

New Coverage Algorithm

Charles Zhang

July 1 2020

```
options(digits = 4)
```

Environment:

```
""
```

```
1 -2 -3 -4
```

```
5 -6 -7 -8
```

```
9 -10 -11 -12
```

```
""
```

```
V = 12
```

```
S = 1:V
```

```
END = 5
```

```
START = 1
```

```
R = matrix(-1, 12, 12)
```

```
R[1,c(2,5)] = 0
```

```
R[2,c(1,3,6)] = 0
```

```
R[3,c(2,4,7)] = 0
```

```
R[4,c(3,8)] = 0
```

```
R[5,c(1,6,9)] = 0
```

```
R[6,c(2,5,7,10)] = 0
```

```
R[7,c(3,6,8,11)] = 0
```

```
R[8,c(4,7,12)] = 0
```

```
R[9,c(5,10)] = 0
```

```
R[10,c(6,9,11)] = 0
```

```
R[11,c(7,10,12)] = 0
```

```
R[12,c(8,11)] = 0
```

```
R[2,1] = 100
```

```
R[5,1] = 100
```

```
R
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## [1,]  -1   0  -1  -1   0  -1  -1  -1  -1  -1  -1  -1
## [2,] 100  -1   0  -1  -1   0  -1  -1  -1  -1  -1  -1
## [3,]  -1   0  -1   0  -1  -1  -1   0  -1  -1  -1  -1
## [4,]  -1  -1   0  -1  -1  -1  -1   0  -1  -1  -1  -1
## [5,] 100  -1  -1  -1  -1   0  -1  -1   0  -1  -1  -1
## [6,]  -1   0  -1  -1   0  -1   0  -1  -1   0  -1  -1
## [7,]  -1  -1   0  -1  -1   0  -1   0  -1  -1   0  -1
## [8,]  -1  -1  -1   0  -1  -1   0  -1  -1  -1  -1   0
## [9,]  -1  -1  -1  -1   0  -1  -1  -1  -1   0  -1  -1
## [10,] -1  -1  -1  -1  -1   0  -1  -1   0  -1   0  -1
## [11,] -1  -1  -1  -1  -1  -1   0  -1  -1   0  -1   0
## [12,] -1  -1  -1  -1  -1  -1  -1  -1   0  -1  -1   0
```

```
Q = matrix(0, 12, 12)
```

```
alpha = 0.6
```

```

rounds = 500
r = 1
get_actions <- function(s) {
  a = c()
  for (i in 1:V) {
    if(R[s,i] != -1) a = c(a, i)
  }
  return(a)
}

while (r <= rounds) {
  s = sample(S, 1)
  while (TRUE) {
    action_space = get_actions(s)
    action <- sample(action_space, 1)
    s_next <- action
    actions_next = get_actions(s_next)
    qs = c()
    for (i in actions_next) qs = c(Q[s_next,i], qs)
    Q[s,action] <- R[s,action] + alpha * max(qs)
    s = s_next
    if (s == END) break
  }
  r <- r+1
}
Q

```

```

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
## [1,]  0.0 93.75  0.00  0.00 93.75  0.00  0.00  0.00  0.00  0.00  0.00
## [2,] 156.2  0.00 56.25  0.00  0.00 56.25  0.00  0.00  0.00  0.00  0.00
## [3,]  0.0 93.75  0.00 33.75  0.00  0.00 33.75  0.00  0.00  0.00  0.00
## [4,]  0.0  0.00 56.25  0.00  0.00  0.00  0.00 20.25  0.00  0.00  0.00
## [5,] 156.2  0.00  0.00  0.00  0.00 56.25  0.00  0.00 56.25  0.00  0.00
## [6,]  0.0 93.75  0.00  0.00 93.75  0.00 33.75  0.00  0.00 33.75  0.00
## [7,]  0.0  0.00 56.25  0.00  0.00 56.25  0.00 20.25  0.00  0.00 20.25
## [8,]  0.0  0.00  0.00 33.75  0.00  0.00 33.75  0.00  0.00  0.00  0.00
## [9,]  0.0  0.00  0.00  0.00 93.75  0.00  0.00  0.00  0.00 33.75  0.00
## [10,]  0.0  0.00  0.00  0.00  0.00 56.25  0.00  0.00 56.25  0.00 20.25
## [11,]  0.0  0.00  0.00  0.00  0.00  0.00 33.75  0.00  0.00 33.75  0.00
## [12,]  0.0  0.00  0.00  0.00  0.00  0.00  0.00 20.25  0.00  0.00 20.25
##      [,12]
## [1,]  0.00
## [2,]  0.00
## [3,]  0.00
## [4,]  0.00
## [5,]  0.00
## [6,]  0.00
## [7,]  0.00
## [8,] 12.15
## [9,]  0.00
## [10,]  0.00
## [11,] 12.15
## [12,]  0.00

```

```

path = c()
state = START
Q[Q == 0] <- 1000
while (length(path) < V)
{
  pre_state = state
  path = c(path, state)
  state = match((min(Q[state,])), Q[state,])
  Q[pre_state, ] = 1000
  Q[, pre_state] = 1000
}
path

```

```
## [1] 1 2 3 4 8 12 11 7 6 10 9 5
```