گزارش پروژه دوم هوش مصنوعی

حسین بهشتی فرد (۹۶۳۱۰۱۱)

---

۱) **هیوریستیک تعداد قارچ های باقی مانده در صورتی قابل قبول است که هدف خوردن تمامی قارچ ها باشد. در این مسئله اگر یک قارچ آبی و یک قارچ قرمز خورده شود به هدف رسیده ایم پس بصورت خوشبینانه می توان در دو حرکت به هدف رسید در صورتی که اگر تعداد قارچ ها بیشتر از ۲ باشند این هیوریستیک غیر قابل قبول است زیرا ممکن است تخمینی که می زند بیشتر از تعداد گام های مورد نیاز برای رسیدن به هدف باشد. اما این هیوریستیک باعث می شود در حالاتی(قارچ خورده می شود) به گام های قبلی بر نگردیم و بتوان از شانه ها فرار کرد. زیرا هنگامی که قارچی خورده می شود هزینه های تخمینی کمتر می شود و مسیر های جدید بیشتر مورد انتخاب قرار می گیرند. به عنوان یک سناریوی ساده فرض شود که در یک مسیر تک بعدی هستیم و فقط اکشن بالا و پایین را داریم و در مسیر تعدادی قارچ وجود دارد. پس از خوردن قارچ اگر مجدد به بالا برود تا اکشن های بالا را انجام دهد سری بعدی قطعا مستقیم به پایین به سمت هزینه تخمینی کمتر می آید و در شانه بالا و پایین نمی کند.**

هیوریستیک کوچک ترین فاصله منهتن یک هیوریستیک قابل قبول است. زیرا همواره هزینه تخمینی کمتر مساوی از هزینه ی واقعی است. اگر هنوز هیچ قارچی خورده نشده باشد یا نزدیک به قارچ همرنگ خورده شده باشد، پس از خوردن قارچ باید به سراغ قارچ دوم برود. با فرض اینکه یک قارچ خورده باشد و نزدیک به قارچ با رنگ مخالف است با فرض اینکه مانعی بر سر راه نباشد این تخمین برابر هزینه واقعی می شود.

