

Alunos: Rubia e Willian

- 1) Foi utilizado como heurística a distância euclidiana. É admissível, pois para qualquer nó  $n$ , a função  $h(n)$  é menor ou igual que a distância real até o objetivo.

- 2) `deliberar()` para busca online:

```
ct++
plan = busca.gerarSolucao()
if não chegou ao objetivo
    executarlr(plan[ct])
    atualiza custo
    estAtu = suc(estAtu, plan[ct])
else chegou ao objetivo
    return -1
```

`deliberar()` para busca offline:

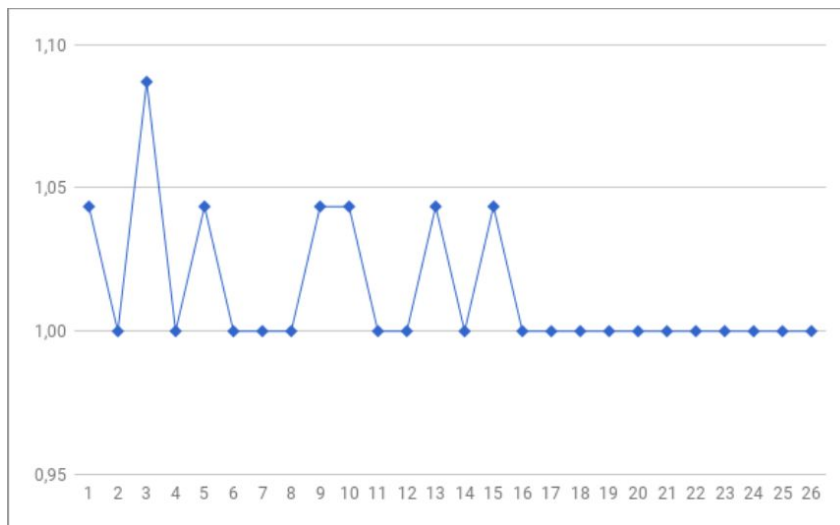
```
if não chegou ao objetivo
    decisao=busca.decideAcao(ações possíveis)
    plan.add(decisao)
    executarlr(decisao)
    atualiza custo
else
    return -1
```

- 3) Foram encontradas 6 soluções para o problema:

```
N NE SE SE L L NE L NE
N NE SE SE L L NE NE L
N NE NE NE L L SE L SE
N NE NE NE L L L SE SE
N NE SE SE L L L NE NE
N NE NE NE L L SE SE L
```

O custo mínimo obtido foi de 11.5.

- 4) Foram necessárias 16 execuções do algoritmo para que a razão de competitividade atingisse o valor de 1 de forma estável e 15 execuções para encontrar todas as soluções ótimas.



5)

- 6) Foram necessárias 23 execuções do algoritmo para que a razão de competitividade atingisse o valor de 1 de forma estável e 40 execuções para encontrar todas as soluções ótimas.

