up} < T_{min}$  则 then
13:       $a_{up} \leftarrow 0$ 
14:    end if
15:    if  $I^{up} > T_{max}$  则 then
16:       $a_{up} \leftarrow 1$ 
17:    end if
18:    if  $I^{down} < T_{min}$  则 then
19:       $a_{down} \leftarrow 0$ 
20:    end if
21:    if  $I^{down} > T_{max}$  则 then
22:       $a_{down} \leftarrow 1$ 
23:    end if
24:     $a \leftarrow (a_{up}, a_{down})$ 
25:    在环境中应用动作  $a$  并得到奖励  $r$  和下一个状态  $s'$ 
26:    将四元组  $(s, a, r, s')$  添加到经验池  $D$  中, 同时如果  $|D| > M$ , 删除经验池中最旧的经验
27:     $s = s', i = i + 1$ 
28:    if  $|D| > M$  且  $i \bmod P = 0$  则 then
29:      从经验池  $D$  中随机取出  $B$  个四元组  $(s, a, r, s')$ 
30:      通过式 (2.18) 计算损失函数  $L$ 
31:      使用 Adam 反向传播通过  $L$  更新  $\theta$ 
32:      每进行  $O$  次学习, 更新目标网络参数  $\theta^- = \theta$ 
33:    end if
34:  end for
35: end for
```

Algorithm 2 DRL-TSBC 的推理过程

Require: 训练好的 DQN 参数 θ , 最大发车间隔 T_{\max} , 最小发车间隔 T_{\min}

Ensure: 最终公交时刻表 (发车时间点列表)

- 1: 初始化公交仿真环境
 - 2: **for** 所有 $i = t_s$ 转到 t_e 进行 **do**
 - 3: $a = \arg \max_a Q(s, a; \theta)$
 - 4: 通过 a 计算 I^{up} and I^{down}
 - 5: **if** $I^{up} < T_{\min}$ 则 **then**
 - 6: $a_{up} \leftarrow 0$
 - 7: **end if**
 - 8: **if** $I^{up} > T_{\max}$ 则 **then**
 - 9: $a_{up} \leftarrow 1$
 - 10: **end if**
 - 11: **if** $I^{down} < T_{\min}$ 则 **then**
 - 12: $a_{down} \leftarrow 0$
 - 13: **end if**
 - 14: **if** $I^{down} > T_{\max}$ 则 **then**
 - 15: $a_{down} \leftarrow 1$
 - 16: **end if**
 - 17: $a \leftarrow (a_{up}, a_{down})$
 - 18: 在环境中应用动作 a 并得到下一个状态 s'
 - 19: $s = s', i = i + 1$
 - 20: **end for**
 - 21: 选择发车次数更多的方向
 - 22: 从时刻表中删除该方向的倒数第二次发车
 - 23: $k \leftarrow$ 此时该方向的总发车次数
 - 24: **while** 时刻表中的第 k 次发车和第 $k - 1$ 次发车的间隔大于 T_{\max} 进行 **do**
 - 25: 将第 $k - 1$ 次发车的时间推迟直到其与下一次发车的间隔为 T_{\max}
 - 26: $k = k - 1$
 - 27: **end while**
-