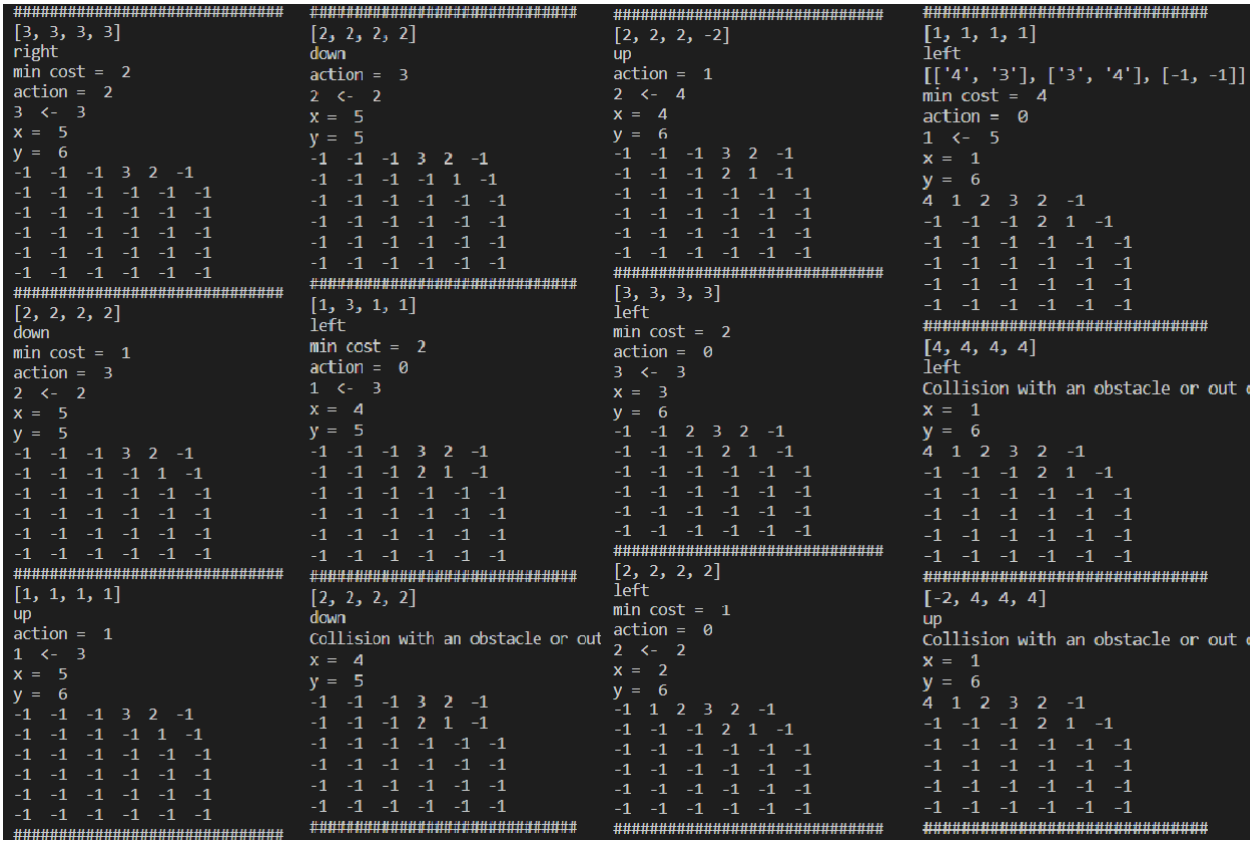
از مشکلاتی که این هیوریستیک می تواند داشته باشد، در نظر نگرفتن موانع سر راه است و مجبور است زیرا اگر قارچ پشت موانع باشد اکشن ها را انجام می دهد تا موانع را کشف کند. همینطور پس از خوردن قارچ هیوریستیک های قبلی محاسبه شده بی اعتبار می شوند، زیرا آن ها بر حسب خوردن قارچ فعلی بوده و پس از خوردن قارچ در صورتی که به هدف نرسیده باشیم، این تخمین ها غیر واقعی اند و می تواند موجب انجام حرکات اضافه برای خارج شدن از مینیموم محلی شود.

بیشترین فاصله منهتن بین هر دو قارچ هیوریستیکی غیر قابل قبول است زیرا می تواند تخمین بیشتری از هزینه ی واقعی بزند.

۲) نمونه خروجی با استفاده از هیوریستیک کمترین فاصله منهتن تا هر قارچ:

در هر مرحله هزینه H تمامی استیت ها و تخمین هزینه هر اکشن در استیت حاضر نشان داده شده است. و همینطور در صورت تغییر هر کدام هزینه تخمینی جدید چاپ شده است.

در این نمونه با ۳۶ حرکت به هدف رسیدیم.



```
##############################
[-2, -2, 4, 4]
down
Collision with an obstacle or out of range
x =  1
y =  6
4  1  2  3  2  -1
-1  -1  -1  2  1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[-2, -2, 4, -2]
right
action =  2
4  <-  2
x =  2
y =  6
2  1  2  3  2  -1
-1  -1  -1  2  1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[5, 1, 1, 1]
up
Collision with an obstacle or out of range
x =  2
y =  6
2  1  2  3  2  -1
-1  -1  -1  2  1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
```

```
-1  -1  -1  -1  -1  -1
##############################
[5,  2, 1, 1]
right
action =  2
1  <-  3
x =  3
y =  6
2  1  2  3  2  -1
-1  -1  -1  2  1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[2, 2, 2, 2]
left
action =  0
2  <-  2
x =  2
y =  6
2  1  2  3  2  1
-1  -1  -1  2  1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[5, -2, 3, 1]
down
min cost =  2
action =  3
1  <-  3
x =  2
y =  5
2  3  2  3  2  -1
-1  2  -1  2  1  -1
-1  -1  -1  -1  -1  -1
 1   1   1   1   1   1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
location x =  1 location y =  6
Action = right
2  <-  4
H Updated //  2  <-  4
##############################
```

```
##############################
[2, 2, 2, 2]
left
Collision with an obstacle or out of range
x =  2
y =  5
4  3  2  3  2  -1
-1  2  -1  2  1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[-2, 2, 2, 2]
down
min cost =  1
action =  3
2  <-  2
x =  2
y =  4
4  3  2  3  2  -1
-1  2  -1  2  1  -1
-1  1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[1, 1, 1, 1]
down
Collision with an obstacle or out of range
x =  2
y =  4
4  3  2  3  2  -1
-1  2  -1  2  1  -1
-1  1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
```

```
##############################
[1, 1, 1,  2]
right
[['4', '3'], [-1, -1], [-1, -1]]
min cost =  2
action =  2
1  <-  3
x =  3
y =  4
4  3  2  3  2  -1
-1  2  -1  2  1  -1
-1  1  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[2, 2, 2, 2]
left
action =  0
2  <-  2
x =  2
y =  4
4  3  2  3  2  -1
-1  2  -1  2  1  -1
-1  1  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[1, 1, 3, -2]
left
min cost =  4
action =  0
1  <-  5
x =  1
y =  4
4  3  2  3  2  -1
-1  2  -1  2  1  -1
4  1  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
```

```
##############################
[4, 4, 4, 4]
right
action =  2
4  <-  2
x =  2
y =  4
4  3  2  3  2  -1
-1  2  -1  2  1  -1
2  1  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[5, 1, 3, -2]
up
action =  1
1  <-  3
x =  2
y =  5
4  3  2  3  2  -1
-1  2  -1  2  1  -1
2  3  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
location x =  1 location y =  4
Action = right
2  <-  4
H Updated //  2  <-  4
location x =  2 location y =  5
Action = down
2  <-  4
##############################
[-2, 2, 2, 4]
right
Collision with an obstacle or out of range
x =  2
y =  5
4  3  2  3  2  -1
-1  2  -1  2  1  -1
4  3  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
```

```
##############################
[-2, 2, -2, 4]
up
action =  1
2  <-  4
x =  2
y =  6
4  3  2  3  2  -1
-1  4  -1  2  1  -1
4  3  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[5, -2, 3, 3]
right
action =  2
3  <-  3
x =  3
y =  6
4  3  2  3  2  -1
-1  4  -1  2  1  -1
4  3  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[2, 2, 2, 2]
right
action =  2
2  <-  4
x =  4
y =  6
4  3  2  3  2  -1
-1  4  -1  2  1  -1
4  3  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
```

```
##############################
[3, 3, 3, 3]
right
action =  2
3  <-  3
x =  5
y =  6
4  3  2  3  2  -1
-1  4  -1  2  1  -1
4  3  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[2, 2, 2, 2]
down
action =  3
2  <-  2
x =  5
y =  5
4  3  2  3  2  -1
-1  4  -1  2  1  -1
4  3  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[3, 3, 1, 1]
right
min cost =  2
action =  2
1  <-  3
x =  6
y =  5
4  3  2  3  2  -1
-1  4  -1  2  1  2
4  3  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
```

```
##############################
[2, 2, 2, 2]
right
Collision with an obstacle or out of range
x =  6
y =  5
4  3  2  3  2  -1
-1  4  -1  2  1  2
4  3  2  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[2, 2, -2, 2]
down
min cost =  1
action =  3
2  <-  2
x =  6
y =  4
4  3  2  3  2  -1
-1  4  -1  2  1  2
4  3  2  -1  -1  1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
##############################
[1, 1, 1, 1]
left
[['3', '1'], ['6', '1'], [-1, -1]]
min cost =  2
action =  0
1  <-  3
x =  5
y =  4
4  3  2  3  2  -1
-1  4  -1  2  1  2
4  3  2  -1  2  1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
-1  -1  -1  -1  -1  -1
steps number = 36
```